



Universidad  
del País Vasco

Euskal Herriko  
Unibertsitatea

# Robotika kolaboratiboa nabigazio autonomoarekin bihurketa prozesuak egiteko etengabeko lanetan



Eraginkorra

Industria 4.0

Ergonomia

Kalitatea

Etengabeko-lana

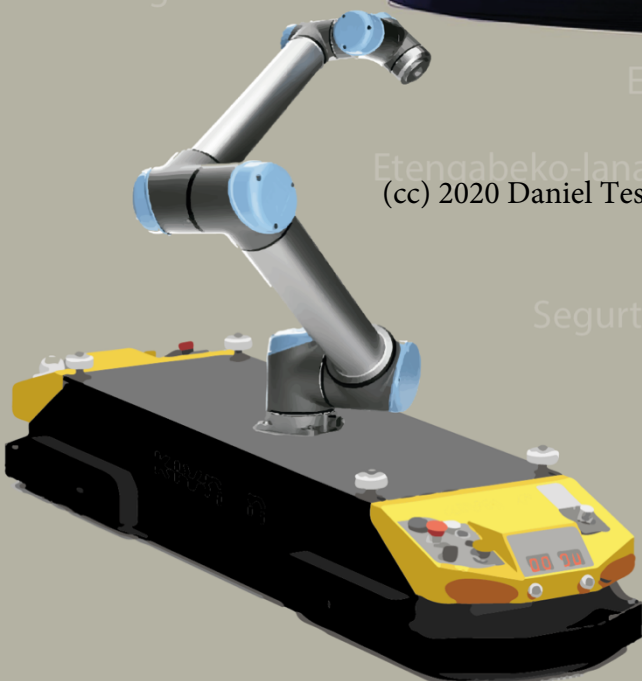
(cc) 2020 Daniel Teso Fernández de Betoño (cc by-nc-nd 4.0)

Segurtasuna

Produktibitatea

Kolaboratiboa

Eskuragarria



Daniel Teso

Fernández de Betoño -2020-



# Robotika kolaboratiboa nabigazio autonomoarekin bihurketa prozesuak egiteko etengabeko lanetan

---

---

Tesiaren autorea:

**Daniel Teso Fernández de Betoño**

Tesiaren zuzendaria:

**Ekaitz Zulueta Guerrero**

Doktoretza programa:

**Kontrol Ingeniaritza, Automatizazioa eta Robotika**

Aurkezpen urtea:

**2020**



**Mercedes-Benz**

FÁBRICA VITORIA

eman ta zabal zazu



Universidad  
del País Vasco

Euskal Herriko  
Unibertsitatea

VITORIA-GASTEIZKO  
INGENIARITZA  
ESKOLA  
ESCUELA  
DE INGENIERÍA  
DE VITORIA-GASTEIZ



# Laburpena

---

---

Proiektu honek bi prototipo ezberdin jasotzen ditu. Alde batetik, nabigazio autonomoa erabiltzen duen AMR prototipo baten garapena erakutsiko da. Bestetik, Mercedes – Benz barnean landuriko tresneria baten ikerketa aurkeztuko da. AMR-ak plataforma mugikor ahaltsuak dira eta hauek barneko nabigazio autonomoa erabiltzen dute, edozein gune ezagunetik mugiarazteko. Horregatik, Gasteizko Ingeniaritza Eskolak halako plataforma baten diseinua burutzen hasi da, lokalizazio algoritmoak lantzeko. Robot mugikor honek elementu industrialak erabiliko ditu eta hauek intelijentzia garapenean zenbait oztopo ezarriko ditu. AMR -ri robot bat atxiki ahal zaio, horregatik Mercedes – Benz barnean elementu komertzialekin AMR bateri robot kolaboratibo bat ezarri zaio. Garapen honek lan postuen efizientziak lantzeko baliagarria izango da eta horretarako robotak etengabeko lanetan mugiarazi, kokatu eta kalitatezko lana burutu behar du.

# Abstract

---

---

This project includes two different prototypes. On the one hand, an AMR prototype with autonomous navigation is developed. On the other hand, in Mercedes – Benz an intelligent tool is researched. AMRs are powerful mobile platforms that use autonomous indoor navigation technique to move from any known place. For this reason, the Engineering School of Vitoria-Gasteiz has started to design these platforms to research indoor localization algorithms by using industrial equipment. These types of industrial devices set some obstacles in the intelligence development. Apart from that, in the AMR a collaborative robot can be attached. Therefore, in Mercedes – Benz by implementing commercial devices, this equipment is assembled. This development will be useful to generate efficiencies on the workplace, in which the robot must move, reference itself and perform quality work in continuous moving.



# Goraintziak

---

---

la hiru urteetan zehar, pertsona askoei eskerrak eman behar zaizkie, beraien laguntza ezin bestekoa izan delako, tesi honetan agerturiko oztopoak gainditzeko. Halako testuak idaztea arraro egiten zait, batez be beti norbait ahaztu dudala pentsatzen dudalako. Horregatik, lehengo hitz hauek nire inguruan egon direnentzako dira, non barreak, garagardoak eta etengabeko sostengua ez diren falta. Mila esker emandako euskarriengatik.

Familia munduko kide inportanteak dira. Sentimendu guztiak jasaten dituztenak dira. Txikitatik edozein arazo aurrean, jarraitzeko inpultsoak ematen dituztenak. Beraz, mila esker Betoñada emandako sostenguagatik. Gainera, amari esker handiak eman behar dizkiot, beraren laguntzaren gabe, dokumentu honetan egindako ikerketa ez lirakeelako existituko. Nire Anaiak, mekanikaz gain, kontrola, automatizaketa eta robotika ikasten hastearen erruduna izan zen. Baina ahal eta be, eskertzen diot edozein arazo baten aurrean, zalantzak argitzeko prest egon dela. Bestalde, familian sarturiko txikikuaskun berriari, lehengo unetik unibertso berri bat sortzeko prest egon delako, baita azken pausuetan proiektua ahurrera ateratzen lagundu duelako ere, mila esker bihotz bihotzez.

Gasteizko Ingeniaritza eskolak garapen honetarako ezin bestekoa izanda ere. Bertako jendea bigarren familia bat da. Hasieran irakasleak, baina orain lagunak dira. Nire ikasketak aurrea nola atera ditudan ikusi duen jendea da eta edozein arazoren aurrean, laguntzeko prest dauden pertsonak dira. Horregatik mila esker. Bereziki nire Tesiaren Zuzendariari, Ekaitz Zulueta Guerrero eskerrak eman behar dizkiot. Berak nire gain ezarritako konfiantza neurgaitza delako eta proiektu hau garatzeko emandako aukera eskertu ezina delako. Gainera, Unai Fernández Gamiz-i eskerrak eman behar dizkiot ere, artikuluetan emandako euskarriengatik. Bi irakasle hauek duela zortzi urte ezagutu nituen eta orain lagun bezala kontsideratzen ditut, azkeneko hiru urte hauetan sorturiko harremanengatik. Azkenik, Iñigo Aramendia eskertzen dizut eskainitako baliabide guztiengatik eta eskainitako laguntzagatik.

Betiko *“KiniKini”* lagunak. Urte asko igaro ditugu elkarrekin eta urteak igaro ahala, harremana gora eta bera joan da. Hala eta be, gure adiskidetasuna mantendu dugu, barreen, haserrean, etab. ondorioz. Gure betiko lemarekin *“Siempre arriba”* gauza asko lortu ditugu, mila esker hor egoteagatik. Lagun berriei eskerrak eman behar dizkiet ere, azken bolada honetan haien iritziak ezin bestekoak izan dira, ikuspuntu berriak lortzeko. Harreman sendo eta *“impexable”* lortu dugu, beste gogoragarri diren momentuak lortuz.

Azkenik, Gasteizko Mercedes – Benz-eko lantegiak emandako aukeragatik, hitz hauek eskainitako laguntza eskertzeko da. Bertan edonork arazo baten aurrean, haren esperientzian oinarrituriko ikuspuntua emateko prest dago eta horrek zure arazoa aurre ateratzeko oso baliagarria da. Emilio Titos-ek, Jose Carlos Velasco-k, Javier Loredok, Javier Gómez-ek, Jose Antonio Hernando-k, José Ignacio Cervera-k, Roberto Dediego-k, Oier Aguirre-k, Tomás Herndadez-ek eta amaierako muntaketa mantentze sailak proiektu hau aurrera ateratzeko eman dituzten baliabideak, gogoak eta laguntzak ezin bestekoak izan dira. Baita zenbait alditan sorturiko barreak gogoragarriak dira. Horregatik mila esker nire gainean ezarritako konfiantzarenagatik eta emandako aukerarenagatik.





# Aurkibidea

---

---

I. ATALBURUA: SARRERA .....	- 1 -
1 Mercedes – Benz Gasteizko Lantegia .....	- 2 -
2 Lantegiaren antolaketa .....	- 4 -
3 Mercedes – Benz-eko ikerketa guneak .....	- 5 -
4 Mercedes – Benz-eko lan filosofia .....	- 6 -
5 Tesiaren garapena .....	- 7 -
6 Proiektuaren garapen filosofia .....	- 8 -
II. ATALBURUA: TEKNIKAREN EGOERA .....	- 11 -
1 Gidaketa autonomoa .....	- 12 -
2 Industria barneko ibilgailuak .....	- 13 -
3 Nabigazioa .....	- 16 -
3.1 Online Nabigazioa .....	- 16 -
3.2 Offline Nabigazioa .....	- 19 -
4 Maparen eraikitzea eta lokalizazioa .....	- 20 -
5 Robot Kolaboratiboen egoera .....	- 21 -
III. ATALBURUA: TEKNOLOGIA FROGATZEN .....	- 23 -
1 Autonomous Mobile Robot .....	- 24 -
1.1 Frogapenak eta Ondorioztapenak .....	- 27 -
1.2 Modelo Matematikoa .....	- 28 -
1.3 Motorren Modeloa .....	- 30 -
1.4 Kivnon-ekin egindako bilerak .....	- 35 -
2 Robot Kolaboratiboa .....	- 35 -
2.1 Roboten ziurtagirien egoera .....	- 36 -
2.2 UR frogapenaren ondorioak .....	- 39 -
3 Bihurkinak .....	- 39 -
4 Simulazioak .....	- 41 -
4.1 Puntu finko batera mugiarazi .....	- 43 -
4.2 Mugikor den puntu batera mugitu .....	- 45 -
4.3 3D simulagailua .....	- 48 -
IV. ATALBURUA: APLIKAZIOAREN BURUTZEA EXISTITZEN DENAREKIN .....	- 51 -
1 Garatutako plataforma .....	- 51 -
2 Languneen azterketa .....	- 52 -
V. ATALBURUA: BEHARREN HAUSNARKETA .....	- 55 -
1 AMR-rentzako puntu mugikorraren erreferentzia .....	- 55 -
2 Robot Kolaboratiboa-rentzako erreferentzia .....	- 56 -
3 Sinkronizazioa .....	- 58 -
4 Integrazioa digitala .....	- 59 -
VI. ATALBURUA: BEHARRAK GAUZATU ETA PROTOTIPOAK BURUTU .....	- 61 -
1 Unibertsitateko prototipoak .....	- 62 -
1.1 Unibertsitatearen Prototipoaren Definizioa .....	- 62 -
1.2 Prototipoaren diseinua .....	- 63 -
1.3 Inteligentziak garatzen .....	- 67 -
1.3.1 LiDAR-aren integrazioa .....	- 67 -
1.3.2 Roomba Mode .....	- 72 -
1.3.3 Segurtasun funtzioa TwinCat barnean .....	- 77 -
1.3.4 Unibertsitateko AMR-ren modelo matematikoa .....	- 78 -
1.3.5 Barne lokalizazioa .....	- 80 -

1.3.6	Sare neuronalen bitartezko lokalizazioa.....	- 85 -
1.3.7	Sare Neuronalak Nabigazio algoritmoentzako .....	- 88 -
1.4	Lehen Prototipoaren Gurpilen kontrola .....	- 94 -
1.5	Prototipoaren zehaztasunaren hobekuntza .....	- 96 -
1.6	Prototipoaren Posizionamendua .....	- 100 -
1.7	Ondorioak eta hurrengo pausuak.....	- 104 -
2	Mercedes – Benz-eko prototipoak .....	- 106 -
2.1	Lehengo Fasea .....	- 106 -
2.2	Bigarren Fasea .....	- 108 -
2.3	Hirugarren Fasea .....	- 111 -
2.3.1	Sare neuronalen .....	- 112 -
2.3.2	Pattern Matching.....	- 119 -
2.3.3	Geometric Matching.....	- 121 -
2.3.4	Kontrola .....	- 124 -
2.3.5	Prozesuaren egoera zehazteko programa .....	- 125 -
2.3.6	Ondorioak.....	- 126 -
2.4	Laugarren fasearen prototipoa.....	- 126 -
2.5	Bosgarren fasearen prototipoa .....	- 130 -
VII. ATALBURUA: EKARPENAK, ONDORIOAK ETA HURRENGO URRATSAK.....		- 131 -
1	Autorearen ekarpenak.....	- 131 -
2	Ondorioak eta hurrengo urratsak.....	- 133 -
ERANSKINAK.....		- 137 -
1	Eskema elektrikoa.....	- 137 -
2	Tesi honetan burututako Kodeak .....	- 138 -
ERREFERENTZIAK.....		- 171 -
WEBGUNEN ERREFERENTZIAK .....		- 175 -

## Irudien Aurkibidea

---



---

IRUDIA 1	DKW F89L EREDUA, BERE AZPIAN GASTEIZKO LANTEGIAREN FABRIKAZIO PROZESUA. ....	- 2 -
IRUDIA 2	DKW F 1000-REN ALDIZKARI BATEKO ARGAZKIA.....	- 3 -
IRUDIA 3	MERCEDES - BENZ FURGONETA DESBERDINAK A) VITO MODELOA B) V-CLASS C) EQV .....	- 3 -
IRUDIA 4	FABRIKAZIO PROZESU DESBERDINAK URTEETAN ZEHAR A) ESKUZKO SOLDADURA PROZESUA (1968)(OLD BENZ WEBGUNETIK LORTURIKO ARGAZKIA) B) SOLDAKETA AUTOMATIKOA (2015) (FABRICANDO MADE IN SPAIN-ETIK LORTURIKO ARGAZKIA) .....	- 4 -
IRUDIA 5	ROBOTIKAREN KRONOGRAMA .....	- 7 -
IRUDIA 6	U TEORIAREN GARAPENA: A) OTTO SCHARMER AURKEZTUTAKO FILOSOFIA (PRESENCING INSTITUTE-KO ARGAZKIA), B) TESIAN ERABILITAKO METODOLOGIA, FASE BAKOITZA GARATZEKO .....	- 8 -
IRUDIA 7	MUNTAKETAREN EREDUA: A) BI ROBOTEN MUNTAKETA EREDUA, B) LAN EREMUAREN DEFINIZIOA .....	- 12 -
IRUDIA 8	INDUSTRIA 4.0 ARLOAN FINKATZEN DIREN PAUSUAK .....	- 13 -
IRUDIA 9	CARRETTILLAS MAYOR SALTZEN DUEN AGV-A (CARRETTILLAS MAYOR-ETIK LORTUTAKO IRUDIA).....	- 14 -
IRUDIA 10	KIVNON KV-10 PRODUKTUA (KIVNON-ETIK LORTURIKO IRUDIA) .....	- 14 -
IRUDIA 11	MERKATUKO PRODUKTUA (LOGISTICS IQ LORTURIKO IRUDIA) .....	- 15 -
IRUDIA 12	LIDAR-RAREN MAPA.....	- 17 -
IRUDIA 13	CNN BATEN GERUZEN ANTOLAKETA .....	- 18 -
IRUDIA 14	BURUAN IZANDAKO GALDERAK .....	- 23 -
IRUDIA 15	AMR-REN PLC-A.....	- 25 -
IRUDIA 16	AMR-KO KOMUNIKAZIOAK .....	- 26 -
IRUDIA 17	KONEKTORE BIZKORRAK .....	- 27 -
IRUDIA 18	MOTORREN KOKAPENA.....	- 28 -

IRUDIA 19 KIVNON PLATAFORMAREN ABIADURAREN DESKONPOSAKETA .....	- 29 -
IRUDIA 20 Vc ABIADURAREN DESKONPOSAKETA .....	- 29 -
IRUDIA 21 MOTORREN MODELOA LORTZEKO BURUTUTAKO PROGRAMA .....	- 32 -
IRUDIA 22 MOTORREN ERANTZUNA: A) ETA B) LEHENGO FROGAPENEN ERANTZUNA DEN, C) ETA D) BIGARREN FROGAPENEN ERANTZUNA.....	- 33 -
IRUDIA 23 ROBOT INDUSTRIALENTZAKO HIRU MOTATAKO LEGEDIEN EREDUAK [136].....	- 36 -
IRUDIA 24 ISO/TS 15066 LEGEDIAN BILDUTAKO DATUAK, GORPUTZAK JASAN DEZAKEEN TALKAK JAKITEKO .....	- 37 -
IRUDIA 25 DIN EN ISO 13849 - 1:2016-06 AIPATUTAKO SEGURTASUN SISTEMEN SAILKAPENAK.....	- 38 -
IRUDIA 26 TALKAREN ERAGINA AZTERTZEKO TRESNERIAK (PILZ ENPRESAK ESKAINITAKO IRUDIAK) .....	- 38 -
IRUDIA 27 PRESSURE PROFILE SYSTEMS-EK PROPOSATURIKO BESTE PRESIOA NEURGAILUAK {4}.....	- 39 -
IRUDIA 28 UR-KO 140 NM BIHURDURA PROZESUA {5}.....	- 40 -
IRUDIA 29 INDAR PAREAK APLIKATZEKO ESKUEN POSIZIONAMENDUA {6}.....	- 40 -
IRUDIA 30 ESTIC PROPOSATURIKO BIHURKARI ELEKTRONIKOA {7}.....	- 41 -
IRUDIA 31 KIVNON EREDU MATEMATIKOAREN SIMULAGAILUA .....	- 42 -
IRUDIA 32 KIVNON EREDU MATEMATIKOAREN SIMULAGAILUA .....	- 44 -
IRUDIA 33 PUNTU FINKO BATERA MUGIARAZTEKO SIMULAZIOA: A) IBILBIDEAREN ERANTZUNA, B) X ETA Y ERREPRESENTAZIOA DENBORAREKIKO .....	- 45 -
IRUDIA 34 PUNTU MUGIKORRAREN PROGRAMA: A) AMAIERAKO PUNTU MUGIKORRAREN FUNTZIOA, B) SIMULINK-EKO KODEA-46 -	- 46 -
IRUDIA 35 PUNTU MUGIKORRAREN ERANTZUNA: A) AMR-REN ETA PUNTU MUGIKORRAREN IBILBIDEA, B) AMR-REN X ETA Y POSIZIOAREN ALDAKETA DENBORAREKIKO, C) AMR-REN Y ALDAKETA DENBORAREKIKO PUNTU MUGIKORRAREKIKO KONPARATUZ.....	- 47 -
IRUDIA 36 <i>DIFFERENTIAL WHEELED ROBOT WITH LIDAR SENSOR</i> -REN EREDUAREN LAN EREMUA .....	- 48 -
IRUDIA 37 LIDAR SENSOREAREN SIMULINK-EKO BLOKEA: A) BLOKEAREN ERREPRESENTAZIOA, B) PICKEDPOINT-EN DATUEN BILKETA. ....	- 48 -
IRUDIA 38 PROIEKTUAREN SIMULAZIOAREN LAN EREMU BIRTUALA: A) FURGONETA ETA ESEKIGAILUAREN DISEINUA, B) LAN EREMU BIRTUALAREN ERREPRESENTAZIOA.....	- 49 -
IRUDIA 39 ERGONOMIA SAILKATZEKO BURUTUTAKO EXCEL PROGRAMA. ....	- 52 -
IRUDIA 40 SEIGARREN LERROAREN LAN EGOERA .....	- 52 -
IRUDIA 41 LAUGARREN LERROAREN LAN EGOERA: A) LAN BALDINTZEN ERREPRESENTAZIOA, B) FURGONETAK ERABILTZEN DITUEN RIEL-AK (MERCEDES-ETIK LORTUTAKO IRUDIA) .....	- 53 -
IRUDIA 42 ROBOT KOLABORATIBIOEN ERREFERENTZIA SISTEMA MEKANIKOEN BIDEZ (AUDI SMART FACTORY-EK AURKEZTUTAKO ARGAZKIAK) .....	- 57 -
IRUDIA 43 UNIBERTSITATEAN ERABILITAKO ELEMENTU INDUSTRIALAK: A) IPC C6925-0030 B) ETHERCAT EK1100 (BECKHOFF - ETIK LORTUTAKO IRUDIAK {8}) .....	- 63 -
IRUDIA 44 UNIBERTSITATEKO AMR-AK ERABILIKO DITUEN GURPIL MOTORDUNAK.....	- 64 -
IRUDIA 45 ENTSEGU BANKUA .....	- 65 -
IRUDIA 46 BDCM MOTORREN KONTROLEN AUKERAKETA: A) ESC KONTROL MODULUA B) PATINETE ELEKTRIKO BATEK ERABILTZEN DUEN KONTROLA .....	- 66 -
IRUDIA 47 UNIBERTSITATEKO PROTOTIPOAREN DISEINUA .....	- 66 -
IRUDIA 48 ESKOLAKO KORRIDOREAREN MAPAREN ERREPRESENTAZIOA: A) KODEA 8-K SORTZEN DUTEN MAPA, B) KODEA 9 FUNTZIOAK EMANDAKO MAPA, C) KODEA 10 MAPA IRUDIAN BILAKATZEAREN EMAITZA, D) LORTUTAKO IRUDIAREN MAPAREN ERALDAKETA .....	- 71 -
IRUDIA 49 ROOMBA MODE-REN SIMULAZIOA, A) HASIERAKO ALDIUNEKO EGOERA, B) OBJEKUAK SAIHESTEN DUEN EGOERA, C) ERABILITAKO SIMULINK-EKO PROGRAMAZIOA .....	- 73 -
IRUDIA 50 ATE IREKI EGUNA GASTEIZKO INGENIARITZA ESKOLAN EGINDAKO ARGAZKIA (INGENIARITZA_UPVEHU INSTAGRAM-ETIK LORTUTAKO IRUDIA).....	- 75 -
IRUDIA 51 UNIBERTSITATEKO AMR EGITURAREN ERREPRESENTAZIOA .....	- 78 -
IRUDIA 52 UNIBERTSITATEKO AMR-AK BIRAKETA EGOERAN .....	- 78 -
IRUDIA 53 UNIBERTSITATEKO AMR-AK BIRAKETA ABIADURAREN KALKULUEN ERREPRESENTAZIOA .....	- 79 -
IRUDIA 54 UNIBERTSITATEKO AMR-AK KOKAPENA MAPA ABSOLUTU BATEAN .....	- 80 -
IRUDIA 55 PARTIKULEN IRAGAZKIAREN FROGAPENA: A) SIMULAZIO BITARTEKO LEIHOA, B) ERREGRESIO LINEALAREN ERANTZUNA 100 LAGINKETA ONDORIOZ, C) SVD-REN ERANTZUNA 100 LAGINKETARAKO. ....	- 84 -
IRUDIA 56 COMBINATOR FUNTZIOAREN FROGAPENA, BI NEURONA ETA LAU GERUZA ERABILIZ: A) 52. LERROAREN FUNTZIOA ABIARAZITA, B) 52. LERROTIK 68-RA KODEA ABIARAZITA .....	- 86 -

IRUDIA 57 SARE NEURONALEN ENTRENAMENDUAREN ONDORIOZ EGINDAKO FROGAPENA: A) ENTRENAMENDUAREN MAPA ETA ENTRENATZEKO POSIZIOEN ERREPRESENTAZIOA, B) FROGAPENAREN ERROREAREN ERANTZUNA .....	- 87 -
IRUDIA 58 LiDAR-RAK DITUEN LIMITAZIOAK KORRIDORE LUZE ETA SIMETRIKOETAN .....	- 88 -
IRUDIA 59 SARE NEURONALAK EGITEN DUEN FILTROA, OBJEKTU EZBERDINAK SAILKATZEKO .....	- 88 -
IRUDIA 60 VITORIA – GASTEIZKO INGENIARITZA ESKOLAREN KORRIDOREEN ARGAZKIAK: (A)(B)(C)(D) SARE NEURONALA ENTRENATZEKO ERABILITAKO EREDUAK.....	- 89 -
IRUDIA 61 EGITURA SINPLEKO SARE NEURONAL BAT: A) SAREAREN EGITURA, B) SAREAK ENTRENATU OSTEAN EMANDAKO ERANTZUNA, JAKINDA URDINA LURZORUA ADIERAZTEN DUELA .....	- 90 -
IRUDIA 62 ATROUS CONVOLUTION FUNTZIONAMENDUA.....	- 91 -
IRUDIA 63 LEHENGO RESNET – 18 SEGMENTAZIO SAREAN LORTURIKO EMAITZAK: A) SAREARI EMANDAKO EZ IKASITAKO IRUDIA, B) SAREAK SORTURIKO ERANTZUNA, URDINEZ LURZORUA ADIERAZIZ.....	- 92 -
IRUDIA 64 RESNET – 18 SEGMENTAZIO SAREAK EMANDAKO EMAITZAK ERDIKO PUNTUREN KALKULUA APLIKATUTA .....	- 93 -
IRUDIA 65 SARE NEURONAL KONBULUZIONALAK APLIKATUZ, LOGO KONKRETU BATEN IDENTIFIKAPENA: A) ERABILITAKO LOGOA, B)C) LORTURIKO EMAITZAK .....	- 94 -
IRUDIA 66 ESQUINEKO GURPILAREN LEHENGO FROGAPENA .....	- 95 -
IRUDIA 67 ESQUINEKO GURPILAREN FROGAPENA KONTROLAGAILUAREN SARRERAK EZAGUTUTA .....	- 96 -
IRUDIA 68 EL – 7411 – 9014   BLDC: A) TERMINALAREN EGITURAREN ERREPRESENTAZIOA, B) KONFIGURATU BEHAR DIREN PARAMETROAK.....	- 97 -
IRUDIA 69 EL – 6751, BECKHOFF-EN CANOPEN-EKO MODULUA .....	- 98 -
IRUDIA 70 UNIBERTSITATEKO AMR-REN MOTOR BERRIAK: A) NANOTEC MOTORREN IRUDIA, B) AMR-REN MUNTAKETAREN ERREPRESENTAZIOA .....	- 99 -
IRUDIA 71 UNIBERTSITATEAK PLANTEATUKO DITUEN BESTE PLATAFORMA MUGIKORRAK SARE NEURONALAK ERABILITZEKO A) NVIDIA JERTSON AGX, B) PIONEER 3 ROBOT MUGIKORRA, C) GARATUTAKO MOLDAKETAK .....	- 105 -
IRUDIA 72 PROTOTIPOAREN LAN BALDINTZA GUZTIAK.....	- 106 -
IRUDIA 73 V1 PROTOTIPOAREN AURKEZPENA: A) BURUTUTAKO MUNTAKETA, B) DISEINATUTAKO IBILBIDEA .....	- 107 -
IRUDIA 74 V2 PROTOTIPOAREN AURKEZPENA: A) BURUTUTAKO MUNTAKETA, B) ERREFERENTZIA KOLABORATIBOAREN ERAKUSKETA, C) ERRAMINTAREN MUNTAKETA, D) INFORMAZIOAREN ERREPRESENTAZIOA .....	- 109 -
IRUDIA 75 BIHURGAILUAREN ERAGILEREN DISEINUA: A) SERBOMOTORREAN FROGAPENERAKO EREDUA, B) INPRIMATUTAKO BEHATZ FORMAKO ERAGILEA .....	- 110 -
IRUDIA 76 BIHURKINA ROBOTARI ATXIKITZEKO PLANTEATUTAKO ZIRIBORROA .....	- 110 -
IRUDIA 77 GORKA URTARAN ETA RAMIRO GÓMEZ V2 PROTOTIPOAREN ERAKUSKETAREN EGUNKARIETAKO ARGITALPENAK (EL CORREO {9} ETA GASTEIZKO UDALA {10}) .....	- 111 -
IRUDIA 78 INTELIGENTZIA ARTIFIZIALAREN SAILKAPENA (ORCALE BLOGS-ETIK LORTUTAKO IRUDIA).....	- 112 -
IRUDIA 79 LABVIEW BARNEAN DATUAK PRESTATZEKO GARATUTAKO PROGRAMA.....	- 113 -
IRUDIA 80 SAREA ENTRENATZEKO INFORMAZIOAREN PRESTAKETA.....	- 114 -
IRUDIA 81 DATUEN HANDIPENA PRESTAKUNTZA ETA LORTUTAKO EMAITZAK .....	- 115 -
IRUDIA 82 RESIDUAL NET BATEN EGITURA.....	- 116 -
IRUDIA 83 SARE NEURONALEN EGOERA: A) CNN MOTATAKO SARE EREDUAK, B) MERCEDES – BENZ BARNEAN GARATUTAKO KONPARAKETA SAREAREN IZAERAREN ARABERA .....	- 117 -
IRUDIA 84 SARE NEURONALEN EGOERA: A) ETA B) OBJEKTUAK IDENTIFIKATZEKO FUNTZIONAMENDUA, C) ERABILITAKO SAREAREN ARABERA “REGION PROPORSAL NENTWORK” TXERTATU BEHAR DEN GUNEA .....	- 118 -
IRUDIA 85 SARE NEURONALEN FROGAPENA .....	- 118 -
IRUDIA 86 PIRAMIDE MAILA BAKOITZEKO LORTZEN DEN MURRIKETA .....	- 120 -
IRUDIA 87 GRADIENTEEN TEKNIKAREN EREDUA .....	- 120 -
IRUDIA 88 IRUDIA ULERTARAZTEN DUEN METODOA.....	- 121 -
IRUDIA 89 GEOMETRIAREN BATERAGARRITASUNAREN EREDUA.....	- 122 -
IRUDIA 90 HASIERAKO PUNTUEN BILAKETA ETA KURBADUREN OSAKETA .....	- 123 -
IRUDIA 91 R TAULAREN DATUEN ERAGIKETA: A) TXANTILOI BATEN EREDUA, B) R TAULAREN EREDUA .....	- 123 -
IRUDIA 92 KIVNON-EKO ABIADURA KONTROLA .....	- 125 -
IRUDIA 93 BURUTUTAKO BIHURKIN EBALUAZIOAREN PROZEDURAREN PROGRAMA.....	- 125 -
IRUDIA 94 BIHURKIN PROZEDURA BATEN EBALUAKETA.....	- 126 -
IRUDIA 95 KAMERA ALTUERAREN ALDAKETA .....	- 127 -
IRUDIA 96 LAUGARREN PROTOTIPOAREN AURKEZPENA: A) BIBRAZIOAK EKIDITEKO PATINAK, B) KAMERAREN KOKALEKU BERRIA, C) LAUGARREN PROTOTIPOA .....	- 129 -
IRUDIA 97 AMR NABIGAZIO AUTONOMOAREN ONDORIOZ, ERREMINTAN ZUZEN BEHAR DEN POSIZIONAMENDUA: A) PROGRAMAZIOA EGITERAKO ORDUAN ROBOTEN FINKATUTAKO PUNTUA, B) ZUZENDUTAKO POSIZIONAMENDUA .....	- 129 -

## Taulen Aurkibidea

---

---

TAULA 1. AZTERTUTAKO AMR-AK .....	- 15 -
TAULA 2. ONLINE NABIGAZIOZKO ALGORITMOAK .....	- 19 -
TAULA 3. ROBOT KOLABORATIBOEN MERKATU EGOERA .....	- 22 -
TAULA 4. MOTORREN FROGAPENEN DENBORA ERANTZUNAK .....	- 34 -
TAULA 5. EK1100 MODULUAK ESKEGITA DITUEN ELEMENTUAK .....	- 64 -
TAULA 6. PARTIKULA IRAGAZKIAREN KONPARAKETAREN ERANTZUNAK .....	- 85 -
TAULA 7. SEGMENTAZIO SARE BATEK DITUEN GERUZAK .....	- 90 -
TAULA 8. GURPIL MOTORDUNEN EZAUGARRIEN LABURPENA .....	- 97 -
TAULA 9. PLATAFORMA MUGIKORRENTZAKO AUKERATUTAKO MOTORRAK .....	- 98 -
TAULA 10. KAMERA ETA HARDWAREAREN EBALUAZIOA .....	- 127 -
TAULA 11. AUTOREAREN EKARPENAK GUZTIAK .....	- 133 -

## Laburdurak

---

---

Laburdurak	Esanahia
ADS	Automation Device Specification.
AGV	Automated Guided Vehicle.
AMR	Autonomous Mobile Robot.
BBox	Bounding Box
BDCM	Brushless Direct Current Motor.
Bps	Bits per second
CNN	Convolutional Neural Network
COM	Serial Communication Port
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture
DP	Deep Learning
EKF	Extended Kalman Filter

Fps	Frames per second
GPU	Graphics processing unit
GSD	General Station Description
HMI	Human Machine Interface
ICP	Iterative Closed Point
IPC	Industrial Personal Computer
LR	Linear Regression
LIDAR	Light Detection and Ranging
ML	Machine Learning
PF	Particle Filter
PLC	Programable Logic Controller
RPM	Revolution per minute
SLAM	Simultaneous Localization and Mapping
SQDCM	Safety, Quality, Delivery, Cost and Morale

---

# I.

# Atalburua

Gaur egungo zientzia, bihar egungo teknologia -Edward Teller-

## Sarrera

---

---

Tesi honetzako Euskal Herriko Unibertsitateak eta Mercedes – Benz Gasteizko lantegiak hitzarmen batean bildu dira, gaur egungo zientzia teknologia bilakatzeko. Konkrétuki fabrikazio prozesuen efizientziak hobetzeko, robotika kolaboratibo autogidatua erabiliz. Berez, proiektua garatzeko bi teknologi desberdinak erabili behar dira, alde batetik robot kolaboratiboa eta bestetik plataforma mugikorra. Robotika kolaboratiboak lankideak arriskuan ez jartzeko sistema ezberdinak erabiltzen dituzte. Horrek, edonork robot inguruan lan egiteko aukera ematen du. Horregatik, dispositibo hauek robot industrial konbentzionaletik bereizten dira. Sistema autogidatua lorrarazteko, robota plataforma mugikor batean atxiki behar da. Robot mugikor hauek era autonomo batean lantegi inguruan zehar mugitu daitezke, elementuen garraioa era eraginkor batean egitez. Bi teknologia hauek lotuta, enpresa barneko lan postu batean ezarri nahi da, non muntaketa prozesuak etengabeko mugimenduan dauden. Proiektuaren eragina jakiteko, automobilgintzan emandako roboten historia jakin behar da.

Duela berrogeita hamar urte, edozein automobilgintza sektorean, robotak piezen soldaketetan eta manipulazioetan erabiltzen hasi ziren, prozesu egonkorragoak, seguruagoak eta errepikakorragoak sortzeko. Pixkanaka-pixkanaka “*Body Assembly*” edo egitura muntaketa guneak automatizatzen joan ziren, enpresaren produkzio denborak eta kostuak hobetuz.

Gaur egungo zientziak, robotika kolaboratiboaren teknologia eskaintzen du. Non, elementu teknologiko hauei esker, “*Final assembly*” guneak automatizatzeko aukera ematen duten, lan postuetan gizaki - robot lan kolaboratiboa sortuz. Tesi honen garapenarekin, lan postuen ergonomia hobetzeko asmoa dago, robotak langileren esfortzuak garatuko dituelako. Prozesu kolaboratiboa sustatuz, bai lankidea, bai robota postu berean lan egin dezakete, baina lan esfortzuak haien kapazitate arabera banatzen dira.

Robot (robot kolaboratiboa + plataforma mugikorra) honen garapena lortzeko, bai Mercedes – Benz barneko errekurtsoak, baita Gasteizko Ingeniaritza Eskola dituen baliabideak erabiliko dira. Edozein tresneriaren garapena egiteko, elementu komertzialak eta industrialak erabili beharko dira, batez be lantegi barneko mantentze estandarrak mantentzeko. Baliteke gune batean elementu ez industrialak erabiltzera behartuta egotea eta kasu horretan teknologia horren beharra balioztatuko da.

Lehengo atalburu honetan, proiektuaren nondik-norako informazioa bilduko da, bai zientziaren, bai teknologiaren egoerak azalduz. Dena den, Mercedes – Benz Gasteizko lantegiaren historia ezagutzea ezin bestekoa da, Gasteizko ekonomiarentzako oso garrantzitsua delako eta momentu honetan tesiaren garapenari emandako aukera eskertu behar delako.

## 1 Mercedes – Benz Gasteizko Lantegia

---

1950-ko azaroaren 17-an, bost milioiko kapitalarekin IMOSA lehengo administrazio kontseilua sortu zen. Handik hiru urte geroago automobilgintzarako lantegia eraiki zuten Gasteizen. Hau da, duela hirurogeita hamar urte tesi hau burutzeko gunea sortu zen. Non gaur egungo zientziaren eboluzioagatik eta lantegiaren garapenarengatik, proiektu honen hasiera 2018-garren urtean eman zen. Atal honetan bildutako informazioa eta argazkiak Patrimonio Industrial Vasco webgunetik lortu dira {1}.

Lehengo kontseilutik lau urte igaro behar dira lehengo furgonetaren fabrikazioa ikusteko, konkretuki 1954-ko martxoaren 20-rarte. IMOSA taldekideen artean DKW F89L modeloa fabrikatu nahi zutela adostu zuten. Modelo honek 2 zilindrotako eta 700 cc-ko motorra erabiltzen du. Motor horrek 22 zaldi potentzia ematen ditu eta 750 Kg lan zamako kargak eramateko aukera ematen du.



Irudia 1 DKW F89L eredua, bere azpian Gasteizko lantegiaren fabrikazio prozesua.

Automobilgintzaren fabrikazioa aurrera zihoan eta lantegi barnean eraginak ekarri zituen. Adibidez, 1957-an furgonetaren eskaeraren igoeraren ondorioz, lantegian lehengo mugaketak ikusi ziren. Limite horiek gainditzeko, autoen egiturak fabrikatzeko gune berezi bat behar zutela antzeman zuten. Gamarra-Betoño poligono industrialean, Industrias Auxiliaren del Automóvil S.A. sortu zen, mugapen horiek gainditzeko. Enpresa honekin bat, beste zenbait langune sortu ziren Gasteizen zehar.

Urteak pasa hala, enpresarentzako aurrerapen gehiago ekarri ziren, 1963-garen urtean bi meritu lortuz. Lehengo meritua lankide gehiagoren kontratazioetatik dator. Kasu honetan, 1.841 pertsona kontratatu zituzten, furgonetaren demandari aurre egiteko. Bigarrena aldiz, lehengo fabrikazio kontratu eskusiboaren lorpenetik dator. Gasteizko lantegiak DKW F 1000 furgoneta diseinatu, fabrikatu eta mundu osorako saltzeko aukera izan zuen. Urte bat geroago, automobil honi Mercedes – Benz-eko diesel motorrak ezarri ahal zitzaizkion eta DKW F 1000D bezala ezagutzen da. Diesel motorrak autonomia eta indar pare handiagoak ekarri zituen.





Irudia 2 DKW F 1000-ren aldizkari bateko argazkia

Modelo honen salmenta arrakastari esker, enpresaren kapitala aldatzen joan zen eta 1972-an Mercedes – Benz eta Volkswagen batu egin ziren, Compañía Hispano Alemana de Productos Mercedes – Benz y Volkswagen, SA. MEVOSA sortuz. Horrek, enpresaren % 55-ren kapitalak Mercedes – Benz eta Volkswagen artean zati berdinetan banatzea eragin zuen. Hiru urte geroago DKW furgonetak alde batera utzi zituzten, Mercedes – Benz N 1000 fabrikatzeko. Furgoneta berri honekin, lantegiaren berrantolaketa etorri zen. Bartzelonara mekanizazio prozesuak lekualdatu ziren eta Gasteizen soilik furgonetaren muntaketak burutuko ziren.

Izandako arrakastei ezker, lantegiaren kapitala handitzen joan zen eta 1980 -garren urtean Daimler taldeak enpresaren % 52,13 kapitala lortu zuen. Bederatzi urte geroago Daimler taldeak Gasteizko lantegiaren jabetza lortu zuen eta laurogeita hamargarren hamarkadan M-B 100D modeloa aurkeztu zuten, N 1000 modeloa alde batera utziz. 1995-garren urtean, egun hauetan ezagutzen den modeloa kaleratu zuten, hau da, Vito izena duen furgoneta. Izen honek Vitoria hiritik dator, baina SEAT garai horretarako estatuko hiri eta herri guztien izenak erositatu zituen, beraz Mercedes-ekoek izena txikitzea erabaki zuten. Dena den furgoneta honi erosotasuna handitzeko aukera ikusi zuten eta Vito furgonetaren egitura mantenduz V-Class modeloa aurkeztu zuten 2003-an.

Gaur egun, Gasteizen hiru modelo desberdin fabrikatzen dira. Vito, V eta EQV besteak beste. Motorren aldetik aukera ezberdinak daude, laneko Vito furgoneta Diesel motorra edo eragile elektrikoa atxiki ahal zaio. V-Class barnean baita furgoneta elektriko bat dago, EQV izenekoa (ikusi Irudia 3c). Tesiaren aurkezpena egiten den unean, Gasteizko lantegiak 160.000 inguruko furgonetak fabrikatzen ditu eta Euskal Herriko BPG-ren % 5-a suposatzen du. Horrekin batera, 5.500 lankide enpresak kontratatuta ditu eta beste 12.000 lan postu ez zuzenak sortzen ditu.



(a)

(b)

(c)

Irudia 3 Mercedes - Benz furgoneta desberdinak a) Vito modeloa b) V-Class c) EQV

## 2 Lantegiaren antolaketa

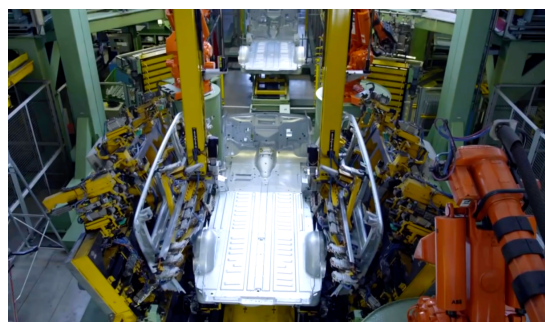
Gasteizko lantegian fabrikazio prozesuak hiru gune ezberdinetan antolatuta daude, tesia aurkezten den aldiunean. Gune bakoitzaren zehar lan-postuen egoera baldintzak aldatzen dira eta hauen arabera, bai pertsonen, baita roboten beharra ere.

Lehenengo guena “*body assembly*” edo egituren muntaketa da, non furgonetaren gorputzaren soldaketa egiten den. Berez, gorputza fabrikatzeko prozedura batzuen beharra dago, baina hauek lantegiaren historian azaldu den moduan, Bartzelonan egiten dira. Soldatze prozesu hauek era automatiko baten bidez egiten dira, lantegiak SQDCM filosofia mantenduz. Gainera, Mercedes – Benz-eko kalitate estandarrak betetzeko, hainbat postutan kalitate kontrolak ezartzen dira, hauek era automatikoan edota langileen bidez izan daitezke. Berez, kalitate estandarrak eta fabrikazio filosofia, enpresa osoan zehar mantentzen da, Mercedes – Benz duen prestigioa mantentzeko.

“*Body assembly*” guneak, robot teknologiarri esker, lankideen lan egoera hobetu egin dira, prozesu seguruagoak eta eraginkorragoak lortuz. Makinaria hauek segurtasunaz gain, kalitate eta errepikakortasun hobekuntzak dakartzate lan postuentzako. Irudia 4-an bi urte desberdinetako fabrikazio prozesuak agertzen dira. Ezkerraldeko kasuan, soldaketa prozedurak eskuko lan bitartez burutzen ziren. Aldiz, eskuinaldeko kasuan, era automatiko batez furgonetaren gorputza soldatzen da. Eragiketa bat era automatiko batean egiteko eta SQDCM filosofia mantentzeko, hainbat mekanismoen beharra dago. Alde batetik, robotak eskuz programatu behar dira, soldatzaile profesional baten ezagutzaz baliatuz. Horrekin bat, kotxearen edo furgonetaren metalezko piezak posizio konkretu batean eutsi eta mantendu behar dira. Horren ondorioz, euskarri berezien diseinuaren beharra dago. Fabrikaziorako balio erantzia sortzen ez dituzten zenbait elementu behar dira, batez be segurtasun elementuak ezarri behar direlako, gizakia eta makinaren lan eremuak bereizteko. Kapitulu honen sarreran aipatu den moduan, robot industrialak kolaboratiboetatik segurtasun elementuengatik bereiz daiteke. Robot kolaboratiboek integratutako sentsoreei esker, zenbait hezi edo bestelako segurtasun elementuen beharra ez daukate. Segurtasun elementu gehigarrien beharrak, diru gehiagoren inbertsioa erakartzen du, fabrikazio prozesuaren automatizaketarako. Dena den, segurtasun guneak robotaren jarrera aldatzeko erabiltzen dira. Langile batek gune automatiko batean sartzerako orduan, fabrikazio prozesuek jarrera konkretu eta seguru bat adosten dute.



(a)



(b)

Irudia 4 Fabrikazio prozesu desberdinak urteetan zehar a) Eskuzko soldadura prozesua (1968)(Old Benz webgunetik lorturiko argazkia) b) Soldaketa automatikoa (2015) (Fabricando Made in Spain-etik lorturiko Argazkia)

Behin furgonetaren gorputza eraikita izanda eta honek kalitate estandarrak gaintuta, gorputzak gunez aldatzen da. Bigarren eraikuntza honetan, furgonetari bezeroak eskatu duen kolorea aplikatuko zaio. Margotze prozedura hasi baino lehen, bezeroak hautatzen ez dituen zenbait produktu aplikatzen zaizkio metalezko gorputzari. Adibidez, gorputzak soldaketa prozesuan hainbat zikinkeriak hartzen ditu eta hauek garbitu behar dira. Bati-bat olioak,

taladrina, etab. Horregatik, margotze prozesuaren kalitatea ez murrizteko, edozein tratamendu jaso baino lehen, karrozeriak garbitze prozesu batetik pasatu behar da.

Metalezko gorputza oxidoen aurkako babesak behar ditu ere, furgonetak beste zenbait etapatik pasaz. Guztira 5 etapa jasan behar ditu, hau da, hiru etapa kolore aurreko produktuak dira. Produktu hauen izaeraren arabera, furgonetak tratamendu termiko batetik igaro behar dira. Gorputzaren eraikitze prozeduran bezala, gune honetan urteak igaro hala, lan eremu ezberdinak automatizatu egin dira. Dena den, margotze prozeduren automatizaketak, langileak produktu toxikoren manipulaziotik aldentzen ditu, lankideen osasuna hobetuz. Horrekin bat, furgonetaren gorputzak jasan behar dituen tratamendu ezberdinak optimizatu egiten dira, furgoneta guztiek produktu kantitate berbera eramaten dutelako.

Behin kalitate ezaugarriak pasata, gorputza hirugarren eraikuntzara pasa daiteke. Dena den, margotze eta "*final assembly*" eraikuntzen artean bitarteko eremu bat dago. Gune berezi honek biltegi automatizatu bat ezkututzen du, non margotutako karrozeriak gordetzen diren. Biltegi honek, eraikuntzen arteko birikia deritzote, bi eraikuntza arteko fabrikazio denbora bereizteko. Horrela, "*final assembly*" eraikuntzan fabrikazio arazoak izanez gero, margotze prozesuak ez dira eteten. Azkeneko eraikuntza honetan, furgoneta kaleratzeko behar dituen ekipamenduak muntatzen dira. Horrek ergonomia aldetik lan baldintza egokienak ez izatea eragiten du. Adibidez, furgoneta barnean egiten diren muntaketetan, lankideak gorputzari jarrera txarra izatera behartzen dio. Halako egoerak ekiditeko, zenbait postuetan lankideen parte hartzea handitu behar da, fabrikazio kostuak handituz.

Lan egoera hauek ikusita, tesi honek dituen helburuen artean, lankideen erosotasuna hobetzea aurkitu daiteke. Azken finean, beste eraikuntzetan egindako automatizaketa prozesuak, "*final assembly*"-ra erakarri nahi dira, baina gaur egungo teknologia erabiliz. Hau da, robotika kolaboratiboa ezarriz.

### 3 Mercedes – Benz-eko ikerketa guneak

---

Gasteizko lantegian lan eremu berri eta berezi batzuk prestatu dira, bati-bat ideia berriak aurrea ateratzeko. Berez, laborategiak unibertsitate baterako ohiko guneak dira, baina automobilgintzan lantegientzako eremu berritzaileak dira. Bertan lantegirako proposatzen diren berrikuntza guztiak pentsatu, eraiki, frogatu eta lankideak formatzeko erabiltzen diren espazioak dira. Garapen eremu horiek, tesi honetarako gune estrategiko bat izan da, fabrikazio prozesu baten simulazioak egiteko espazioa egon delako. Simulazio hauek, lantegiko lan baldintza berberekin egin dira, hau da, laborategian furgoneta gorputz ezberdinak egon dira eta gorputz hauek mugiarazteko, linea elektrikoa daukate. Baita furgonetak altuera ezberdinetan ezar daitezke ere, horrela lan postu guztien baldintzak errepikatu daitezke. Naiz eta proiektu honetarako baliabidea hauek simulazioak egiteko erabili izan diren, normalean formazioak egiteko erabiltzen dira. Berez, laborategia bi multzotan bereiz daiteke, alde batetik lankideentzako eremuak eta bestetik prototipoen garapenarentzako guneak. Lehengo multzoaren barnean, bi gune bereiz daitezke, bata ergonomiakoa eta bestea formakuntzakoa. Bigarren multzoan, aldiz, "*think-tank*" eta froga-gunea bereiz daitezke.

Ergonomia aldetik, lankidearen gorputzaren erosotasuna lantzen da. Berez, ergonomia bi erataria landu daiteke. Alde batetik, lankideari bere gorputza nola erabili behar duen irakatsiz, batez be lesioak ekiditeko. Beste aldetik, lanpostu bakoitzaren lan zamak aztertuz, lan postuaren ebaluazio bat eginez. Parametro hauei esker, zenbat lankide kopurua behar den zehazten da. Azken finean lankideak zaindu behar dira, beraiek furgonetak eraikitzen dituztenak direlako. Ergonomi alde batera utzita, pertsonen jakin-mina handitzeko formazio guneak daude. Horrek, lantegi barnean, gaitasunak hobetzen eta tartekatzen laguntzen du, lankide bakoitzaren konfiantza handituz. SQDCM-ko filosofia aplikatu nahi ezker, gizaki bakoitzean pentsatu behar da, amaierako produktuaren kalitatea hobetzeko elementu automatikoaz gain, pertsonen formazioak laguntzen dituzte ere.

Prototipoen garapena burutzeko, lehengo pausua asmakuntzari bere espazioa uztea da eta horretarako *“think-tank”* gunea dago. Bertan langileak behar besteko erosotasunak dituzte, ideia berriei askatasuna emateko. Horrekin bat, mantentze lan prediktiboak kontrolatzeko gela dago, ezinbesteko erabakiak hartzeko eremu aproposa delako.

Atal honekin bukatzeko, laborategia edo frogua guena dago. Izenak adierazten duen moduan, ekoizpen prozesuan sartu aurretik makinariaren azterketak egiteko gunea da. Naiz eta gune hau diru inbertsio bat izan, egin daitezkeen frogapenei esker, produkzio mailan dirua aurreztu daiteke. Gune hau tesiaren garapena egiteko ezinbestekoa izan da eta honen aurrerakuntzen balorazioak egiteko ere. Azken finean, frogapenei ezker fabrikazioaren kalitatea balioztatu daiteke ekoizpen barnean sartu baino lehen eta aurre ikusi sarturiko hobekuntzak prozesuaren kalitatea handitu edo bermatu egiten duten.

## 4 Mercedes – Benz-eko lan filosofia

---

Mercedes – Benz SQDCM fabrikazio filosofia jarraitzen du. Hau da, Safety, Quality, Delivery, Cost and Morale atalak tratatzen ditu, haren fabrikazio baldintzak hobetzeko. Beraz, tesian burutu nahi diren garapenak, filosofia hau jarraituko beharko luke eta proiektuak ekar ditzakeen etekinak, SQDCM lantzen dituen atalei lotuta egongo dira. SQDCM-ren filosofia *“Lean Manufacturing”*-enetik dator, berez neurriko produkzioari lotuta dago. Metodologia honek, fabrikazio prozesuen hobekuntzetan oinarrituta dago, produktuari balio oinarri gehitzen ez duten prozesuak edo lanak murriztuz.

Ingelesetik *“lean”* datorren hitza, gure hizkuntzan hainbat sinonimo baititu, baina azken finean, metodo honek prozesuren eraginkortasun eza lortzea helburu bakarra dauka, lan kostu minimoak lortuz. Halako murrizketak egiteko, *“Lean Manufacturing”* erreminta desberdinak aurkezten ditu, prozesuaren aktibitate bakoitzari prozesu efizienteena aplikatzeko, langilearen lan baldintzaz ahaztu gabe. Beraz, lan partaideen motibazioa kontuan hartu behar da, kalitatezko produktu bat lortzeko. Metodologia honen ezaugarriak honako hauek dira:

- **Lehengo aldiuneko kalitatea lortzea:** Akats hutseko produkzioa planteatu behar da, non problema baten haurrean, haren jatorria detektatu eta zuzendu egiten den.
- **Zarrastelkeria murriztu:** Baliabideak (Kapitala, lankideak eta lan-eremua) optimizatuz, balio oinarri gehitzen ez duten aktibitateak gutxitu egiten dira.
- **Etengabeko hobekuntza:** Kostuen murriztu, kalitatearen hobetu, produktibitatearen areagotu eta prozesuen informazioaren tartekatu.
- **“Pull” edo tiratze lan prozedura:** Amaierako produktua bezeroak tira egiten baitituzte, eta ez enpresek. Hau da, bezeroak produkzio egoerak ezartzen ditu.
- **Malgutasuna:** Produktu ezberdinak ekoiztuz, efizientziak mantenduz, naiz eta produkzio bolumen ezberdinak izan.
- **Hornitzaileekin harreman sendoa** eraiki, arriskuak, kostuak eta informazioa tartekatuz, hitzarmen ezberdinak osatzerakoan.

Proiektuaren etekinak balioztatzeko, *“Lean Manufacturing”* proposatzen dituen aholkuak jarraitu egin dira. Proiektu honen garapenaren ondorioz, lau etekin nagusi kontsideratu dira. Lehengoa, kalitate inguruari egiten dio erreferentzia, non robot kolaboratibo autogidatuarekin **lehengo aldiuneko kalitatea lortu** nahi den. Etekin hau, edozein robotarekin lor daiteke, bati-bat haien ezaugarrien artean errepikakortasun handiko elementu mekanikoa delako. Lankideak aldiz, denbora igaro hala ezaugarri hau alda daiteke, bai lan-baldintzengatik, baita kontzentrazio galeragatik. Horregatik, tesi barnean garatutako den robotaren laguntzaren ondorioz, lan prozesuaren errepikakortasuna handitu daiteke, aldi berean muntaketa kalitatea hobetuz. Horrekin batera, **prozesu seguruak, egonkorak eta sendoak** lor daitezke eta horrek kalitatearekin zerikusi zuzena ez duten prozesuen kostuak murriztu daiteke. Adibidez, bihurkin

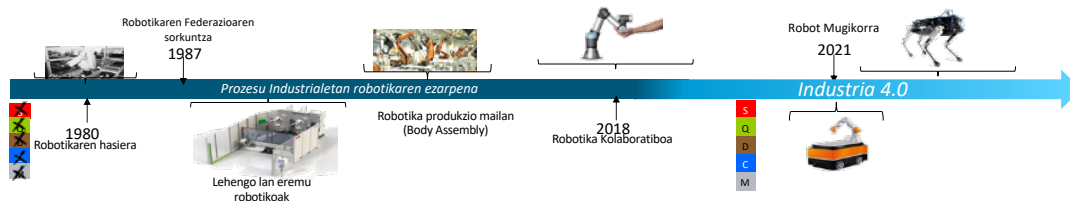
prozesu batean indar-pare konkretu batzuk aplikatu behar dira eta lankidearen gorputzak indar horiek jasan behar dituzte, inpaktuaren arabera eragin ezberdinak sortuz. Ergonomia aldetik, lan postu horiek, lanpostu txarrak kontsideratuko dira. Erosotasun galera hori konpentsatzeko, langune horietan pertsonak bikoiztuz edo hirukoiztuz zuzentzen dira, lanpostuaren kostuak handituz eta produktuari balio gordinik ez erantziz. Proiektuan proposatzen den mekanismoarekin, indar-pareak elementu mekanikoak jasango ditu, lan postu horren gaitasunak desagertuz. **Prozedura honekin gizaki-robot kolaborazioa** sor daiteke, non pertsonak haren inteligentzia eta ustekabeak tratatzeko gaitasunak ezartzen dituzten eta robotak haren errepikakortasun kapazitatea konpartitzen du prozedurarekin, produktibitate hobekuntza erakarritz.

Bigarrena, **absentismoaren inguruko arazoekin** bat dator. Halako aspektuekin enpresari gaitasunak sortzen dizkio, produktuen fabrikatze balioa handitzen dituelako. Horretarako, lanpostuaren erosotasuna edota ergonomia handituz halako jarrerak zuzendu daitezke, langileren konforta handitu delako. Horrekin bat, lanpostuaren eskuragarritasuna hobetu daiteke eta desgaitasunak dituzten pertsonak halako postuetara hautatu daitezke. Gainera, aldaketa demografikoa dela eta, halako lan baldintza hobetuta dituzten postuak, edozein adinako pertsonak aukeratu daitezke.

Proiektuaren etekinak laburbilduz, lanpostuaren efizientziak eta kalitatea landuz, SQDCM atal guztiak hobetu daitezke. Kontuan izanda bai enpresarako, baita lankideentzako etekinak lortzeko aukerak daudelako.

## 5 Tesiaren garapena

Behin ikusita lantegiaren filosofiak, tesiaren zer burutuko den aipatuko da. Berez, industria aldetik, efizientziak sortzen dituen Industria 4.0-ko robot kolaboratibo autogidatu bat eraiki nahi da. Horretarako inteligentzia eta ekipamendua ezberdinak ikertu, garatu eta prozesu batean ezarri behar dira. Irudia 5-ak robotikaren egoera aurkezten duen kronograma irudikatzen du. Hasiera batean SQDCM filosofia industria barnean ez zegoen aplikatuta. Roboten erakarpenekin eta prozesu esanguratsuak automatizatuz lan baldintzak hobetzen dira. 2018 tesia ematen da eta urte horretan lantegian lehengo robot kolaboratiboak ezartzen dira. Hortik aurrera egiten diren ekarpenak tesiarekin erlazionatuta egongo dira.



Irudia 5 Robotikaren kronograma

Horretarako proiektu batek, hainbat erronka gaitu behar ditu, ezarritako helburuak betetzeko. Lehenik eta behin, robotak egin behar dituen **mugimenduak zehatzak** izan behar dira, kalitatezko prozedura egiteko. Bigarrena, robotak lan-prozesuek dituzten **abiadura ezberdinak jarraitu** behar ditu, furgoneta dituen hainbat **erreferentzia gunez bilatuz**. Hirugarrena, lantegiko **beste ekipamenduekin integratuta** egon behar da, fabrikazio informazioa tartekatuz. Azkenik, robotak egin behar dituen prozesuak **seguruak, ziurtatuak eta egonkorrak** izan behar dira. Helburu hauek betetzeko, proiektuan zehar hainbat fase ezberdinak planteatu dira, proiektuaren jasagarritasuna balioztatuz. Honako hauek omen dira:

- Proiektua zehatz mehatz definitu
- Erreferentzia sistema kolaboratiboak landu era estatikoan
- Etengabeko laneko inteligentzia garatu
- Produktzorako behar diren lan egoerak tratatu
- Industrializatu
- Tesiaren ondorioak.

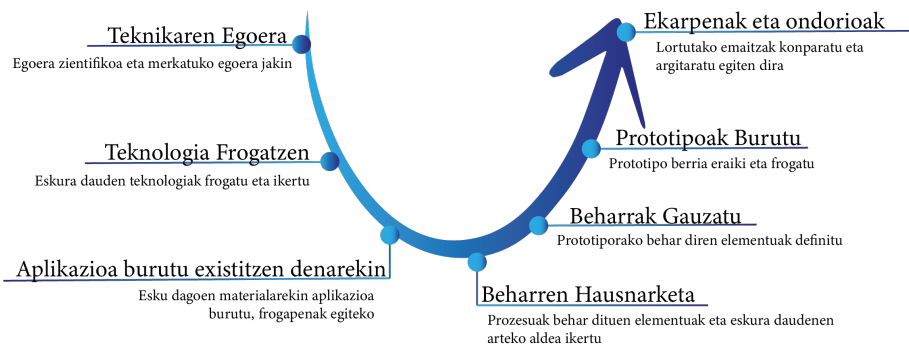
Horrekin bat faseak garatzeko lan filosofia konkretu bat erabili da eta liburu honetan egingandako aktibitate ezberdinak azaltzeko oinarri bezala erabili da. Non Atalburu bakoitzak pentsamendu honi lotuta dagoen. Filosofia horri esker, fase bakoitzak prototipo bat aurkezten du. Horrela lantegiko amaierako muntaketaren zuzendariari, Jose Carlos Velasco, era erraz batean erakusketa bat egin daiteke.

## 6 Proiektuaren garapen filosofia

Aurreko atalean aipatu den moduan, lan filosofia konkretu bat jarraitu da, batez be proiektuaren bideragarritasuna kontrolatzeko. Horretarako, Otto Scharmer-en U teoriaren filosofia adostu eta moldatu da, fase bakoitzean aplikatzeko. Jatorriz, lan eredu honek pertsonen pentsamendua aldatzeko prestatuta dago, baina ideia nagusiak erabiliz edozein proiektu burutzeko moldatu daiteke. Tesian ezarritako moldaketak Mercedes – Benz-eko lantegietan dituzten beste zenbait proiektuetan U teoria oinarri bezala ezarri da.



(a)



(b)

Irudia 6 U teoriaren garapena: a) Otto Scharmer aurkeztutako filosofia (Presencing Institute-ko argazkia), b) Tesian erabilitako metodologia, fase bakoitza garatzeko

Dena den, Irudia 6a adierazten dituen oinarriak Irudia 6b-en mantendu dira, bestela U teoriaren filosofia jarraituko ez litzakete. Hasierako pausuak informazioa biltzeko prestatu dira. Ondoren informazioa tratatu eta aztertu egiten da, behar diren beharrak aztertzeko. Lorturiko behar horiek ustiatu egiten dira, azkenik prototipo bat osatzeko. Behi prozesu hau jasanda, lorturiko ondorio guztiak biltzen dira.

Azken finean, teoria hau behin da berriz bizitzan, unibertsitatean, etab. aplikatzen da. Artikulu baten argitalpena egiterako orduan, lehengo etapa informazioa biltzea izaten da. Bildutako informazio aztertuz eta ulertuz, hobekuntza edo berritze puntuak zeintzuk diren ikusten dira eta horiek prototipo eraikitzaera eramaten du. Prototipo horrek ondorioak eraikitze baliagarria izango da eta azkeneko etapari, lortutako ondorioak bildu eta mundura zabaltzen dira, artikuluen bitartez.

Pentsatu daiteke filosofia hau barneratuta dagoela eta era automatikoki egiten dela, baina normalean, gizakiak zuzenean prototipoak eraikitzaera joaten dira. Horrek lan epeetan eraginak ekartzen ditu. Batez ere, fase bakoitzaren atalak errespetatzen ez baldin badira, baliteke prototipoa aurrera ez ateratzea, behar den informazioa ez delako bildu edota lan baldintzak ez direlako ezarri.

Gerta daiteke Mercedes – Benz barnean, Jose Carlos Velasco-ri edota Emilio Titos-ri aurkezpenak egiterako orduan, prototipo baten aurrerapena prestatuta ez izatea. Baina U teoriarekin, modu erraz batean proiektuaren egoera azal daiteke. Horrekin bat, prototipoaren beharrak definitzerako orduan, baliteke elementu berriak erosi behar direla eta horrelako atal bat izatea teoria barnean, goi-kargudun pertsonak behar diren eskaerak baieztatzeko laguntzen du. Horrekin bat, bai Innobide coworking enpresa, baita Mercedes – Benz Estatu Batuetako lantegian filosofia hau jarraituko dutela aipatzekoa da.





# II.

## Atalburua

Gauza berriak ikasteko edozein unea perfektua da -Albert Einstein-

### Teknikaren egoera

---

---

Proiektuaren filosofia eta faseak ezagututa, kapitulu honetan U metodologiaren lehengo atala tratatuko da, non proiektuaren ekarpenak definituko diren. Horretarako, bai aurrekari zientifikoak, baita fabrikatzaileen egoera zein den jakin behar da. Kasu gehienetan, unibertsitate arloko proiektuak, argitalpen bitartez tesiaren helburuak eta egin beharrak finkatzen dira, baina garapen honek Mercedes-Benz Gasteizko lantegian egin denez, salmenta egoera zein den kontrolatu behar da. Halako proiektuen garapena egiteko, robotika inguruan zer dagoen jakitea ezin bestekoa da. Zure produktua lortzeko besteen esperientziatik ikas daitekeelako, edota eraiki nahi den prototipoa beste enpresa batek garatu izan balu, haien produktua erabiliko litzateke tesia aurrera ateratzeko. Bill Gates esan zuen moduan, informazioa izatea boterea dakar.

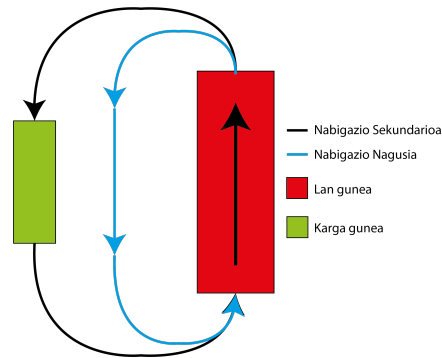
Lantegien informazio bilketa lortzeak zailtasunak ditu. Azken finean, saltzailearen helburua nagusia haien produktua eskaintzea da, baina haien teknologia erabiltzerako orduan, hainbat ustekabeekin topatzeko aukerak baitaude. Horrek merkatuaren egoera zein den jakiteari trabak ezartzen ditu. Halako egoerak saihesteko, hirugarren atalburuan aipatzen den moduan, Mercedes-Benz utzitako lan-eremua laborategi barnean eta unibertsitate barnean proposatutako proiektuak funtsezkoak izan dira.

Egin nahi den muntaketa eta lan prozedura Irudia 7-an labur daiteke. Non, Irudia 7a-n 4.0 industria barneko bi tresneria biltzen dituen muntaketaren errepresentazioa egiten da. Bestetik, Irudia 7b-n ibilbidearen eta lan espazioak definitzen duen mapa agertzen da.

Muntaketan erabili nahi diren tresnerien artean, Unibersal Robot kolaboratiboa eta Kivnon-en plataforma mugikorra edota "*autonomous mobile robot*" bat aurkitzen dira. Tesia hasi denean bi erreminta fabrikatzaile hauek halako muntaketarik ez dute planteatu, baina denbora igaro hala, 2019-an Kivnon-eko burua tesiaren ideia ikusi bezain laster, merkatuarentzako proposamen bat atera zuten. Dena den, dokumentu honetan planteatzen diren lan baldintzen ondorioz, merkatuko muntaketa guztiak baztertzen dira.



(a)



(b)

Irudia 7 Muntaketaren ereduak: a) Bi roboten muntaketa ereduak, b) Lan eremuaren definizioa

Lan eremua irudikapena egin ostean, argi eta garbi ikus daiteke bi ibilbide mota planteatzen direla. Alde batetik, nabigazio nagusia. Bertan AMR-ak lan eremutik karga gunera edo alderantziz mugiarazten da. Bestetik, nabigazio sekundarioa. AMR-aren bateriak karga nahikoa baldin badu, bigarren ibilbide hau erabiliko du, beste kotxe batean lanean jarraitzeko.

Bi plataformen inteligentziak aztertuta, AMR-ek duen barne nabigaziorako tresneria suspertu egin behar dela determinatu da. Aldiz, robotaren kasuan, haren posizionamendua irudietan oinarrituko denez, robotaren kontroletik at geldiarazten da. Segurtasun atalak lantzen baldin badira, orduan robotaren kontrolean aldaketak egin beharko liriateke. Adibidez, robotak geldiaraztea talka egin gabe.

Berez, gidaketa autonomoaren inteligentziaren garapena oinarritzkoa da, ezarritako lan baldintzetan AMR-ren jarrera ulertarazteko. Horregatik, atal honetan, barne nabigaziorako informazio bilketa zabalagoa egin da.

## 1 Gidaketa autonomoa

Tesia garatu den unean, gidaketa autonomoaren arloa gorakada doa. Ibilgailuaren lan baldintzen egoeraren arabera, hau da, barne edo kanpoko gidaketaren arabera, zientziak teknologia desberdinak behar ditu, funtzio berbera egiteko. Kanpo eremuetan inteligentziaren garapenak aurrerapen gehiago ditu.

Automobilgintzan legedien arabera, errepideak elementu arautuak bilakatu dira, erabat normalizatuak egonez. Horrek algoritmoen garapenetan laguntzen du. Berez, errepidearen zuzentasuna jakiteko, errepideko lerro zuriak jarraitu behar dira eta abiadura seinale bitartez irakurri daiteke. Araudiaz gain, zenbait abantaila teknologiko daude. Adibidez, Global Position System (GPS), non sateliteei esker, munduko edozein posizioa jakin daiteken. Bestalde, mapak sortarazi egin dira edozein pertsona ibilbide jarraitzeko aukera izateko.

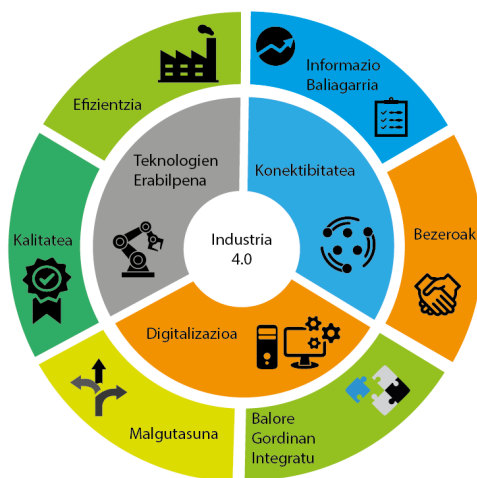
Kanpoko gidaketa autonomoak interes gehiago suspertzen duela argi eta garbi ikus daiteke. Ikerketa egiterako orduan, aurkitzen diren artikuluen artean 1995 –eko informazioa topa daiteke. Horrekin bat, Vicsek et al. [1] gidaketa autonomoaren trantsizio faseak aurkeztu zituen eta fase horiek erreferentzia modura har daiteke. Eredu onetan badaude argibide bezala ar daitezkeen beste autore batzuk. Horien artean Helbing et al. [2], Fangnant et al. [3] eta beste zenbait argitalpen berriagoak [4,5]. Hala ere, errepideetan gidatzeko inteligentziatik hain bat ondorio lor daitezke, barne gidaketarentzako. Bati-bat erabiltzen diren [4,5] zientzia arloko garapenak. Adibidez, irudien tratamenduak, irudien sailkapena sare neuronalekin, etab. aurkitu daiteke [6-10].

Barne gidaketari buruz hitz egiterako orduan, elementu arautuak desagertzen dira. Eraikuntzetan errepide marrak, seinaleak, etab. ez daudelako. Horren ondorioz, garatu behar diren algoritmoak informazioa beste leku batzuetatik lortzera behartzen du. Edozein gidaketa mota egiteko, arazoak multzoka eta atalez atal lantzen dira. Barne gidaketan topatu

daitezkeenak honako hauek dira. Lehendabizi, mapa eraiki behar da. Ondoren mapa horren bitartez, autoaren ibilbidea markatu egin behar da. Azkenik, plataforma mugitzen den bitartez, lokalizatu egin behar da eta zenbait kasutan maparen berritzea egiten da, eremu dinamikoak direlako. Yassin et al. [11] gidaketa autonomoari buruzko informazio bilketa planteatu zuen artikulu batean, momentu honetan aurki daitezkeen egoera azaltzeko. Hala eta be, gidaketa autonomotik at dauden beste arloen egoera jakin behar da, zientzia aurkezten dituen baliabideak sustatzeko [12,13]. Behin zientziaren arloaren sarreraren egoera ikusita, merkatuaren egoera zein den jakin behar da, edozein informazio bilketa jarraitu haurretik. Horrela, proiektuaren erakarpen zientifikoak teknologiarekin bateratzen dira.

## 2 Industria barneko ibilgailuak

Proiektuaren garapena egiteko, merkatuaren bi arlo ulertarazi egin behar dira. Lehendabizi barneko eremuen ibilgailuak dira eta atal honetan aipatuko dira. Beste arloan robot kolaboratiboak landuko dira. Bi elementu hauek, Industria 4.0-ren automatizazioaren garapenerako ezin bestekoak dira. Dena den, gaur egungo industria beste zenbait arloen menpe dago, Lu [14] adierazten duen moduan. Irudia 8-an laburbildu daiteke Industria 4.0 tratatzen diren arlo guztiak. Garapen honetarako erabili diren dispositiboak, produkzio teknologikoekin harreman zuzena daukatelako. Proiektuaren helburuetan ikusi den moduan, badaude beste zenbait arlo era ez zuzen batean tratatuko direnak. Bestak beste: kalitatea, malgutasuna eta digitalizazioa batez be.



Irudia 8 Industria 4.0 arloan finkatzen diren pausuak

Industria berri honetan parte hartzeko eta tesiaren garapena egiteko, barne ibilgailuaren gidaketa autonomoaren garapenaren inteligentziak ikertuko dira. Lantegian bi izaera ezberdineko ibilgailu bereiz daitezke, nabigatzerako orduan dituzten kapazitateen arabera. Lehengoak Automated Guided Vehicles (AGV) dira eta zinta magnetikoen bitartez mugitu daitezke edozein eremutik, lokalizazioa erabili gabe. Bigarrenaren kasuan, Autonomous Mobile Robots (AMR) dira eta gidaketa autonomoaren inguruan aipatutako teknikak erabiliz, enpresaren edozein gunetik abiarazi daitezke, edozein unean haien ibilbidea era erraz batean aldatuz. Ezaugarri honek, piezen garraioa prozesu malguak bilakatzen ditu.

AGV-ak industriaren garraio prozesuen automatizaketaren lehengo esteka da, Martinez – Barberá [15] azaltzen duen moduan. Berez, robot sinpleenak dira, inteligentzia murriztua dutelako. Horrek, plataformaren aldakortasunaren kapazitatea eteten du, aldaketa bakoitzeko erabiltzen duten gida magnetikoak mugitu behar direlako. Bestalde, oztopo bat

aurkitzerako orduan mugimendua eteten dute, oztopoak saihesteko behar den inteligentzia ez dutelako [16]. Hala ere, sinpletasunak prozesu egonkorrak ekar dezake. Arazo baten aurrean martxan jartzeko beharrezko diren pausuak, oso arrutan izaten omen dira eta horrek kostuak murriztu egiten ditu. Bati-bat, langileen prestakuntza gutxiago behar dutelako eta konponbideen sinpletasunari ezker, fabrikaren prozesuen gelditze denborak txikiagoak direlako.



Irudia 9 Carretilas mayor saltzen duen AGV-a (Carretilas mayor-etik lortutako irudia)

Irudia 9-an Mercedes barneko AGV baten eredua da, non lurretik agertzen diren lerro beltzak, ibilbidea markatzen duen zintak magnetikoak diren. Naiz eta sistema hauek zientzia aldetik defasatuta egonda, zinta magnetikoarekiko mantendu behar den posizioaren kontrolari buruzko ikerketak lantzea garrantzitsua da. Sistema teknologiko hauek berriztu daitezke, segurtasuna, malgutasuna eta egonkortasuna hobetuz, Tovarnov et al. [17] azaltzen duen moduan.

AMR-ek aldiz, daukaten ekipamendu eta landutako garapenei esker, sistema malguagoak izaten omen dira [18]. Haien nabigazioa elementu fisiko eta finkoei ez dago lotuta, hau da, zinta magnetikorik ez dute behar. Horretarako beste sistema batzuen beharra dago, horien artean nabigazio algoritmoak, mapak sortzeko teknikak eta lokalizazioa lantzen duten inteligentziak aurkitu daitezke. Bonin-Font et al. [19] irudien tratamenduen arloko informazioko bilketa egin zuen. Hala eta be, AMR-ri buruz, beste zenbait atal aipatzen zituen eta beste zenbait argitalpenekin osatu daitezke, adibidez [20].

Lantegiak merkatuaren egoera nahiko nahasita dagoela pentsatzen du. Aukera asko daude halako plataforma mugikorrek erabiltzeko, baina zehaztasun aldetik, informazio gutxi baitago. Horrekin bat, batzuetan komertzialak azaldutako ezaugarriak jasotako produktuan ez dira aurkitzen. Horrek deserosotasuna sortarazten du, bati-bat robot mugikorrek dituen zehaztasun datuak errealitatearekin bat ez datozelako. Pentsa daiteke fabrikatzaile bakoitzak haren inteligentzia garatu duela eta saltzaileek hala ikusarazten dute. Baina informazioaz bilduz, roboten fabrikatzaileak nabigazioa, lokalizazioa, etab. inteligentziak beste enpresa batzuei erosten dizkiote, geroago haien betiko AGV produktu hori txertatzeko eta AMR bat bilakatzeko. Adibidez Kivnon produktuekin gertatzen den mudan. Produktu hori Irudia 10-an aurreikusi daitezke eta tesi honetan erabilitako tresna da.







Irudia 10 Kivnon KV-10 produktua (Kivnon-etik lorturiko irudia)

Enpresa honek Blubotics dituen produktuak erabiltzen ditu, betiko AGV batek AMR batean bilakatzeko. Dena den, Kukak, Asti, ABB, etc. ere haien katalogoan plataforma mugikor adimentsuak badituzte, baina Kivnon-eko jarrera adostu dezakete, Bluebotics bezalako produktuak erabiliz. Irudia 11-an merkatu guztian dauden produktuak ikusten den moduan, momentu honetan Bluebotics bezala hogei enpresa baino gehiago daude.



Irudia 11 Merkatuko produktua (Logistics IQ lorturiko irudia)

Taula 1. Aztertutako AMR-ak

Enpresa	Irudia	Erakarpena	Deskribapena
Kivnon		Industria barneko hardware-a erabili dute	Kivnon burutako AMR, industria barnean dauden elementuekin egin da. Horrek erakargarri egiten omen du, batez be mantentze lanak egiterako orduan. Berez, Blubotics enpresari erosi diote inteligentzia. Bestalde, Siemens PLC baten erabilpena lantegi batzuen integrazioa errazten du.
MIR		Ikuspena erabiltzen duen AMR	Erabiltzen duen teknologiatatik, beste enpresetatik bereizten dira, azken finean irudien tratamenduak egiten dituzte lokalizazio eta nabigaziorako. Eremua argi aldaketak baldin baditu arazoak izaten ditu.
Omron Mobile Robot		Omroneko inteligentzia erabiltzen du	AMR honek, Omron burutakoa da, beraien ekipamendua eta inteligentzia ezarri dute. Horrekin bat, altuerak neurtzeko sensoreak inplementatu dituzte espazio txikiko guneeetan ez sartzeko.
KMR		Mecanum izeneko gurpilek erabiltzen dituzte.	Kuka-k garatutako plataforma mugikorra, non haien robot kolaboratiboa ezarri daitekeen. Jasotako informazioa komertziala izan da, berez frogapenik ez dira egin, horren ondorioz, informazio gehigarririk ez dago.

“Autonomy Solution Provides” agertzen diren enpresen produktuak erabiliz, betidanik egondako AGV-ak AMR-etan bilakatzen dituztela konfirmatu egite du. Baliteke baieztapen honek egia ez izatea, berez AGV/AMR enpresek haren produktuen jatorria nondik datorren ez dutelako esango. Tesi osoan zehar bildutako informazioa Taula 1-an biltzen da.

Enpresa bakoitzak haien baloreak mantenduz produktu ezberdinak kaleratu dituzte. Berez, 2020-garren urterako bertan aurkezten diren AMR-ei Unibersal Robot bat atxiki ahal zaie, Kukakoari izan ezik. Kukako KMR produktuak, Kuka katalogoan dituzten robotak erabiltzen dituztelako. Horrekin bat, Kivnon, MiR eta Omron produktuak funtzionatzen ikusi dira eta Mercedes – Benz dituen baliabideengatik Kivnon-ekoa gehien ikertu dena izan da. Besteak aldiz, erakustaldietan ikusi dira. MiR-eko produktua besteengandik bereizten da, kamerak integratzen dituelako. Baina erakustaldi batean, itzalen posizio aldaketa ondoren galdu egiten zela argi ikusi zen.

Behin plataformen merkatu antolaketa ikusita, AMR daukaten sistema desberdinen informazio bilketa egingo da, non atal-atalez informazio nabarmena bilduko den. Horretarako, Nabigazioa eta Maparen eraikitzea eta lokalizazioa atalak landu dira. Atal bakoitzean gaur egun aurkitu daitekeen teknika garrantzitsuenak aipatuko dira.

### 3 Nabigazioa

---

Bonin-Font et al. [21] azaltzen zuen moduan, zenbait kasuetan, automobil edo robot mugikorrek mapa baten beharra ez daukate, eremu ezezagun batetik abiarazteko. Teknika hauek Online nabigazioa deritzote. Sentsoreen informazioa erabiliz, hasierako puntu batetik bukaerako beste puntu batera mugitu daiteke, inolako istripuak sortu barik. Honek Offline teknikarik ez dagoenik ez du esan nahi. Offline sistemarentzako eremua ezagutzea ezinbestekoa da, beste hitz batzuk erabiliz, mapa baten beharra dago. Offline teknikak path planning bezala ezagutzen dira eta hauek Online teknika baten menpe egon behar da. Azken finean, path planing-ek bide egokiena zein den adierazten du, baina robotak ez du abiarazten. Plataforma mugikorra mugiarazteko, path following algoritmoak behar dira, Mohanan et al. [22] aipatzen duen moduan. Bi teknika hauek bateragarriak dira, robot bat gune ezezagun batean mugiarazteko.

Laburbilduz, AMR batek Online teknika baten beharra dauka. Hortik aurrera, AMR-ren mugimendua gero eta trebeagoa bilakatzeko, Offline eta Online baliabide desberdinak ezartzen dira. Demagun bezero batek, robot mugikor batean oztopoak saihestea ez duela nahi. Kasu horretan, plataforma mugikor horrek, ibilbidearen kalkulua eta ibilbidearen jarraipenaren teknikak soilik erabiliko ditu.

#### 3.1 Online Nabigazioa

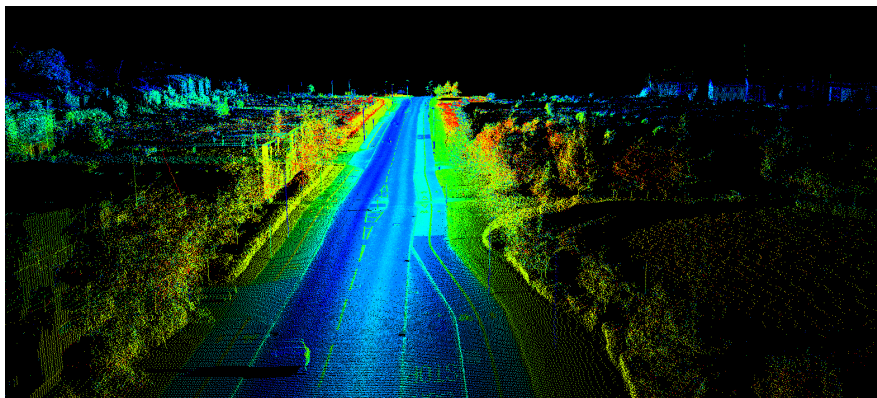
Nabigazio multzo honetan bi azpi multzo bereiz daitezke. Alde batetik, oztopoen identifikapenak eta ibilbidearen desbiderapenak kalkulatzeko dituzten inteligentziak egongo dira. Bestetik, ibilbidearen jarraipena, non AMR-ek offline planifikatuko bideak jarraitzen dituen algoritmoa izango den.

Lehenengo azpimultzoan, Rubio et al. [23] azaldu zuen moduan, nabigazioa azpimota hau abiarazteko sentsoreen erabilpena ezinbestekoa da. Industrietan at dauden robot mugikorrek, sistema hauek erabiltzen dituzte. Adibidez, lurzoruak garbitzen dituzten robotak. Hauek talka bat identifikatzeko orduan haien ibilbidea moldatzen dute, beste gune batera habiaratuz. Oztopoak identifikatzeko orduan, bi metodo erabili daitezke. Lehengo metodoa talka bidezko nabigazioak izango lirateke eta bigarrena talka saihesten duten nabigazioak. Bigarren metodoa honek pertsonen segurtasunerako ezinbestekoak dira.

Talax bidezko multzoan, Wooten et al. [24] sentsore piezoelektrikoak nola erabili behar diren azaltzen du. Talka bakoitzeko robotak ibilbidez aldatu behar du, identifikatutako objektu

hori saihesteko. Baliteke irtengune gabeko gune batera iriztea, beraz talka kopuru konkretu batera irizterakoan, robotak geldiarazi beharko da. Min et al. [25] azaldu zuen moduan, robot kolaboratiboetan motor bakoitzeko indar-pareak aztertzen dira. Parametro hauen arabera, mugapen maila konkretu bat aztertuz, robotak oztopo baten aurrean dagoela jakin dezake. Robot kolaboratiboaren kasuan ibilbideak ez ditu moldatzen, pertsonen segurtasuna mantentzeko geldiarazten da. Indar-pare hauek motorren kontsumoak aztertuz lortzen dira eta horretarako robotaren atzaparretan duen masa jakin behar da, indar pareen mugapena moldatzeko. Parametro hau gaizki ezartzen baldin bada, erreminta objektu bat hartzerako orduan, robotak objektua oztopo bat dela pentsa lezake.

Bigarren multzoan aldiz, ezin dira talkak eragiten. Horrek segurtasuna handitzen du, bati-bat gizakiekin. Horretako metodologia desberdinak planteatzen dira, sentsoreen ezaugarrien arabera. Industria barnean eta AMR-ren kasuetan, LiDAR-ra (Light Detection and Ranging) [26-29] elementu erabilgarrienak dira, azken finean edozein gune automatizatuan, makina eta gizaki lan-eremua bereizteko erabiltzen den tresneria delako. Elementu hauen ezaugarrien artean, espazioaren neurketa bi edo hiru dimentsiokoak izan daitezke. LiDAR-en espazioan zehar dauden elementu bakoitzari neurri bat ematen dio, 2D edo 3D-ko puntuz osatutako mapa bat eraikiz.



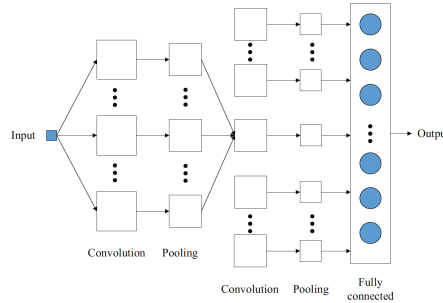
Irudia 12 LiDAR-raren mapa

LiDAR-ak distantziak neurtzen ditu, baina beste tresneria mota batzuk daude, non beste izaera fisiko ezberdinak erabiliz neurketak egin dezaketen. Besteak beste, Ultrasoinuak edo Sonar-rak eta hauek proposamen berdinerako erabili daitezke ere [30,31].

Tesi hau aurkezten den unean, irudien tratamendua punta-puntako teknologia da, baina irudien prozesamendua denbora errealean egiteko, ekipamendu bereziak behar dira. Adibidez, GPU-ak (Graphics Processing Unit)[32-36]. Irudiekin bat datozen teknikak, “*Deep Learning*”, “*Matching Learning*” edota irudien tratamenduak omen dira. Lehengo teknikak momentu honetan lantzen hari den gaia da, sistema hauek gizakiaren burmuinaren funtzionatze modua imitatu egiten dute.

Bengio et al. [37] azaltzen duen moduan, “*Deep Learning*” eta “*Matching Learning*” IA-ren garapenerako ezin bestekoa teknikak dira eta bi hauek ingeniarietza klasikoetatik bereizten dira, lorturiko sistemak irudi batean sortzen diren aldaketen aurrean malguagoak direlako. Hau da, irudien tratamenduak egiterako orduan, normalean hainbat ezinbesteko ezartzen dira, adibidez: argiztapen kontrola, ertzen kalkula, lerroen estimazioak, neurketak, etab. Ingeniarietza klasikoa erabiltzen dituzten teknologiak, beti eremu kontrolatuetan funtzionatzera baldintzatuta daude. Horregatik, edozein auto edo plataforma mugikorraren gidaketa autonomoa planteatzerako orduan, ingeniarietza klasikoak proposatzen dituen teknikak baztertu behar dira. Al gero eta presentzia gehiago lortzen hari du [38], gaur egungo hardwarea halako zientzia erabiltzeko aukera ematen hari delako.

“Deep Learning” barnean, “Convolutional Neural Network” (CNN) sare motak aurkitzen dira, non irudien tratamenduak egiteko aproposenak diren. Izaera ezberdineko sareak daude, baina guztiek proposamen berdina eta antzeko egitura mantentzen dute. Sareak hainbat geruzetan antolatzen dira, geruza bakoitzeko irudiari tratamendu bat aplikatzen zaio. CNN-en geruzen antolaketa Irudia 13 -n ikus daiteke, eta Xin etl al [39] eredu desberdinak azaltzen ditu.



Irudia 13 CNN baten geruzen antolaketa

Aipatu den moduan, sare neuronalaren izaeraren arabera, egiturak eralda daitezke zenbait elementu konstante mantenduz, mota ezberdinetako CNN-ak sortzeko. Entrenamendu eta identifikazio prozesuetan efizientzia handitzeko, sareen egituretan zenbait aldaketa burutzen dira. Canziani et al. [40] CNN motatako sareei frogapen ezberdinak egiten zituen, haien abantailak ikusarazteko. VGG eta Alexnet klasekoak gaizki dimentsionatuta edo “oversize” daudela ondorioztatu zuen. Aldiz, ResNet-18, GoogleNet edota EficienNet oso eraginkorrak direla konturatu zen.

Aplikazioen aldetik, momentu honetan hiru multzo bereiz daitezke. Alde batetik, objektuak identifikatzea, hau da, irudi batean elementu bat baldakoen edo ez adieraztea. Bestetik, irudi batean objektu hori non dagoen bereiztea. Azkenik elementuen sailkapena egitea [41-53]. Berez, aipatutako sareak objektuak identifikatzeko soilik baliagarriak dira eta hauek beste aplikazioetan erabiltzeko, haien geruzak moldatu behar dira. Adibidez, irudiaren elementuaren kokapena emateko, sareak “bounding box” deituriko karratutxoak identifikatu behar ditu eta horretarako sarearen irteeran laukizuzen horren lokalizazioa non dagoen adierazi behar du. Sarearen entrenatzeko prozeduran, karratutxo horien lokalizazioa eman behar zaio, erregio bereziak determinatuz. Geroago sareak objektuak identifikatuko ditu, haien lokalizazioa emanez [54-56]. Irudien segmentazioa egiteko, geruzen eraldaketa egin behar da ere eta teknika ezberdinak aurkezten dira. Adibidez, ResNet-18-ak edo 50-ak DeepLabV3+ bitartez, “encoder-decoder” CNN bat sortarazi daiteke [57-61]. Segmentazioan pixel bakoitzari zein klasetara egokitzen den esan behar zaio eta entrenamenduaren ostean sareak irudi bat ikusterako orduan pixel bakoitzaren klasea itzuliko du.

Behin sentsoreak eta IA ikusita, ibilbidearen jarraipenari buruz informazioa bildu behar da. Teknika hauek sentsoreen informazioa jaso egiten dute eta geroago informazio horiek prozesatu egiten dituzte. Tratamendu honi esker plataforma mugikorrak eremu ezagun edota ezezagunetik mugiarazi daitezke. Teknikak hauek robot mota hauentzako ezinbestekoak dira, bestela lekutik ez liratekeelako mugituko. Aurrekarietan ikusitako moduan, robot mugikor gehienak talka saihesteko sentsoreak erabiltzen dituzte, batez be harrapaketak ekiditeko metodologia bat delako. Istripuak ekiditeen baldin badira, plataformen ziurtagiriak lorrarazteko abantailak ekartzen dituzte.

Path following metodologiak ebazpen matematikoak erabiliz, plataforma mugiaraztea lortzen da. Ebazpen batzuek ekuazio matematiko sinpleak erabiltzen dituzte. Adibidez: batuketak, kenketak eta biderketak erabiltzen dituzte. Corkeek [62] aurkeztutako carrot algorithm-ekin horrela gertatzen da. AMR-ren posizioaren arabera eta helmugaren posizioaren arabera, bien arteko kenketa bat eginez, abiadura eta noranzkoa kalkulaten dira. Beste ebazpen batzuek aldiz, minimizazio funtzioetan oinarritzen dira. Hauen artean Dynamic Window



Approach (DWA) [63-67] aurki daiteke. Laurogeita hamargarren hamarkadan sortutako metodologia da eta gaur egun robot mugikorretan erabiltzen da. Beste zenbait garapen klasikoak aurki daitezke eta Taula 2-a zenbait metodologia laburbildu egin dira.

Taula 2. Online nabigaziozko algoritmoak

Algoritmoak	Erreferentziak	Abantailak	Desabantailak
Dynamic window approach	[63]	Ibilbidea osatzeko robotak mugak kontuan hartzen ditu.	Optimizaziorako pisu finkoak erabiltzen ditu, eta soilik ibilbide bakar bat aztertzen du denbora unitate bakoitzeko
A vector field histogram	[68]	Sentsoreen eta modalizazioaren akatsetatik ziurgabetasunari aurre egitean garrantzi handia ematen dio.	AMR-ren zinematika eta modeloa behar ditu.
TangentBug	[69]	Eremu txikietako informazioa erabiliz, ibilbide txikiak kalkulatzeko ditu	Funtzio Heuristikoa lokalizazio ideala behar du, eta horrek ere nabigazioa moteldu dezake.
PointsBug	[69]	TangentBug baino errendimendu hobea dauka.	Zehaztasun handiko lokalizazioa behar da funtzio trigonometrikoarentzat.
Swarm optimization	[70]	Ibilbidea optimoaren bilaketa partikulen bitartez egiten da.	Emaitzaren optimoa lortzeko etengabeko kalkuluak egin behar dira.
Dynamic movements primitives	[71]	Roboten mugimendu konplexuak mugimendu primitiboetan sinplifikatu egiten dira.	Funtzioaren pisuak etengabe eguneratu behar dira, eta horrek ibilbidearen egonkortasuna zalantzan jartzen du.
Model-based algorithms	[72-75]	Iraganean kalkulatzeko ahaltasuna dauka modelo matematikoei esker.	Funtzionamendua eredu matematiko bati lotuta dago, beraz ebazpena haren zehaztasunari lotuta dago.
Genetic Algorithms	[76]	Ibilbidearen optimoa robotaren parametro zinematikoei eta dinamikoei lotuta dago.	Funtzioak ebazteko kapazitate konputazional altua beharrezkoa da.
Ant colony optimization	[77]	Hainbat agente erabiliz, ibilbidearen kalkuluak optimizatu daiteke	Probabilitatearen distribuzioa exekuzio bakoitzeko alda daiteke, eta moldaketa horiek ez daude prestatuta denbora errealean egiteko, funtzioa konbergitu behar duelako.
Ant algorithm with A* characteristic	[78]	Ant Colony ebazpena baino azkarragoa da eta gainera ibilbidea osoa leunagoa da.	Konbergentzia goizago gera daiteke, limiteen doikuntza ezegokiagatik.
Double Layer ant colony	[79]	Bi exekuzio esker, Ant Colony algoritmoaren prezisioa handitzen da.	Ibilbidearen ebazpena optimizatzeko, beste algoritmo batekin konbinatu behar da.

Taula 2-en ikusitakoaren arabera, “*path following*” egiteko hainbat teknika ezagunak omen dira. Industriako plataformek halako teknikaren bat erabiliko dute. Haien ekipamenduak kaxa beltza bezalakoak direnez, zein metodologia erabiltzen duten ezin da jakin. Dena den eremu ezagun batetik mugiarazteko, “*path planning*” baten beharra dago eta algoritmo hauek zer diren ikusi behar da.

### 3.2 Offline Nabigazioa

Teknika hauek, eremu ezagunetan ibilbidearen ebazteko erabiltzen dira. Berez, AMR - retan gordetako mapei baldintzak ezartzen dizkien teknika da, nondik nora joan behar diren, ze abiadurarekin, etab. zehaztuz. Parametro hauek bi eratara ezar daitezke. Alde batetik, langileak aren eremuaren nortasuna erabiliz, ibilbide bat planteatuz. Bestetik, era automatiko baten bitartez egin daiteke, algoritmoetan balizatuz.

Langileak programaturiko ibilbidea, industria arloan eraginkorra da, lantegiko ibilbideen ebazpena araututa dagoelako. Berez, enpresa batean beste elementu batzuekin

espazioa tartekatu behar du, hau da, plataforma mugikorrek lankideekin, eskorgekin, etab. lan eremu berebean egoten dira. Logistika aldetik lan astuna dela esan daiteke eta algoritmoak oso baldintzatuta egongo lirateke, langileak planteatutako ibilbidea baino optimizatuago bat ebazteko. Era automatiko diren teknikak aldiz, eremu ez arautuetan erabiltzen dira. Hauek zientzia mundurako interesgarriagoak dira, ikerketarako gunea dagoelako bati-bat. “*Path following*” egiteko erabiltzen diren soluzioak, offline barnean ere erabiltzen dira. Adibidez, Castillo et al. [80] algoritmo genetikoaren ebazpena erabili zuen mapatik zehar ibilbidea markatzeko edota Bayindir [81] “*swarm*” inteligentziaren ikerlana egin zuen proposamen berdinerako. Bertan, bai online baita offline teknikarentzako erabili daitekeela ikusten zen. Dena den, algoritmo klasiko hauetatik at beste teknika ezberdinak topa daitezke [82-88].

Tesiaren garapena burutzeko, “*path planning*” atala ez da garatuko. Azken finean, AMR - ren lan eremua azaldutako lehengo kasua delako, non langileak aren esperientziagatik ibilbidea planteatzen duen. Naiz eta offline teknikak industrian ez burutu, mapak sortu eta robotak lokalizatu behar dira, horretarako laugarren atalean, zein algoritmo erabiltzen diren planteatuko dira.

## 4 Maparen eraikitzea eta lokalizazioa

---

Mapa eraikitzeke Simultaneous Localization And Mapping (SLAM) teknika erabiltzen da. Bi etapa ezberdinetik pasa behar da. Lehengo etapak datu bilketarako erabiltzen da. Datu horien artean, oztopoen posizio absolutuak biltzen dira. Bigarren etapan aldiz, SLAM-ak datu bilketa horrekin mapa osatu eta AMR lokalizatu egiten du. Mapa aldiune bakoitzeko berriztuko da, momentu bakoitzari egokitzeko. Ebazpen honek konputazionalik oso kalkulu astunak izaten dira eta momentu honetan soilik mapak eraikitzeke erabiltzen dira.

Aipatu den moduan, merkatuko AMR batek enpresan sartu bezain laster mapa eraikitzea behartzen du. Soilik nabigazioa eta lokalizazioa sistemak erabiliz AMR abiarazten da. Enpresaren lan eremua asko eraldatzen bada, mapa berri bat osatu beharko da, lokalizazioaren algoritmoak gordeta duen maparekin plataforma mugikorraren posizioa ezin duelako determinatu. Datuen bilketa egiteko, online nabigazioak erabiltzen dituzten sentsoreak erabiltzen dira, sentsoreen sistemen bikoizpena saihesten delako. Azaldu den moduan, industria barnean eta kasu gehienetan LiDAR-ra eta gurpilen mugimendua erabiliko dira. SLAM-ari buruz hitz egiterako orduan, beti lokalizazio absolutuan oinarrituta dagoela pentsa daiteke, hau da, jatorria puntu finko batean jarriz plataforma mugikorra puntu horretatik non dagoen adieraztea datza. Baina lokalizazio erlatiboa ere aurki daiteke. Horretarako bi aldiuneko informazioaz baliatuz, robot mugikorraren posizioa edo mugimendua estimatzeko aukera ematen du [89].

Lokalizazio absolutuaren barnean, bi ezagun diren ebazpen mota aurki daitezke. Lehena Particle Filter (PF) [90] da eta bigarrena Extended Kalman Filter (EKF) [91] da. PF izaera mantenduz, teknika eta garapen desberdinak planteatu dira. Edozein sentsore motarako prestatuta daude. Hau da, algoritmo honek bai ikuspenera erabiliz, baita LiDAR-aren informazioa erabiliz ebazpen ezberdinak ikusi dira. Gainera, teknika ezberdinen konbinaketak eginez, beste zenbait ikerkuntza ikus daitezke. Laburbilduz, algoritmo honi zenbait hobekuntza planteatu zaizkio, haren zehaztasuna hobetzeko [35,89,92-99].

Hala ere, teknika guztiek izaera berbera daukate. Mapa osoan zehar banandutako partikulei esker, elementu baten posizioa determinatu daiteke. Horretarako haien lokalizazioaren arabera, plataforma mugikorraren lokalizazio estimatu bat planteatu daiteke, probabilitatezko funtzio bat erabiliz. SLAM-arekin amaitzeko, mapari berrizteko funtzio bat aplikatzen zaio, non plataforma mugikorrek dituen sentsoreak baliatuz eta partikulek estimatutako posizioaren arabera, maparen egoera konparatzen da. Mapa honek AMR-ren disko gogorrean gordeta dago eta bertan oztopo guztien informazioa biltzen da. Konparaketaren bitartez oztoporen bat posizioa aldatu baldin bada, haren posizioa berriztuko da, mapa beti eguneratuta izateko.

EKF teknikak SLAM prozedura egiteko lehengo algoritmoa izan zen eta edozein teknika bezala, eraldaketak jasan ditu, sistema ez linealekin erabiltzeko [72,95,100-107]. Teknika honek, sistema linealekin funtzionatzeko prestatuta zegoen, beraz robot mugikorren inguruan dauden prototipo edo produktu guztiak filtro honen erabilpenetik at uzten baititu. Taylorren serieak erabiliz, sistema ez lineal batek sistema lineal batean bihurtu daiteke, lan baldintza batzuk ezarri. Hurbilketa bat egiten denez, lokalizazioan zehaztasuna galdu daiteke.

Bi ohiko teknika hauez gain, zenbait SLAM saihebidetarik ezberdinak sortu omen dira. Batzuk PF oinarria erabiltzen dituzte, adibidez FastSLAM [95,96,98,108-114]. Ebazpen honek eremu ezezagun batean hain bat marka edo "Landmark"-en posizio ezagunetan ezarri, robotaren lokalizazioa estimatu daiteke. Rao – Blackwellized, erabili daitekeen beste prozedura oso ezaguna da eta zenbait kasuetan PF algoritmoaren hobekuntzarako erabili da [93,96,97,110,115-118]. Hauekin bat GraphSLAM [119-124] eta RGB-D SLAM sistemak aipagarriak dira. GraphSLAM-ek sakabanatutako matrizeak (tamaina handiko matrize bat da eta haren datu gehienak nuluak dira) eraikitzen ditu, interdependentzien ikuskatzaile baten faktoreen grafiko bat sortuz. Hau da, bi ikuskatzaile marka edo "Landmark" berbera identifikatzean, interdependentzia bat sortarazten da. Aldiz RGB-D SLAM[31,108,109,125-131], lokalizazioa burutzeko RGB irudietan eta sakonerako sentsore batez 3D eremuak eraikitzeko erabiltzen da. Horregatik, baieztatu daiteke SLAM-ak irudietatik datorren informazioa erabili dezakeela. Behin AMR-ak nola mapa eraikitzen eta lokalizatzen duten ikusita, robot kolaboratiboen inguruan dauden erakarpenak aipatuko dira. Azken finean, planteatu nahi den sistemak bi elementu beharra behartzen du eta bien inguruan dagoen informazioa tratatu behar da.

## 5 Robot Kolaboratiboen egoera





---

Robot kolaboratiboak merkatuan etorri berriak dira, hauek gizakien lan eremuan bestelako segurtasun gabe lan egin dezakete. Ezaugarri honi esker, beste robot arruntetatik bereiz daitezke. Robot klasikoek zeluletan gizakiak eta robotak lan eremua bereizita dago, segurtasun hesiak erabiliz. Hesi hauetan langile bat sartzerako orduan robotaren jarrera eraldatu egiten da, segurtasuna handituz robotak geldiarazten delako.

Momentu honetan teknologiak ematen dituen aukerengatik, lan eremua tartekatzea da. Horretarako, robotak bestelako segurtasun ekipamendua integratuta izan behar ditu. Sistema normalenak talka bidezkoak izaten omen dira, non erresistentzia baten aurrean mugimendua eteten duten. Inpaktu honen eragina ahalik eta txikiena izan behar da. Inpaktu honek legedi batean jasotzen da eta gorputzaren gunearen arabera limite ezberdinak daude. ISO / TS 15066 izenarekin ezagutzen da eta edozein robot kolaboratiboren zelulen ziurtagiriak lortzeko talka frogapenak gaitu behar ditu.

Irudia 11-n robot kolaboratibo inguruan merkatuaren egoera argi eta garbi ikusten da. Komertzialak emandako informazioa Taula 3-n irudikatzen da. Kasu batzuetan, informazioa hobetzeko zuzeneko bilerak antolatu dira, haien esperientzian oinarrituriko adibideak ikusteko.

Taula 3. Robot Kolaboratiboen merkatu egoera

Enpresa	Irudia	Erakarpena	Deskribapena
Universal Robot		Lau robot kolaboratibo	Berez merkatuaren produktu merkeena da. Horrekin bat, legedia nahiko sakondu dute, betebeharrezko diren arauak betetzeko. Horrek ez du esan nahi, robot txarra denik, baizik eta merkea izateko segurtasun barean dauden multzoekin jolastu dutela.
ABB Yumi		Bi besotako robota	Pieza txikiak muntatzeko erabiltzen den robota, berez haren besoak oso luzeak ez direlako, baita lan zama txikiak jasaten dituelako.
Fanuc		Lan zama handiko robotak	Fanuc robot arrunteei funtzionamendu abiadurak murriztuz eta motorren kontsumoen analisia eginez, gainezko esfortzuak edo talkak identifikatzeko; balio izan zaie robot kolaboratiboak sortzeko
liwa		Robot sentzitiboa	Kuka haren robotari kolaboratiboa deitu beharrea sentzitibo deritzote. Hareatarako, talkak identifikatzeko sentsoak motorren kontsumoetan oinarritu beharrea, beste sentso mota erabili dituzte, eta enpresak ez ditu adierazi.

Berez tesi honetan, robotikarekin zuzenean zer ikusia duten ikerketak ez dira egon. Aldiz, robotarentzako erreferentziak nola lor daitezkeen informazioa bilatu egin da eta aukera guztietatik irudien tratamenduan oinarrituriko ikerketak jarraitu dira. Kontuan izanda, Mercedes – Benz barnean zenbait egoera ezberdin planteatu izan direla. Hauen artean: erreferentzia kolaboratiboak, mekanikoak eta ikusmen oinarritutakoak aurkitu daitezke. Baina prototipo berri bat denez, azkenekoa planteamendua aukeratu da, batez be zientzia aldetik erakarpen gehiago daudelako.

Erreferentziatzko egiteko, nabigazio online moduan, ikusmena bi multzotan bereiz daiteke. Ingeniaritza beharrezko duten tratamenduak eta AI teknikan. Azkeneko multzo honetan ezagun diren “*Pattern Recognition*” edo “*Deep Learning*” aurkituz. Ingeniaritza klasikoak ematen dituen balia bideak baztertu dira, burutu nahi den aplikazioa gun e aldakorra delako eta edozein prozesuetarako malgua izan behar delako. Azken finean elementuen identifikapenerako erabiltzen diren teknikak, bai plataforma mugikorrean, baita robotean topa daitezke. Hala eta be, badaude robotekin erlazionatuta dauden ikerketak [132-135].

# III.

## Atalburua

Pertsona bat ulertu nahi baduzu, haren hitzak ez entzun, begiratu haren jarrera  
-Albert Einstein-

### Teknologia frogatzen

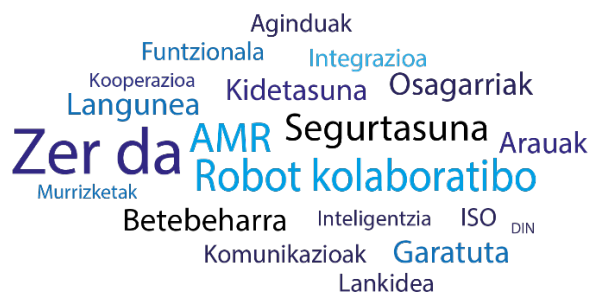
---

---

Hirugarren atalburu honetan, proiektuaren garapen filosofiaren bigarren atalari dagokio. Bertan bai Unibertsitatean, baita lantegi barneko laborategian eskura dauden industria 4.0 produktuak frogatu dira, haien mugapenak jakiteko. Merkatuaren egoera aztertzerako orduan, zenbait atal ulertu gabe gelditu ziren eta frogapen hauekin sorturiko zalantzak argitu nahi dira. Komertzialekin zalantza tekniko konkrituak argitzeko denbora galera izaten omen dira, produktua saltzen duen arduraduna, ingeniari taldeak garatutako algoritmoei buruz informazio gutxi dutelako. Normalean, tesiak behar zuen informazio konfidentziala izaten delako bezeroentzako. Batzuetan saldutako teknologiatik espero diren helburuak ez betetzea gerta lekiok, batez be AMR -ak elementu berriak direlako eta enpresek egin dituzten azterketak eremu oso kontrolatuetan izan omen direlako, produktu hauetan itxaropena galduz. Hala ere, adimen pixka batekin eta asmakizunetik baliatuz, teknologiak aurrera eraman daitezke eta frogapenetatik lortzen diren ondorioak, aurrerapenak dakarte prototipoentzako.

Merkatuan gelditu beharrean, zientziaren munduan aurkezten dituzten baliabideak ere aztertuko dira. Azkeneko urteetan, edozein ikerketa proiektuan simulagailuek ezin bestekoak dira. Beraiekin ideiak era erraz batean irudikatzen dira eta edozein elementu fabrikatu aurretik, haren jarrera, forma, etab. planteatu daiteke. Bestalde, Erreminta birtual hauek aurkezpenetarako aproposak izaten dira. Zenbait kasutan, zientziak garatzen dituen inteligentziak era bisual batean aurkezteko modu bakarra izaten da eta hori edozein pertsonari azalpenak emateko baliagarriak da. Adibidez, garapenetik at dagoen pertsonentzako simulatutako auto baten mugimendua erakargarriagoa izango da, naiz eta mugimendu hori adierazpen matematikoen bitartez egonda.

Edozein ikerketa proiektu bezala, ezezagun den mundu batean sartzerako orduan burura galdera asko etortzen dira. Irudia 14-an era bisual batean izandako galderak ezarri dira, non atal honetan galdera hauei erantzuna bilatuko den.



Irudia 14 Buruan izandako galderak

Frogapen guztiak era egituratu batean azaltzeko, lau multzotan bereizi egin dira. Lehendabizi “*autonomous mobile robot*”-eko ikerketak aurkeztuko dira. Geroago, robot kolaboratiboetatik ikasitako informazioa bilduko da. Behin robota aztertuta, honi ezarritako erremintaren ezaugarriak aipatuko dira, konkretuki bihurkarien informazioa jasoko da. Azkenik erabilitako simulagailuen ereduak aurkeztuko dira. Saillkapen hau biltzen duen informazioa, Mercedes – Benz eskaintako baliabidetik lortu da gehien bat.

## 1 Autonomous Mobile Robot

---

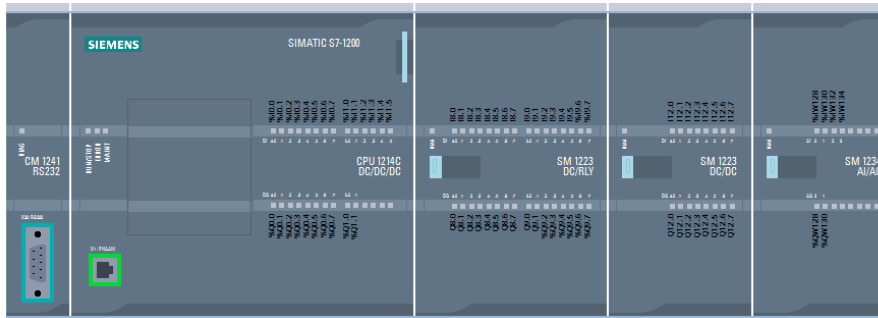
Hain bat gunetan aipatu den moduan, tesi honen garapenerako bi ezinbesteko elementu komertzialak erabiliko ditu. Atal honetan AMR-ri buruzko informazioa bilduko da eta Irudia 10-an aurkezten den modeloa erabili da. Produktu honek industriadako abantaila asko ekarri ditu. Hasiera batean, zientzia arlotik erakarpen asko zituela pentsatu zen, baina denbora pasa hala baieztapen hau desagertu zen. Hala ta be, pausuz - pausu 4.0 tresneria honek enpresa aldetik dituen puntu interesgarriak azalduko dira.

Edozein lantegirentzat AMR-ak duen elementu interesgarriena “*Programmable Logic Control*” (PLC) da. PLC-ak automatismoen munduan ezaguna den tresneria da, non edozein makina industrial batek halako ekipamendua erabiltzen duen. Kasu honetan, AMR - ak erabiltzen duen PLC marka, enpresa osoan zehar erabiltzen da. Horrek, langileentzako elementuen komunikazioak, programaketak, etab. egiterako orduan sinpletasuna eta erosotasuna ekar dezake. Fabrikatzaileen arabera ekipamendua eralda daiteke, adibidez zenbait AGV-etan fabrikatzaileak haien ekipamenduak diseinatzen dituzte. Kontuan hartu behar da, AGV - ak AMR-rekin inteligentzia aldetik konparatuta, AGV-ak sinpleagoak dira.

Kivnon enpresak saltzen dituzten plataforma mugikor sinpleetan PLC-ak erabiltzen dituzte, haien bezeroak horrelako ekipamenduak eskatzen dituztelako. Horregatik fabrikatzaile honetan, AVG-ren eta AMR-ren diferentzia aipagarriena, BlueBotics ekipamendua da. BlueBotics beste fabrikatzaile bat da. Bere produktua ordenagailu industrial bat da eta edozein plataforma mugikorrari nabigazio autonomo kapazitatea ematen dion hardwarea da. Kivnon nabigazio autonomoaren zientzia garatu beharrean, haien plataformetan produktu teknologiko bat txertatu du. Azken finean, Kivnon, beste edonor bezala, merkatuan dauden dispositibo industrialak erabiliz, haien AGV-a AMR-n bilakatu dute. Osagai industrialen muntaketan zeozer falta baldin bada eta merkatuan ezin bada topatu, beraiek integratuko lukete.

Mercedes – Benz barnean erabiltzen duten fabrikatzailearen arabera, izaera berdineko bi tresneria erosi daiteke, baina haien mugimendua burutzeko sistema ezberdinak erabiltzen dituzte. Kasu batzuetan AMR-ak hobeagoak direla pentsa daiteke, batez be nabigazio autonomoa erabiltzen dutelako eta horrek enpresa barnean malgutasuna dakarrelako. Ikasitakoaren arabera, AMR-a edo AGV-a hautatzeko, lan egoera baldintzatuko dute. Dena den, tresneri malgua eta aldakorra izatea, interesa suspertzen du. Batez be, gaur egungo fabrikazio balditzek, prozesuak etengabe aldatzera behartzen duelako produktuaren fabrikazio kostua murrizteko.

Enpresak edozein produktua erosterako orduan, haren enpresa barnean adostutako hardwarea izatea gustatzen saio. Edozein aldaketa egiterako orduan, langileak eraginkorragoak izango direlako. Produktu bat ezagutzerako orduan esperientzian oinarritutako ikasketa lortzen da, arazoak azkarrago konponduz. Kivnon-en kasuan Siemens-eko PLC-ak erabiltzen ditu, konkretuki S7-1200. PLC horri zenbait portu atxiki zaizkio, AMR-ak dituen osagai artean informazioa tartekatzeke (Ikusi Irudia 15).



Irudia 15 AMR-ren PLC-a

Hurrengo lerroetan AMR-ren osagai guztiak azalduko dira. Horrela unibertsitateak planteatutako prototipoaren muntaketarako, behar duen informazioa bilduko da. Ezkerretik hasita, lehengo modulua serie portu (CM1241 RS232) bat da. Portu honekin BlueBotics-aren nabigazio sistemarekin eta Bluetooth modularekin komunikatzeko erabiltzen da. Hurrengo elementua CPU-a da, non informazio guztia prozesatu egiten den. SM1223 DC/RL txartela sarrera eta irteeran digitalak dira, oso baliagarriak periferian dauden elementu elektroniko eta elektriko guztiak kontrolatzeko. Adibidez, segurtasun interruptoreen seinaleak konektatzeko eta potentzia altuko sarrerak eta irteerak sortarazteko. Horretarako ekipamendu honek errele baten bitartez, konektaturiko elementu guztiak elikatze iturritik energia hartzen dute. Laugarren tresneria, SM 1223 DC/DC, potentzia txikiko sarrerak eta irteerak izateko erabiltzen da, hauen artean botoien seinaleen erregistroa egiteko. Azkenekoa SM 1234 AI/AQ, sarrerak eta irteerak analogikoak dira, berez motorrak kontrolatzeko erabiltzen dira.

Esan den moduan PLC-az aparten Bluebotics-eko ekipamendu bat erabiltzen da, nabigazioa kontrolatzeko dispositibo dena. Merkatura esekitzen diren aukera guztietatik Ant Localization Plus tresneria erabiltzen da. Berez hitz hauek ez daukate informazio nahikorik eta horretarako ezaugarrietara joan behar da. Ekipamendu honek industriarako prestatuta dagoen ordenagailu bat da, normalean IPC deritzona. Etxean aurkitu daitekeen edozein ordenagailu bezalako izan daiteke, baina hauek industria barnerako lan baldintzetarako prestatuta dauden ekipoak dira. Industria barnean gainditu behar dituzten baldintzak honako hauek dira:

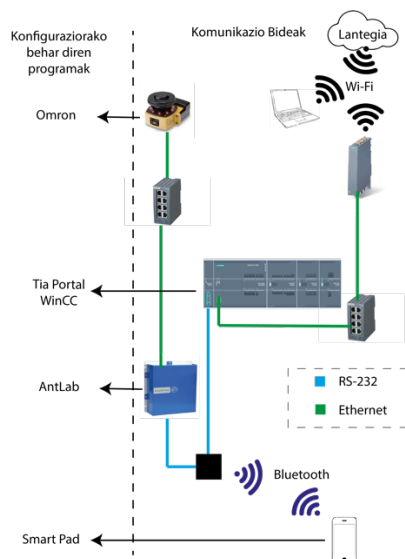
- Aurkako lan baldintzak: Bibrazioak, hautsak, hezetasuna, eremu elektromagnetikoak, tenperatura altuak, eta 24/7 lan orduak.
- Komunikaziorako prest: Integratutako sarrerak eta irteerak egokituta daude lantegietako komunikazioetarako.
- Urte askoko piezen horniketeta. Ordenagailu batek bat edota bi urte igaro hala osagai berriak dituzte, IPC-etan elementu elektriko eta elektroniko hauek merkatuan denbora gehiago mantentzen dira, gutxienez 10 urte.

Bluebotics-eko dispositiboak maparen eraikitzea, lokalizazioa, path planning-a eta path following-a egiteko aukera ematen ditu. Dena den, path planning-a langilearen menpe dago, ekarpenetan azaldu den moduan haren esperientzian oinarritutako ibilbideak egiteko. Bluebotics erabiltzen dituen inteligentziak ezezagunak dira, baina fabrikatzaileak  $\pm 1$  cm-ko eta  $\pm 1^\circ$ -ko erroreak izan dezakeela kasu hoberenean aipatzeko du. Baita 20 Hz-etan funtzionatzen duela konkretatzen du ere.

Nabigazio fabrikatzaile honek aurkezten duen nabigazioa funtzioak, lan eremuan zehar erreflektore batzuk behar ditu. Hauek espazioan zehar puntu finko edo "landmark" bezala funtzionatzen dute. Horretaz gain, erreflektoreak ezin dira edonola ezarri, AMR-ak gutxienez bi-hiru erreferentzia ikusi behar ditu, adierazitako zehaztasuna lortzeko. Aurrekarietan ikus den moduan, erreferentzia hauek lokalizaziorako erabiltzen dira eta AMR-ak puntu hauek etengabe ikusi behar dira.

Erreflektore hauek ikusteko, erabilitako AMR modeloak, bi Omron-eko segurtasun LiDAR erabiltzen ditu, bata aurrean jarrita eta bestean atzean. Konfigurazio mota honek, bi noranzkotan abiarazteko erabiltzen da. Sentsore industrial hauek 270º ikusteko ahalmena daukate, baina AMR-an 200º-tara mugatuta daude. Bi noratzetan abiarazi nahi baldin bada, plataformak puntu itsuak izatea onar gabea delako.

Ikusitakoaren arabera, gutxienez hiru sistema komunikatu behar dira. LiDAR-rak Ethernet bidez IPC-arekin informazioa tartekatzen dute eta IPC-ak PLC-arekin RS-232 portua erabiliz. Baina badaude beste periferiako elementu batzuk, funtzionamendurako ezinbestekoak diren (Begiratu Irudia 16).



Irudia 16 AMR-ko komunikazioak

Irudia 16-aren ezkerreko aldean, konfigurazioak egiteko beharrezkoak diren programak azaltzen dira eta eskuinaldean AMR-ren eta enpresak dituen barne komunikazioak. Alde batetik Ethernet komunikabidea dago, non switch-ak komunikazioen kontrola egiten duen. Elementu honek, LiDAR-ren informazioa, S7-1200-ren komunikazioa, AntLab edo BlueBotics eskaintzen duen baliabideen eta Jual –aren kontrola egiten du. Jual-ak lantegiak dituen beste osagai informatikoekin informazioa tartekatzeko erabiltzen da, baina komunikazio honek Wi-Fi -ren bitartez egiten da.

BlueBotics eskaintzen duen produktuaren komunikazioa ezaugarri bitxia da. Irudia 16 -aren eskema ikusten baldin bada, naiz eta Ethernet konexioa izanda soilik sarrera moduan erabili daiteke. AntLab-eko irteera bakarra serie bitartez egiten da. Horregatik, AntLab kutxatik lerro urdin bat ateratzen da. Lerro urdin horrek zuzenean kutxa beltz batera joaten da. Bertan, Kivnon-eko programazioa bat gordeta dago eta MainBoard deritzote. Bertatik 2 komunikabide ateratzen dira, bata Bluetooth-a eta bestea RS-232-a Siemens-ekin AntLab eta MainBoard komunikatzeko. Bluetooth sistemak AMR-ak mugikorretik mugiarazteko erabiltzen da. Horregatik MainBoard BlueBotics-eko dispositiboarekin konektatua egon behar da, mugikorrek mugitzen den bitartean, aginduak jasotzeko eta lokalizatzen joateko. Bestetik mugikorrek motorren kontsignak Siemens-era bidali behar dira, bertan motorrak abiarazten direlako. Ikusten denez, komunikazioak nahiko nahasita daude, batez ere RS-232-eko komunikabidean. Bai Mercedes – Benz, baita unibertsitatean, elementu hau Ethernet bitartez egin beharko lirakeela deritzote, komunikazio BUS-a azkorragoa delako.



## 1.1 Frogapenak eta Ondorioztapenak

Behin AMR-ren osagaien antolaketa ikusita, Kivnon-eko plataforma mugikorrei egindako frogapenak aztertuko dira. Mercedes – Benz eskainitako laborategiari esker, teknologia honen funtzionamendua aurreikusteko aukera egon da, Kivnon-ek eta Bluebotics-ek esandakoa betetzen den frogatuz. Horretarako, lantegi barnean dauden beste proiektu batzuk jarraitu dira.

Kivnon-eko produktua, elementu ia industrialak kontsideratu daitekeela esan daiteke. Hasiera batean sistema industrialak bezala saldu zuten, baina barnean zenbait elementu elektroniko eta elektriko horrelako izaera ez daukate eta frogapen ondorioz makinaren jarrerari ikusten da. Aurkitutako lehen arazoa, Siemens eta AntLab bitartean dagoen kutxatxo beltzetik dator. Zenbait kasutan mugimendu arraroak eragiten zituzten, batez ere abiadura hartzerako orduan eta bibrazio bat sortzerakoan. Kasu honetan bat-batean AMR-ak geldiarazten zen, motorren gelditze sistemak sartuz. Beste zenbait kasutan AMR-aren lokalizazioan eraginak sortzen zituen. Horregatik komunikazio galera bat zegoela susmatu zen. Behin kable guztiak aztertuz, RS-232 kableak kutxatxo beltzean konektore bizkorak erabiltzen dituzte (Beguiratu Irudia 17). Berez konektore hauek prototipoak egiteko aproposak dira, baina ez makina industrial baterako.



Irudia 17 Konektore bizkorak

Halako muntaketa ikusita, Kivnon-ei azalpenak eskatu zitzaizkien, zergatik erabiltzen zituzten halako konexio ez profesionalak ulertarazteko. Beraien argumentua MainBoard (Kutxa beltza) halako konexioak behar zituzela izan zen. Bilera amaitu ostean, halako batean Arduino motatako elementu bat zegoela susmatu zen. Gainera, horrek azalduko luke zergatik Bluetooth konexioa MainBoard barnean zegoen eta ez Siemens modulu batekin.

LiDAR-arekin beste arazo batzuk topatu dira. Arazo hauek izaera fisikoagatik datoz, azken finean argi izpi bat elementu batean islatzerako orduan, sentsoreak objektua identifikatzen dute. Islapen horren kalitatea neurtuz, LiDAR-aren softwarea erreflektore bat den determinatu dezake. Hau da, edozein objektu argi izpi horren islapenaren kalitatearen muga gainditzen baldin badu, erreflektore bezala klasifikatuko da. Adibidez Kivnon-eko zuzentzaileak kotxeen hagunekin gerta lekiela abisatu zuen, baina laborategian segurtasun jantziarekin, edota segurtasun botekin gerta daiteke frogatu zen.

Arropen erreflektoreen arazoa AMR-an mapa baten eraikitzerako orduan topatu zen, bi pertsona AMR-ren inguruan lan egiterako orduan. Une horretan, AntLab-eko softwarean bide erdian bi erreferentziako puntuak zeudela ikusi zen eta puntu horiek laborategian ezarritako erreflektoreak ez zirela argi zegoen. Bi oztopo handi zirela pentsa daiteke, baina programak oztopoak eta Landmark-ak ondo bereizten eta irudikatzen ditu. Horrek, AMR-ak egongo den eremuaren kontrola mantendu behar dela ondorioztatu daiteke. Bestela, haren lokalizazioa eta zehaztasuna eralda daitezke.

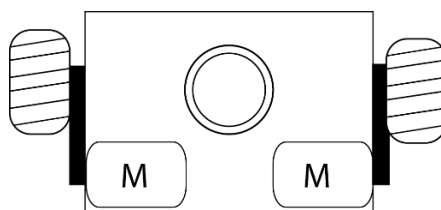
Azkeneko arazo aipagarriena lokalizazioaren funtzionamendua da. Berez, hiru urteetan plataforma mugikor honi aldaketa eta moldaketa asko egin zaizkio, tesian garatu nahi den robotak lortzeko eta plataformak dituen beste galera batzuk zuzentzeko. Gertakizun guztietatik, lokalizazioa burutzeko gorpilaren abiaduraren neurketaren eragina aipagarriena da. Berez, nabigazio erlatiboaren egiteko abiadurak neurtzen dituzten sentsoreak erabiltzen dira, baina kasu

honetan lokalizazio absoluturako erabiltzen dira. Zientziak erakutsi duen moduan, halako lokalizazio globala lortzeko, hainbat sentsoreen neurketak nahastu behar dira, ahalik eta posizionamendu zehatzagoa lortzeko. Teorian, gurpilen abiaduraren analisia egitea sentsuzko da, baina errealitatean ez. Adibidez, industria eremuetan gurpilak irristatu egin daitezke, desplazamendu irakurketa txarra eginez. Horregatik AMR gehienak LiDAR-ren informazioetan baliatuz islatzaileak topatzen ditu eta lokalizazioa puntu finko horiei esker lantzen da. Laborategian dagoen Kivnon-a gurpilen desplazamenduan soilik oinarritzen dela ondorioztatu zen, nahi gabeko frogapen batean. Berez, lantegiko lurzorua laua ez denez, plataforma mugikorra tontor batean harrapatuta gelditu zen, gurpilak mugimenduarekin jarraituz. Horrek sentsoreak desplazamendu ez egokiak neurtzen jarraitu zituen, aldagai batean metroak metatuz. Behin AMR-ak egoera horretatik irteten lagunduz, hasierako posizioa bueltatzea agindua eman zitzaion, baina hasierako posizioa itzuli beharrean, aldagaian gordetako metro kopurua kentzen hasi zen, zerora irizteko asmoz. Azkenean hasierako posiziotik lau metrotara desbideratuta gelditu zen. Bertatik beste ondorioztapen bat lortu zen, LiDAR-ak lokalizaziorako ez dira erabiltzen eta halako egoerak motorren mugimendua eten ezin dutela.

## 1.2 Modelo Matematikoa

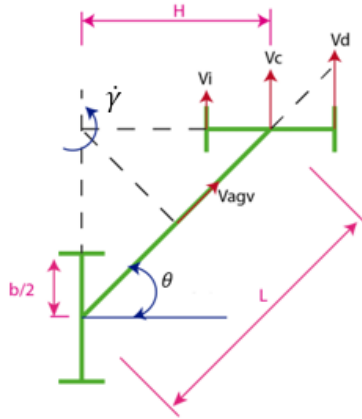
AMR-a simulatzeko intentzioarekin, AMR-aren ekuazio dinamikoak definitu behar direla ondorioztatu zen. Horregatik, atal honetan Kivnon-eko modelo matematikoa aurkeztuko da. Plataforma honek, biraketa erradioa murrizteko mekanismo berezi bat du.. Berez, Kivnon-a 1800 mm luzerako eta 400 mm zabalerako plataforma mugikorra da. Plataforma hauek sagu bezala ezagutzen dira eta masa altuko piezen garraioa egiteko aproposak dira. Biraketetan trebetasuna handitzeko, Kivnon-eko diseinatzaileak gurpilen bi ardatzak poleen bitartez lotu zituen, egitura berezi bat osatuz.

Motorren kasuan, kokapen berezi bat daukate. Bi motorrak aurreko ardatzean jarrita daude, tanke baten erara. Motor bakoitzaren abiadura aldaketarekin, biraketa ezberdinak sortarazten dira. Biraketa hauek, poleen bidez atzeko ardatzari biraketa mugimendua transferitzen zaio. Atzeko gurpilak jarraitzaileak izango dira, eragile elektrikorik ez daukatelako.



Irudia 18 Motorren kokapena

Erakarpenetan egindako ikerketen ondorioz, modelo matematikoaren sarrerak abiadura ( $V_c$ ) eta noranzkoa ( $\gamma$ ) izango dira eta irteerak posizio baldintzatzen dituzten abiadurak, hau da,  $V_x, V_y, \dot{\theta}$ . Plataforman zehar abiaduraren eragina ikusteko, hauen deskonposaketa aztertuko da (Ikusi Irudia 19).



Irudia 19 Kivnon plataformaren abiaduraren deskonposaketa

L eta b plataformaren egituraren arabera finkatzen dira, non L bi gurpil ardatzen arteko distantzia ezartzen duen parametroa da eta b gurpil ardatzaren arteko zabalera adosten duena.  $V_{agv}$  AMR-ak eramango duen abiadura da eta  $\dot{\gamma}$  eramango duen norabidea. Orduan eskuineko ( $V_d$ ) eta ezkerreko ( $V_i$ ) motorearen abiadurak kalkulatzeko, ( 1 ) ( 2 ) ekuaziotan oinarritu beharra dago.

$$V_i = \dot{\gamma} \cdot (H - b/2) \quad (1)$$

$$V_D = \dot{\gamma} \cdot (H + b/2) \quad (2)$$

H errotazio puntutik  $V_c$  abiadurara dagoen distantzia da eta  $\dot{\gamma}$  norabidearen abiadura angeluarra da. Motorrek daukaten abiadura erabiliz eta b parametroari esker lortu daiteke  $\dot{\gamma}$  parametroa ( begiratu ( 3 )-garren ekuazioa).

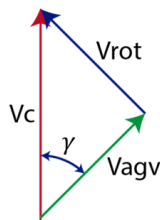
$$\dot{\gamma} = \frac{V_D - V_I}{b} \quad (3)$$

$V_{agv}$  abiadura  $\dot{x}$ -n eta  $\dot{y}$ -n bereiz daiteke. Bereizketa horrekin koordinatu polarretan eta era absolutuan X eta Y posizioak lortzeko baliagarriak dira.

$$V_x = \|\vec{V}_{agv}\| \cdot \cos \theta \quad (4)$$

$$V_y = \|\vec{V}_{agv}\| \cdot \sin \theta \quad (5)$$

$\theta$  mapa absolutuan daukan errotazioa adierazten du eta denborarekiko eraldatuko da.  $V_{agv}$ -ren balioa ( 6 )-garren ekuazioarekin lortzen da, eta horretako  $V_c$ -ren abiaduraren deskonposaketa nola planteatu den jakin behar da. (Ikusi Irudia 20).



Irudia 20 Vc abiaduraren deskonposaketa

$$\|\vec{V}_{agv}\| = V_c \cdot \cos \gamma \quad (6)$$

Horrekin bat,  $V_{rot}$ -aren kalkulua planteatu daiteke.

$$V_{rot} = V_c \cdot \sin \gamma \quad (7)$$

$$\frac{\dot{\theta}L}{2} = V_{rot} \quad (8)$$

Beraz, (7) eta (8) ekuazioak konbinatuz, (9)-arren ekuazioa lor daiteke

$$\theta = \frac{2}{L} \cdot V_c \cdot \sin \gamma \quad (9)$$

Orduan, azaldutako ekuazioen konbinaketa eginez modelo matematikoa lortuko da (ikus (10)-garren ekuazioa).

$$\begin{bmatrix} V_x \\ V_y \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \gamma \cdot \cos \theta \\ \cos \gamma \cdot \sin \theta \\ \frac{2}{L} \cdot \sin \gamma \end{bmatrix} * V_c \quad (10)$$

Laburbilduz, Kivnon-eko modelo matematikoa (10)-arren ekuazioarekin definitzen da eta edozein posizio lortzeko erantzunen integrazioa egin beharko lirateke. L eta b parametroak AMR-etik lortu behar diren datu bakarrak dira. Edozein nabigazio algoritmo garatzerako orduan, honek AMR-ri  $V_c$  eta  $\gamma$  etengabe emango dizkio, plataforma mugikorraren posizioa jakiteko. Modelo hau hobetzeko, motorren modeloaren denbora erantzuna txertatu beharko lirateke.

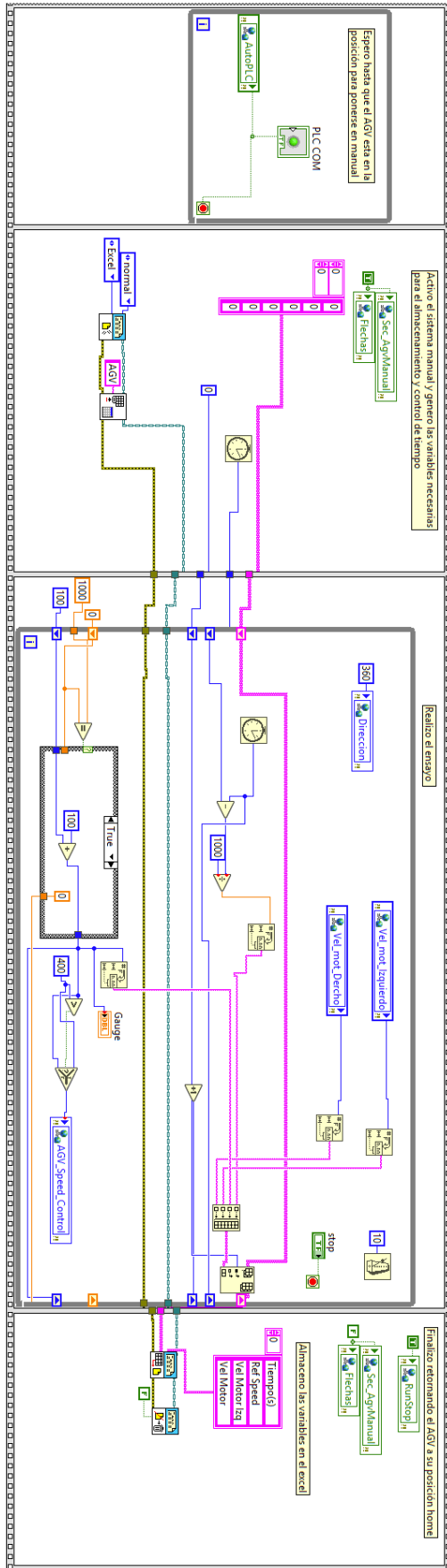
### 1.3 Motorren Modeloa

Motorrak sistema fisiko bat direnez, denborarekiko erantzun konkretu bat izango dute. Kasu berezi honetarako, AMR-ren motorren erantzuna kontrolagatik eta azelerazio aldaketekin denbora erantziari eragingo diote, jarrera konkretu bat adostuz. Edozein unean horrelako parametro bat moldatzen bada, lorturiko denboraren erantzunaren emaitza ez luke balio. Garapen honek modelo matematikoa hobetzeko asmoarekin burutzen da. Horretarako, datuak jasotzeko sistema bat diseinatu eta abiarazi behar da.

Hautatutako plataforma Labview da eta bertan abiarazitako programa baten bitartez abiadura kontsignak aldatzen ziren, gurpilen sentsoreen informazio jasoz eta Excel batean gordez. Labview eta Siemens arteko komunikazioa sortarazteko OPC-Server baten erreminta erabili zen. Halako zerbitzariak denbora errealean ezin dira abiarazi, hau da, 10 ms-ro funtzionatzeko gaitasunik ez daukate, baina froga egin zen unean, eskura zeunden baliabideak ziren.

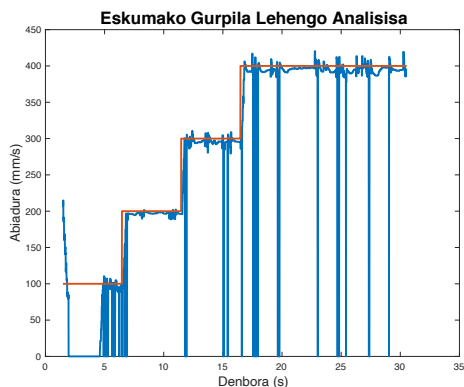
Labview-ko programa lau atal desberdinetan banandu zen, hauek bata bestearen atzetik abiaraziz. Sistema mota hau exekuzioa baldintzatzen du, aurrekoak prozesuak bukatu arte hurrengokoak ezin dira hasi. Programaren funtzionamendua ikusteko, prozesuak zatiz-zatiz azalduko dira. Lehengo prozesuen multzoan, AMR-earen eskuzko kontrola aktibatzen duen kontsigna martxan jartzen du. Prozedura honen bitartez, Labview-ek AMR-ren kontrola adosten du, softwareak emandako edozein agindua betez. Horretarako BlueBotics nabigazioak sortzen dituen kontsignak alde batera utzi egiten ditu eta AMR-ak mugikor batekin kontrolatuta dagoela pentsatzen du. Trikimailu honi esker Kivnon-eko kodea ez da aldatzen eta sortzen diren kontsigna guztiak eskuzko kontrolatik (mugikorra Bluetooth-ren bitartez abiarazten duen moduan) aginduak bidaltzen ditu. Trikimailua aktibatzeko Labview-ek zenbait aldagai aktibatu behar ditu, PLC barnean baldintza batzuk betetzeko. Aldagai hauek guztiak aktibatu arte, Labview-eko programa beste exekuzioak ezin ditu abiarazi.

Bigarren multzoan aldiz, gorde nahi diren aldagai guztien hasiera ematen da. Ahaztu gabe Excel artxiboa sortu behar dela, geroago datuak bertan gordetzeko eta ikertzeko. Hirugarrenean, datu bilketa ematen da eta datuak zuzenean Excel-en gorde beharrean, bektoreetan gordetzen dira. Horrela disko gogorrek eta Excel gordetze prozesuak sar ditzakeen atzerapenak ekiditeen dira. Horrez gain, abiaduraren aldaketak sortzeko funtzio bat planteatzen da, non denbora pasa hala, abiadura 100 mm/s handitzen da 400 mm/s arte. Abiadura aldaketa hauek eskailera izaera daukate. Behin datuen bilketa eginda, Excel-aren bitartez gordetzen dira eta AMR-ak hasierako posiziora itzultzeko agindua ematen zaio. Emandako azalpen guztiak Irudia 21-an laburbildu daitezke.

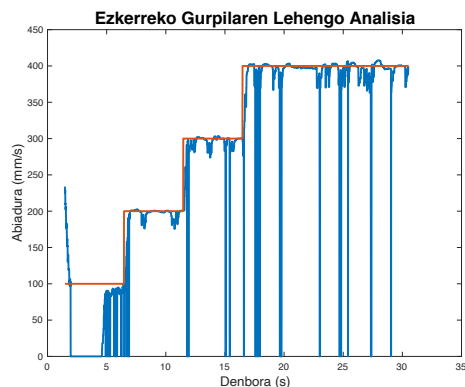


Irudia 21 Motorren modeloa lortzeko burututako programa

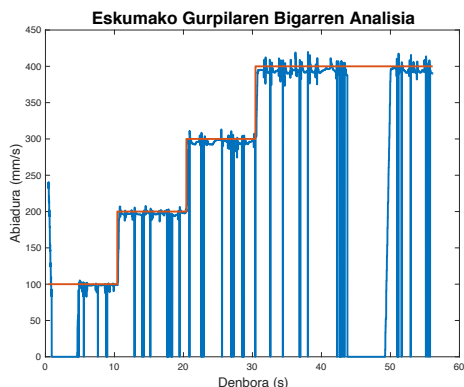
Excel programa bitartez, datuak grafikoetan planteatzeko aukera ematen du. Horrela era bisual batean arazoak aurkitu daitezke. Grafikoei esker, OPC-Server buruz sortutako zalantza adierazi zen, non zenbait kasuetan komunikazioa galerak agertzen diren eta galera horietan abiadura zero balioarekin adierazten dira (begiratu Irudia 22). Lehengo errepresentazioan informazio galerak ikusterakoan, bigarren frogapen bat egin zen. Horrek ere datu gehiago lortzeko aukera ematen zuelako. Grafikoetan ikusten den lerro gorria AMR-ak jarraitu behar duen abiadura kontsignaren errepresentazioa da.



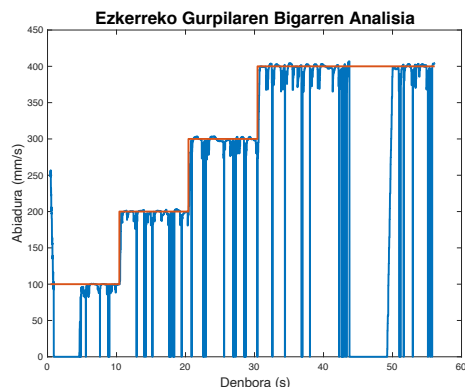
(a)



(b)



(d)



(e)

Irudia 22 Motorren erantzuna: a) eta b) lehenengo frogapenen erantzuna den, c) eta d) bigarren frogapenen erantzuna

Grafikoak aztertuz, lehenengo aldiunetan agertzen diren balioak esperimendutik at gelditzen dira. Kasu guztietan, AMR-ak abiadura nuluetatik hasten delako. Agertzen diren balio horiek OPC-Server-ren buffer-ean gordeko balioak zirela ikusi zen. Horregatik lehenengo segunduetan OPC-Server abiadura balio ez nulua jasotzen zituen. Behin segundo bat igarota robot mugikorak geldituta zegoela irudikatzen zuen. Horrek, halako komunikazioa ez erabiltzeko bi arrazoi ematen ditu. Lehengoa komunikazio erorikoak egoten direla eta bigarrena datuek atzerapen batekin heltzen direla. OPC-Server-rak errepresentazio grafikoak egiteko pentsatuta dagoela eta denbora errealetik at gelditzen den tresneria dela baieztatzen du. Dena den, Labview-ek beste aukera batzuk eskaintzen ditu.

Bigarren frogapenean 40 s eta 50 s artean, pertsona bat bide erdian jartzen da, AMR-ri geldiune bat sortzeko. Behin pertsona hori bidetik kenduz ostea, plataforma mugitzen hasten da, 0 mm/s-tik 400 mm/s-tara pasatuz. Abiadura aldaketa horrek gain bultzada bat eragin dezake, kontsignako abiaduraz pasaz. Berez, geldiune hori horrelako erantzunak ikusarazteko planteatu zen. Baina errealitatean halakorik ez da gertatzen, Kivnon barnean diseinatutako

kontrolagailuak, halako jarrerak ekiditeen dituzte azelerazio aldapak erabiliz. Horrek hasieran pentsatutako sistemaren erantzuna guztiz moldatzen du. Azken finean bigarren mailako sistema baten erantzuna izan beharrean, lehengo mailako jarrera daukalako. Azelerazio aldapak gain bultzada ekiditeen dutenez, bigarren mailako  $\xi$  parametroak bat balioa izango du, hau da sistema kritiko moteldua dela adierazten du. Lehengo mailako sistema maten parametroak lortzeko, abiadura kontsignako % 63 heltzen denean, zenbat denbora pasa den ikertu behar da. Horregatik Taula 4-an %63 irizteko behar duen denborak jasotzen dira.

Frogapenetik beste ondorio bat lortu zen. AMR-ren izaerarengatik, motor batek besteak baino abiadura handiagoa eraman behar du, zuzentasuna zuzentzeko. Lorturiko emaitzetan, zenbait kasutan motorrak abiadura kontsignatik pasatzen dira. Biraketa prozedura honek honela funtzionatzen du: Ezkerrera biratu behar baldin bada, eskuineko gurpilari abiadura kontsigna handiago ezartzen zaio. Beste kasua ematen baldin bada, alderantzizkoa gertatuko. Kivnon-ekoek plataforma mugikorraren noranzkoa jakiteko, motorren abiadura erabili beharrean, potentsiometro baten bidez kalkulatzeko dute.

Taula 4. Motorren frogapenen denbora erantzunak

Abiadura erreferentzia (mm/s)	Irudia 22 a) (s)	Irudia 22 b) (s)	Irudia 22 c) (s)	Irudia 22 d) (s)
100-200	0,28	0,33	0,215	0,215
200-300	0,225	0,272	0,225	0,2
300-400	0,275	0,275	0,25	0,25
Media	0,26	0,29	0,22	0,22

Sistemaren erantzuna kalkulatzeko, analisi bakoitzaren denboraren media hartuta datuak lortuko dira. A eta C kasuak eskuineko gurpilaren datuak dira eta balioa 0,24 s inguruko da. Beste datuak aldiz, ezkerreko motorenak dira eta 0,255 s balioa inguru ezar daiteke kalkulua. Kotsideratuz bi motorrak izaera berbera daukatela eta 0,015 s-ko diferentzia dagoela, bi motorrentzako 0,25 s-ko balio erabiliko da.

Sistemaren zatikizuna kalkulatzeko, ( 11 )-garren adierazpena erabiliz lortzen da. Zatikizunaren balioa 100 izango da, salto bakoitzeko 100 mm/s-ro egiten delako. Behi hori ebatzita, ( 12 )-arren ekuazioan lehengo mailako sistemaren ekuazioa adierazten da eta lorturiko balioak ordezkatzuz, motor bakoitzaren denborarekiko duten erantzuna irudikatzen da, La - Place planoan.

$$K = \frac{\Delta Y}{\Delta U} = \frac{200 - 100}{100} = \frac{300 - 200}{100} = \frac{400 - 300}{100} = 1 \quad (11)$$

$$Motor_{eskuma} = Motor_{ezkerra} = \frac{K}{Ts + 1} = \frac{1}{0,25s + 1} \quad (12)$$

Behin Kivnon-eko teknologia frogatuta izanda eta haren modelo matematikoa izanda, zenbait hobekuntza planteatu ziren. Lehendabizi oztopoak ekiditeko sistema bat burutu da, non LiDAR-eko informazioaz baliatuz eta eragiketa sinpleak erabiliz, AMR-ak objektuak ekidin dezake. Eragiketa hauek optimizazio prozeduretan ez daude oinarrituta, horrela modu erraz batean PLC batean programatu daiteke, haren kapazitatea murriztu gabe.

Printzipio hauek erabiliz, autoren lehengo erakarpena etorri zen. Gainera, bertatik ikasitako algoritmoaz, autorearen bigarren erakarpena egiteko aukera etorri zen ere. Bien kasuan LiDAR-ra sentore moduan erabiltzen da, baina bigarrenak optimizazio funtzio bat erabiltzeaz gain, sare neuronal batean oinarritutako garapen bat egiten da. Bi argitalpen hauek, zazpigarren atalburuan Autorearen ekarpenak atalean landuko dira.



## 1.4 Kivnon-ekin egindako bilerak

Tesi honek merkatuan dauden zenbait produktu erabili dituzenez, informazioa lortzeko zenbait enprekin bileren bitartez informazioa tartekatu da. Garapenean sortaraziko arazoak konpontzeko, buruzagiarekin eta zenbait teknikariekin bilerak prestatu ziren. Horregatik, atal honetan Kivnon-ekin jasandako arazoak eta konponbideak aztertuko dira. Hau da, atal honek informazio tekniko gutxi aurkeztuko du.

Lehenengo urtean zehar, AMR-ak zuen softwarea aldatzen joan zen, PLC barnean zituen funtzioen ikusteko gaitasuna murriztuz. Horrek Mercedes – Benz-eko proiektuaren aurrerapenei zailtasunak ezartzen zizkion, batez be Siemens-eko programan. Funtzioen programazioa ikustea ezin bestekoa da, segurtasun guneak izan ezik. Hauek makinaren funtzionamendua nolakoa den antzemateko baliogarria da. Jordi Carrera-rekin (departamentu teknikoaren arduraduna) zenbait emailen bidez, tresnaren informazioa tartekatu zen. Non elementuen informazioa jasotzeaz gain, nabigazioari buruz informazio gehiago bildu zen. Horrekin bat, mapa baten eraikitzerako orduan ze elementu eta non kokatu behar ziren azaldu zuten. Hortik aurrera teknikari honek informazio tartekatzeari utzi zuen eta berarekin harremana guztiz galdu egin zen, Mercedes-Benz-ko Gasteizko Laborategian planteatutako lehenengo eredu ikusi zuenean.

Bertatik aurrera Xavi Munells-ekin (departamentu tekniko zuzentzailea) kontaktuan jartzeko adierazi zigun. Berez, Xavi-k plataforma mugikor hau diseinatu eta burutu zuen pertsona da. Beraz, berarengandik informazio gehiago lortzeko aukera bazegoen. Denbora pasa hala, eskatutako informazioa heltzen ez zen eta kodearen eskuragarritasunaren murrizketaren azalpenak etortzen ez ziren.

Hala ikusita Mercedes-Benz barnean alarma guztiak piztu ziren eta Juan Prieto-kin (kivnon - eko zuzentzailea) bilera bat antolatu zen, urte osoan zehar agerturiko arazoak tratatzeko. Haien konfiantza erakusteko tesiaren aurkezpena egin zitzaion, lantegi barean beraien plataforma zertarako erabili nahi den aurrez-aurre jartzeko. Bilera honi esker, bi enpresen artean hitzarmen bat sinatu uztean, kodeko funtzio guztiak libre moduan utzi zituzten, proiektuaren garapenerako.

Laburbilduz, beste enpresen produktuak erabiltzerako orduan, lan filosofia ondo ezarri behar da, bestela haien teknologiaren babespena bilatuko dute, gero eta zailtasun gehiago jarritz.

## 2 Robot Kolaboratiboa

---

Proiektuak behar duen beste tresneria robot kolaboratibo bat da. Kasu honetan Universal Robot-eko robot kolaboratibo bat zegoen laborategian. Haren funtzionamendua ikusarazteko, enpresa barnean dauden beste proiektuak jarraitu ziren. Bertan segurtasunez buruzko gaiak landu ziren bati-bat. Enpresa honekin izandako harrera guztiz ezberdina izan zen. Enpresa hauek badakite robot batek ekoizpen prozesu batean ezartzeko, robotak manipulatu eta programatu behar direla eta horrela azaldu zuen César Gonzalvo-ek (Iruña zuzentzaile komertziala) Kivnon-ekin izandako arazo azaldu ostean. Horregatik, robot industrialen legedietan, robotak duen segurtasun ziurtagiriaz gain, lan espazioaren eta programaren ziurtagiriak behar dira.

Robot kolaboratiboen kasuan, tesia aurkezterako orduan, ISO arautegia aprobata gabe dago. Etorkizunean aprobatako den legedia jarraitzeko, ISO/TS 15066:2016 legedian adierazten diren aholkuak erabili behar dira. Legedi honek, ISO 10218-n oinarrituta dago, hau da, edozein elementu edo dispositibo robotiko industrialentzako legedia da. Erakarpenetan roboten egoera azaltzerako orduan, talkak nola identifikatzen diren aipatzen da. Berez, Robotika klasikoan lankideekin talkak saihesten dira. Horretarako, ISO 10218-ek segurtasun tresneria ezberdinen erabilpenak jasotzen ditu. Elementu hauen artean: LiDAR-ra, segurtasun hesiak eta atea, abiadura murrizketak etab. daude. Hala ere, legediak zehatz mehatz zer egin behar den aipatzen

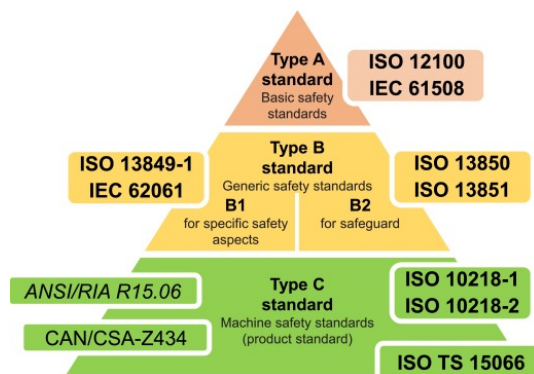
ez du. Horregatik, robotaren lan-eremua diseinatzen duenaren interpretazioaren arabera suspertzen da eta ziurtagiria lortzeko, entitate batek lan espazio hori behar dituen segurtasun elementu elektriko eta elektronikoko balioetsi behar ditu.

ISO/TS 15066:2016 legediaren arabera, robot-lankide sor daitezkeen talken ebaluazio planteatzen du. Bertan planteatzen diren mugapen ezberdinak direla eta, prozesu segurua dela ziurtatu daiteke. Horretako, PILZ entitateak halako ebaluazioak egiten dituzte. Proiektuaren garapen egoera jakiteko Toni Guasch-ekin (PILZ-eko robotikaren garapenetan aditua zen) elkarrizketa bat planteatu zen, non bai robot kolaboratiboen informazioa jaso zen, baita AMR ingurukoa ere.

## 2.1 Roboten ziurtagirien egoera

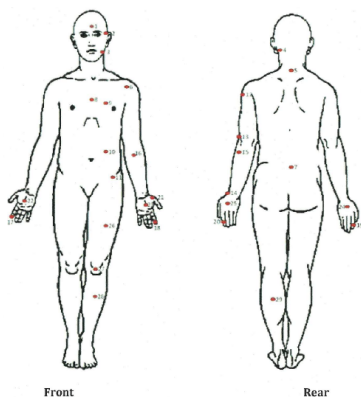
Atal honetan legedia betearazteko dauden ziurtagiri ezberdinak azalduko dira, azken finean laborategian tresneria frogatzeko gunean aurkitutako oztopo bat izan delako eta oztopo honek proiektuarentzako betebeharrak bat delako.

Segurtasun ziurtagiri barnean hiru motakoak daude. A motakoak oso legedi ahulak direnak, berez izaera informatiboa globala daukate, goitik ikusitako perspektibak jasoz. B-koetan aldiz, informazioa zenbait kasuetan zehaztu egiten ditu. Tresneriak edo beste edozein elementuak guztiz definitzen dituzten legediak, C motakoak dira. C motako legediak bai A-n, baita B-n agertzen diren aholkuak jasotzen dituzte eta legedi orokor horiek aplikazio bakoitzari izaera partikulara ematen dizkio. Irudia 23-an robot industrialentzako hiru legedien ereduak jasotzen dira.



Irudia 23 Robot industrialentzako hiru motako legedien ereduak [136].

Lehen azaldu den moduan, robot kolaboratibo barnean ISO/TS 15066 bete behar da. Arautegi honen barnean, gizakiaren gorputzean zehar jasan dezaketen talken eraginak azaltzen dira. Mugapen hauek lortzeko, pertsonekin kalte entseguak egin dira eta Irudia 24-ean legedian jasotzen den esperimenduaren emaitzak ematen dira. Aplikazioaren arabera eta robotak egiten dituen mugimenduen arabera, gorputzaren gune ezberdinetara hel daiteke. Horregatik, ebaluazio hau lan baldintzak eta eremua definituta dagoenean egin behar da. Tesi honek burutu nahi duen aplikazioagatik, buru inguruan sortzen diren talkak aztertu behar dira, gune arriskutsuenak direlako.



Body region	Specific body area	Quasi-static contact		Transient contact	
		Maximum permissible pressure <sup>a</sup> $P_s$ N/cm <sup>2</sup>	Maximum permissible force <sup>b</sup> N	Maximum permissible pressure multiplier <sup>c</sup> $P_T$	Maximum permissible force multiplier <sup>c</sup> $F_T$
Skull and forehead <sup>d</sup>	1 Middle of forehead	130	130	not applicable	not applicable
	2 Temple	110		not applicable	
Face <sup>d</sup>	3 Masticatory muscle	110	65	not applicable	not applicable
Neck	4 Neck muscle	140	150	2	2
	5 Seventh neck muscle	210		2	
Back and shoulders	6 Shoulder joint	160	210	2	2
	7 Fifth lumbar vertebra	210		2	
Chest	8 Sternum	120	140	2	2
	9 Pectoral muscle	170		2	
Abdomen	10 Abdominal muscle	140	110	2	2
Pelvis	11 Pelvic bone	210	180	2	2
Upper arms and elbow joints	12 Deltoid muscle	190	150	2	2
	13 Humerus	220		2	
Lower arms and wrist joints	14 Radial bone	190	160	2	2
	15 Forearm muscle	180		2	
	16 Arm nerve	180		2	

Irudia 24 ISO/TS 15066 legedian bildutako datuak, gorputzak jasan dezakeen talkak jakiteko

Naiz eta legedi berri honetan inpaktuen analisiaren aipamena sartu, ISO 10218-an robot bat kolaboratibo bilakatzeko neurriak adierazten dira. Bertan adierazten diren lau proposamenetik, gutxienez bat bete behar da. Planteatutako baldintzak honako hauek dira:

- **Segurtasun kontrola geldiunearekin:** Langile bat robotaren lan eremuan sartzerako orduan identifikatu behar da eta momentu horretatik aurrera langilearen posizioa kontrolatu behar da, robotaren lan baldintzak aldatuz. Pertsona oso gertu baldin badago, geldiunea segurua gertatuko da.
- **Eskuzko kontrola:** Metodo hau robota programatzeko erabiltzen da. Horretarako robotak egoera horretan dagoela jakin behar du. Hemendik aurrea langileak edozein ibilbide aldaketa egin dezake.
- **Abiadura aldaketa eta urrunketa:** Langilearen distantzia neurtuz aren abiadura moldatzen joaten da. Oso gertu dagoenean robotak ez ditu mugimenduak eteten, baizik eta motorren mugimendua oso murriztua egongo da.
- **Potentzia eta indarren murrizketa:** Indarren eta potentzia kontrola eginez, talkak aurreikusi daitezke, eta behi detektatuta robotak mugimendua eten behar du.

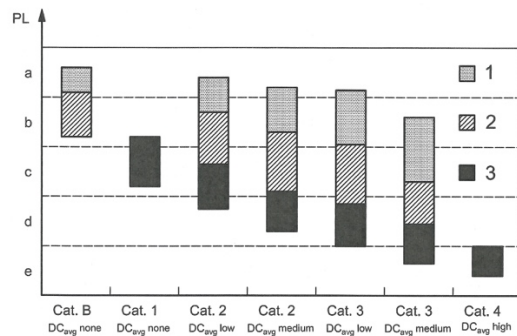
Dena den proposamen hauek betearazteko erabili behar diren sentsoreak ez dira aipatzen, baina gutxienez segurtasun sistema bat PL "D" 3 kategoriakoa izan behar du. UR-ren kasuan, haren segurtasun botoia honako baldintza hau betearazten du eta beste segurtasun elementuak PL "D" 2-koak dira. Sailkapena hau hurrengo kontzeptuetan oinarrituta dago:

- PL: Dispositibo bakoitza ordu bakoitzeko izan dezakeen arazo probabilitatea. Parametro honek letra batekin adierazten da, non A-tik E-ra sailkatzen den. Non A ordu bakoitzeko arazo gehienak izan dezakeen.
- Kategoría: Sistema eta azpisistema bakoitzaren eraikuntzaren izaera. Horretarako zenbakiekin adierazten dira.

Mercedes-Benz barnean dagoen teknologia arabera, segurtasun botoia legediarekin betetzen du eta horrek robota kolaboratibo bilakatzeko erabiltzen dituen indar pareen sentsoreetan beste kategoría bat ezar daiteke, haren produktua merkatuz. Azken finean potentziaren eta indarren analisia egiten baldin bada eta robotak talkak sortzerako orduan, Irudia 24-an agertzen diren baldintzak betetzen baldin badituzte, haren makina CE segurtasun ziurtagiria lortuko du. Berez, segurtasun kontuak direnez, PL "D" motako tresneriak erabili behar direla argi eta garbi gelditu behar da. Kategoría aukeratzeko orduan bi motetakoak izan daitezke, hau da, bigarren edo hirugarren kategoría bat hautatu behar da. Bien arteko

ezberdintasunak hardwareen komunikazio baldintzetan aurkitzen dira eta ez arazo bat izateko probabilitateetan. Katgoria handiagoan informazioa bikoiztu behar da, horrela sentsore bat komunikazioa erortzen baldin bada, informazioa tartekatzeke beste bide bat dago. Horregatik, hirugarren katgoriak honek robotak garestitu egiten du. Azalpen hauek DIN EN ISO 13849 - 1:2016-06 barnean aurki daiteke eta Irudia 25-an informazioa laburbiltzen da. Bertan ikusten den moduan, bigarren katgoriako sentsore on batek hirugarren katgoriako sentsore bat baino hobea izan daiteke. Hobeagoak dira, apurtzeko probabilitate txikiagoak dutelako.

Hiru sentsore mota aurkitu daitezke katgoria bakoitzeko. Beraz, enpresa batek adierazi dezake haren sentsorea PL "D" 3 katgoriako sistema duela, baina sentsore horrek bigarren mailakoa (Karratua inklinatutako lerroekin beteta) sentsorea izan. Horrek PL "D" 2-ko katgoria duten sentsoreak eta hauek hirugarren mailakoa (Karratua guztiz beteta) izanda, segurtasun aldetik, hobea izango dira.

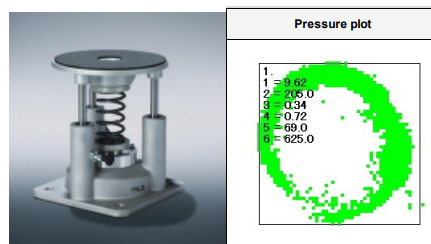


Irudia 25 DIN EN ISO 13849 - 1:2016-06 aipatutako segurtasun sistemen sailkapenak

Legediak nola interpretatu behar diren ikusita, ziurtagiria lortzeko beharrezko diren pausuak azalduko dira, konkretuki PILZ-ek emandakoak. Prozesu hau gertutik jarraitu zen, geroago tesian garatuko den tresnerian aplikatu behar zelako eta horretarako pausu guztiak ulertu behar zirela adostu zen.

Robot kolaboratibo baten ziurtagiria lortzeko gorputzean jasaten dituzten indarren entseguak azalduko dira. Indarrak aztertzeke tresneria konkretu baten beharra dago eta Irudia 26-an PILZ-ek erabiltzen duen tresneria aurkezten da:

- Indarren neurgailua: Sistema mekaniko batean oinarrituta dago, non desplazamendua inpultso elektrikoetan bilakatzen den. Tresneria honen mugikortasuna murrizteke malgukiak erabiltzen dira, non malgukiaren izaeraren arabera, gorputzaren guneak simulatu daitezkeen.
- Presio neurgailua: Enpresa honek paperezko markagailuak erabiltzen dituzte. Presioa eragiterakoan, bai indarrak eragiten duen azterna irudikatzen du, baita jasandako presioa gune bakoitzean kolore degradazioari esker. Software baten bidez, datuak interpretatu daitezke.



Irudia 26 Talkaren eragina aztertzeke tresneriak (Pilz enpresak eskainitako irudiak)

Dena den, merkatuan halako frogak egiteko beste proposamen batzuk daude. Adibidez, presio neurgailuak. Hauek beste era eraginkor batean neurtzeko erabili daitezke eta Pressure Profile Systems enpresak saltzen ditu. Konkretuki txantilo digitalak saltzen dituzte, presioaren ebaluazio zuzenean aztertuz, paperezko markagailuak erabili gabe.



Irudia 27 Pressure Profile Systems-ek proposaturiko beste presioa neurgailuak {4}

## 2.2 UR frogapenaren ondorioak

Universal Robot enpresa eskaintzen duen produktu guztietatik, UR-10 erabiliko da. Tesi hau aurkezterako orduan lau produktuetatik, masa handiak jasaten duten robotak aukeratuko dira, burutu nahi den aplikazioak indar – pare ahaltsoak jasan beharko dituelako. Robotak zenbat masa jasaten duen jakiteko, haien izenean duten zenbakia aztertu behar da. Adibidez, UR-10-ak hamar Kg-ko masak jasaten ditu. Bestalde, haren lan eremua zein den jakin behar. UR-10-rrren kasuan 1.300 mm helmena dauka eta burutu nahi den aplikaziorako aproposena da. Horrekin bat, sei artikulaziozko robota da eta lan eremua haren singularitasun ondorioz, murriztua gelditzen da. Momentu konkretu batean robotaren bi ardatzak lerrokatu egiten direnean, azelerazioen kalkuluak infinitura doaz, singularitasuna sortuz. Frogapenak burutu ostean, UR-10-rrren 1300 mm lan espazioa 1000-1100 mm murriztua gelditzen da. Beraz, helmena eta hauen lan espazioa kontuan hartu behar dira, aplikazioa garatzerako orduan. Hala ere, robotak AMR - ri lotzeko egitura mekaniko baten beharra dago eta hau edozein altaratukoa izan daiteke.

Horrekin bat, Iruñako enpresak robotak prozesu batean modu kolaboratibo pasa zintezkeela adierazi zuen. Baina aplikazioa nola burutu behar zela ez zuten burutu. Robota momentu jakin batean kolaboratibo pasatzeko, programa baten bidez landu behar zela ondorioztatu zen. Modu honekin, robotak posizio batera hurbildu daiteke, era automatiko batean eta azkeneko posizionamendu zehatzak gizakiak dezake. Hortik aurrera robotak puntu horrekiko kokatuta geldituko lirateke, bere lanarekin jariatuz. Dena den, modu hau Unibersal Robot barnean burutu gabe dagoenez, lehengo prototipoan azaltzen den gunean, datu gehiago adieraziko dira.

## 3 Bihurkinak

---

Proiekturako garatu nahi den mekanismoari erreminta bat atxiki behar zaio. Kasu honetan bihurtura prozesu bat denez, bihurkinak erabiliko dira. Hauek prozesuaren izaerarengatik, mota ezberdinetakoak izan daitezke. Bihurdura prozesu askotan, indar-pare konkretu batzuk eman behar dira eta hauek kasu normalean 50 Nm pasatzen dira. Horretako, tresneriaren espezifikokoak erabili behar dira eta laborategiaren kasuan Atlas Copco-ko bihurkinak dira.

Lehendabizi 70 Nm-ko bihurkinaren ezaugarriaz baliatuz, informazio bilketa bat egiten da. Erreminta honek 2,9 Kg-ko masa dauka, hau da, ia hiru kilo hauek robotak jasan beharko ditu. Horrek, robotaren kapazitatea murrizten du, 7,1 Kg-tara iritziz. Bestalde, bihurkin hori eusteko robotak atzaparra baten beharra du, beraz beste kilo bateko murrizketa kontuan hartu behar da. Bihurketa hazi aurretik robotak jasan dezakeen masa totala 6,1 kg-tara txikitu. Beraz, 61 N -eko gaitasuna dauka gutxi gora behera. 70 Nm-eko indar pare eman nahi baldin bada

eta bihurkinaren luzera 0,533 m-koa baldin bada, 131 N-ko indarra eragingo dio robotari. Indar hori segurtasun parametroak gaindituko zituzkeen robotak geldiaraziz. Azken finean 13 Kg talka bat jasaten duelako eta berarentzat gizaki batekin inpaktu bat izan duela adieraziko du. Momentu honetan, halako indarrak aplikatzeko, maneatzailer mekanikoen bitartez egiten dira, langileak halako talkak ez jasotzeko.

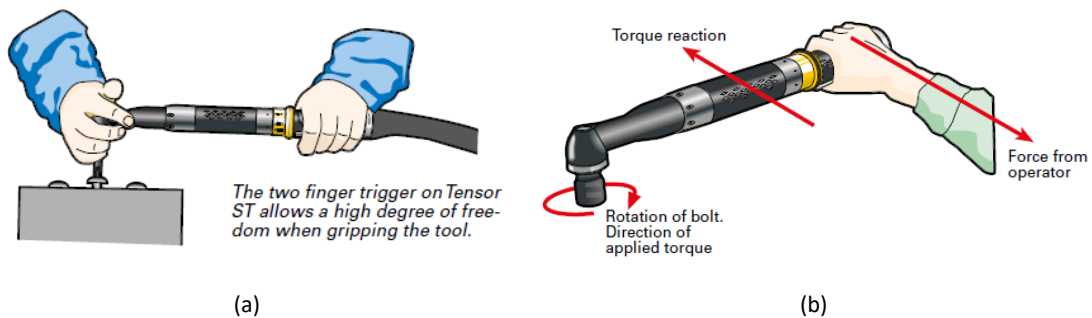
Halako arazoak ekiditeko, beste teknika batzuk badaude. Teknika hauek bihurtura prozesuan sortzen diren indarrak murrizten dituzte. Atlas-Copco-ko erremintak aldatu behar dira, batez be izaera mekanikoa aldatu behar delako. Sistema mekanikotik elektronikora pasatu behar da. Sistema elektronikoa horren bitartez, inpaktuak murriztu daitezke. UR-ek bihurkin elektronikoa hauekin 140 Nm-ko indar pareak ematea lortu dute eta Iruñan Iruñako lantegian halako aplikazioaren erakusketa batean konfirmatu zen.



Irudia 28 UR-ko 140 Nm bihurtura prozesua {5}

Atlas Copco-ko ikerketa baten ondoren, indar pare altuko bihurturak ekonomiarekin lotuta dagoela adierazten du. Batez be langile batek esfortzuaren ondorioz lesionatu daiteke, hau da, lan postuaren ergonomia faltaren ondorioz, enpresari zenbait kontra puntu dakarkio. Adibidez, birgaitze ordainketak, ordezkoaren ordainketa eta efizientzia eta kalitate galerak neketasunarengatik. Ondorio hauek, tresneriei esker murriztu daitezke.

Langileak halako eragiketarak egiterakoan, eskuen posizioa ezinbestekoa da eta ezarritako aholkuak errespetatuz ez gero, efektu txarrak ekar ditzake. Irudia 29 (a)-an agertzen den moduan, bi eskuak erabili behar dira indar parean emateko, non ezkerreko eskua torloju gertutik jarri behar den eta eskuinekoa aldiz, bihurturaren katuan ezartzen den. Alderantziz egiten baldin bada, indar pareak eragiten duen kontrako indarrak, besoak harrapatu ditzake.

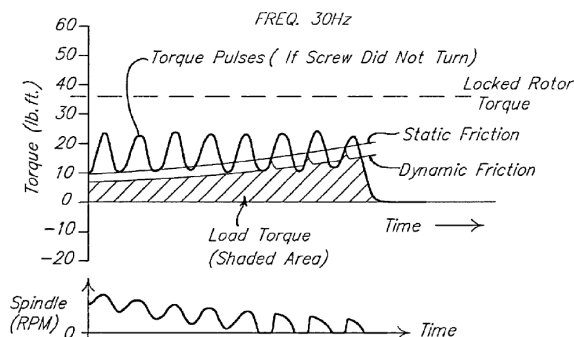


Irudia 29 Indar pareak aplikatzeko eskuen posizionamendua {6}

Irudia 29b-n langilean edo bihurkinaren euskarria jasan behar dituen indar pareak agertzen dira. Gero eta biraketa puntutik urrunago egonda, orduan eta eusteko indar txikiagoak egin behar dira. Ekuazio batean irudikatu izanez gero, ( 13 )-arren bezala adieraz daitezke.

$$\text{LangileIndarra} = \frac{\text{IndarParearearenErreakzioa}}{\text{Distantzia}} \quad (13)$$

Lehen aipatu den moduan, bihurkin elektronikoak lankideak jasaten duen indarra murriztu daiteke eta ESTIC mota honetako bihurkinak ekoizten ditu. Indarraren murriztapena inputso elektromekanikoekin aplikatuz lortzen da. Berez, bihurkinak denborarekiko indar-parea aldakortasunari esker, torlojuari nahi den indar pare aplikatu ahal zaio, indar erreakzio minimoa jasanda.

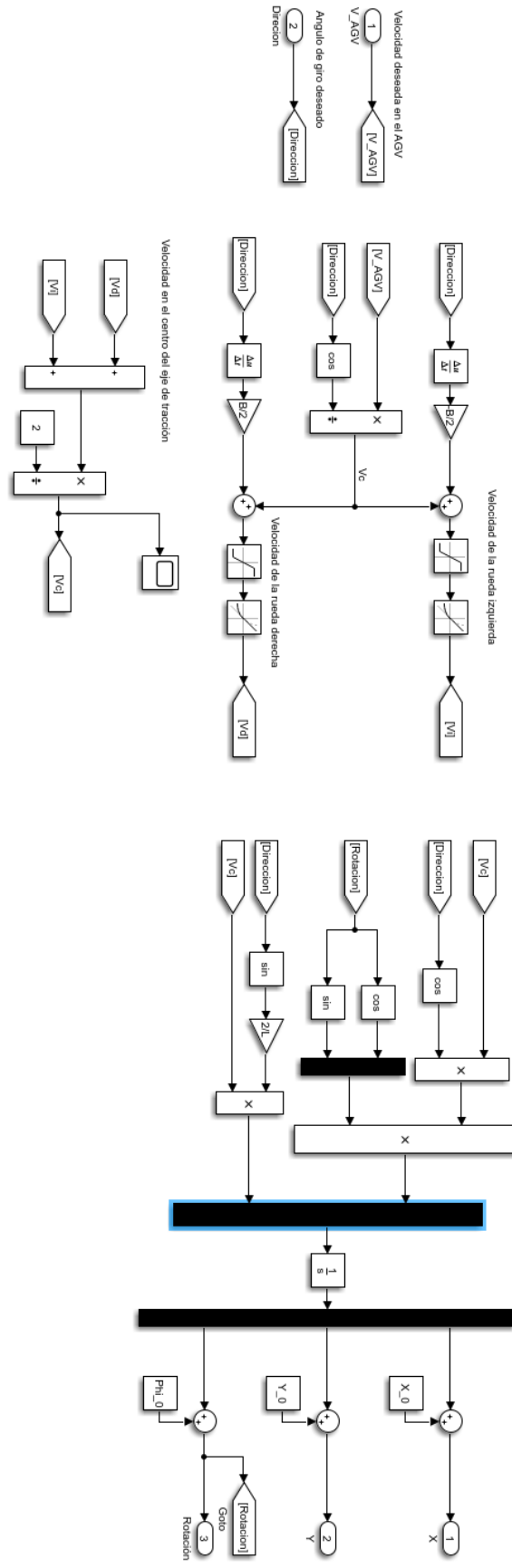


Irudia 30 ESTIC proposaturiko bihurkari elektronikoa {7}

## 4 Simulazioak

Behin proiektuak erabiliko dituen erreminta guztiak aztertuta, ordenagailuak ematen dituen abantailak aztertzeko garaia da. Asmaketak prototipoetan ezarri baino lehen, simulagailu batean eraiki dira. Berez, halako teknologiak ingeniartzako frogapen prozesuak murriztu ditu, batez be simulazioak prozedura bizkorrak eta merkeak direlako. Halako tresneriak, 4.0 industrian ezin bestekoak dira eta kivnon-eko garapen matematikoa aprobetxatuz, Simulink - en bitartez froga-gunea garatu zen.

Ezkerreko aldean, simulazioaren lehengo etapa aurkitzen da, non abiadura eta noranzko balioak gordetzen diren. Abiaduraren eta noranzkoaren deskonposaketa egin behar da, horretarako bigarren zatia dago. Bereizketa egin ostean, motor bakoitzari aren abiadura agindua bidali al zaio. Motorren aginduetan, abiadura eta azelerazioa mugatu behar dira, Kivnon-eko plataformak dituen izaerari egokitzeko. Horrek denborarekiko erantzun konkretu bat adostuko du eta X-ren, Y-ren eta  $\gamma$ -ren balioak denborarekiko aldatuz. Horregatik, simulazio blokeetatik  $V_c$  berriz kalkulaten da. Behin balio berri hori izanda, AMR posizioa estimatu daiteke eskuinean agertzen diren blokeekin.



Ezkerreko Aldea

Irudia 31 Kivnon eredu matematikoaren simulagailua



#### 4.1 Puntu finko batera mugiarazi

Simulagailu honekin planteaturiko lehenengo garapenari dagokio atal honek. AMR-ak puntu finko batera mugiarazteko garatutako algoritmo bat da. Plataforma mugiarazteko hurrengo ebazpenak jarraitu dira:

$$\Delta x = X_{End} - X \quad (14)$$

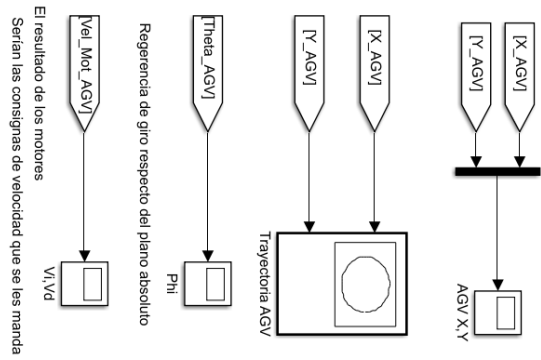
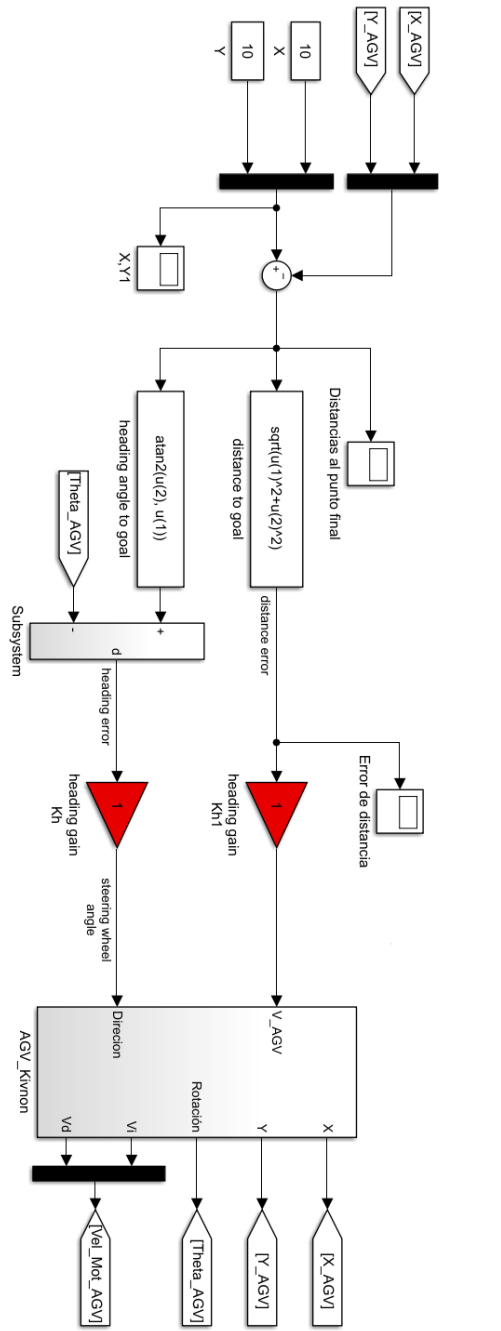
$$\Delta y = Y_{End} - Y \quad (15)$$

$$E_{distantzia} = \sqrt{\Delta x^2 + \Delta y^2} \quad (16)$$

$$E_{angelua} = \text{atan}\left(\frac{\Delta y}{\Delta x}\right) \quad (17)$$

$$\Delta_{angelua} = E_{angelua} - \theta \quad (18)$$

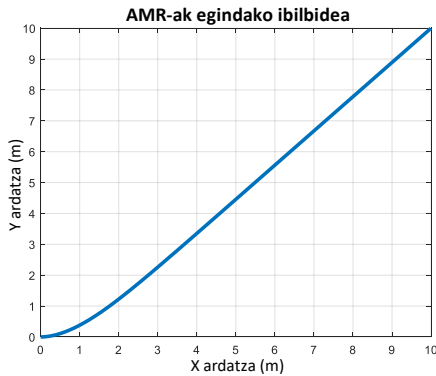
Adierazpen hauek laburbilduz, inteligentziaren funtzionamendua ikus daiteke. Helmugara irizteko gelditzen zaion distantzia etengabe ebaluatzen da ( 16 ). Horrez gain, AMK-ak zuzentzeko desbiderapen bat estimatzen da ( 18 ). AMR-aren modelo birarazteko, bai distantziaren ebaluazio, baita desbiderapenaren estimazioa garaitze baten bidez biderkatzen dira (ikusi Irudia 32).



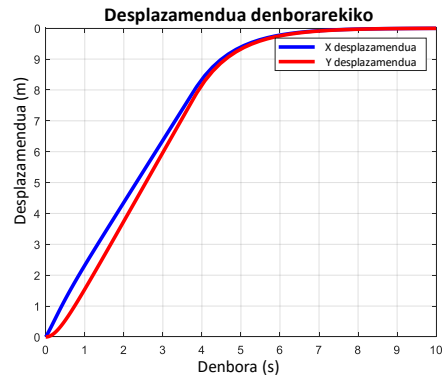
El resultado de los motores Serían las consignas de velocidad que se les manda

Irudia 32 Kivnon eredu matematikoaren simulagailua

Erantzuna Irudia 33-an jasotzen da, non a-n ibilbidearen informazioa lor daiteke. Kasu honetan  $X = 10$  m eta  $Y = 10$  m posiziara mugiaraztea agindu zaio eta b-n ibilbide hori lortzeko AMR-ren posizio aldaketa denborarekiko nola moldatzen den ikus daiteke.



(a)

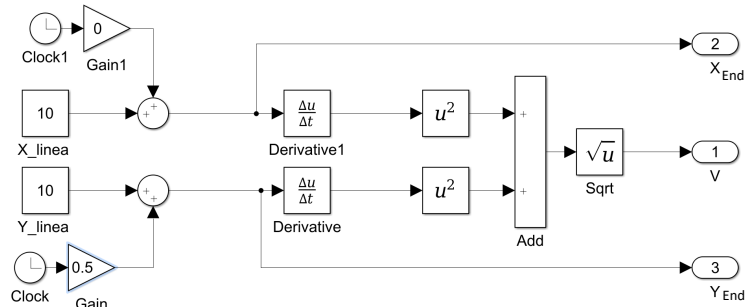


(b)

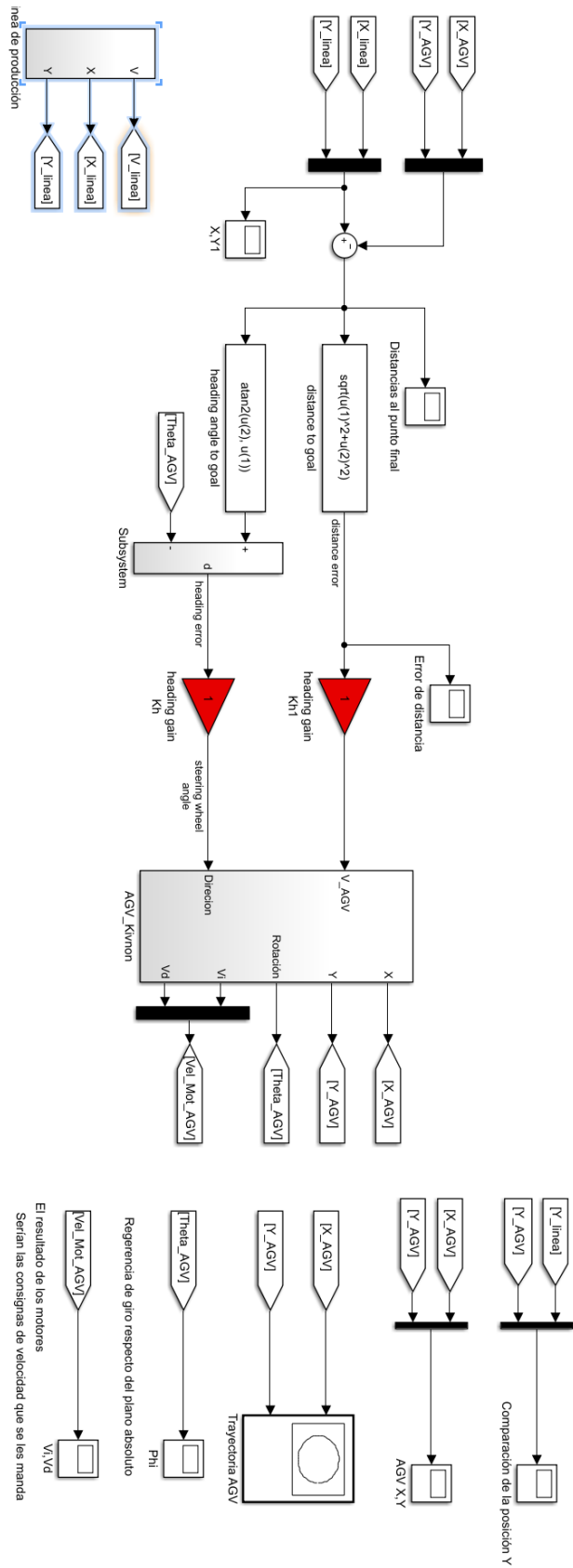
Irudia 33 Puntu finko batera mugiarazteko simulazioa: a) Ibilbidearen erantzuna, b) X eta Y errepresentazioa denborarekiko

#### 4.2 Mugikor den puntu batera mugitu

Bigarren ebazpen honek, AMR-ak objektu mugikor bat jarraitzen duen ikusi nahi da. Aurreko atalaren ekuazioak erabiliz,  $X_{End}$  eta  $Y_{End}$  denborarekiko izaera aldakorra ezarri zaizkie. Horrela, Mercedes – Benz-eko produkzioaren mugikortasuna simulatu daiteke. Bestalde, AMR-ak puntu mugikor batera nola mugiarazten den aztertu daiteke. Amaierako puntuen mugimendua sortarazteko, funtzio bat planteatu da. Bertan segundo bakoitzeko distantzia konkretu bat amaierako puntua mugiarazten du.



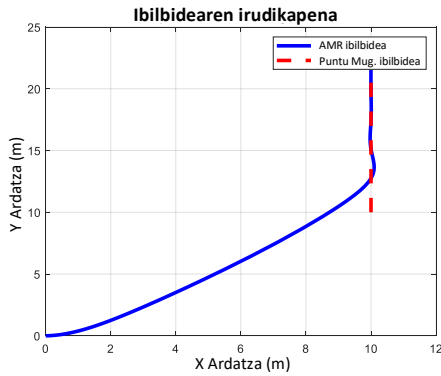
(a)



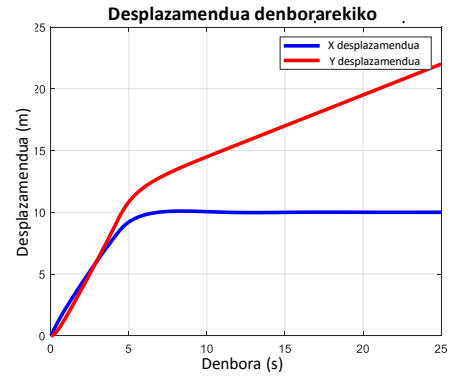
(b)

Irudia 34 Puntu mugikorraren programa: a) Amaierako puntu mugikorraren funtzioa, b) Simulink-eko kodea

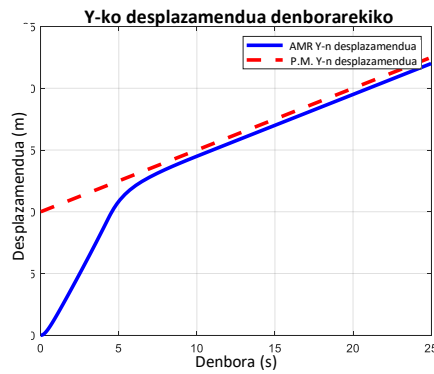
Kasu honetarako AMR  $X = 0$  m eta  $Y = 0$  m posizioan hasiko da, baina lan eremua  $X_{\text{End}} = 10$  m eta  $Y_{\text{End}} = 10$  m kokalekuan dago eta  $Y$  ardatzean segundo bakoitzeko  $+ 0,5$  m mugitzen da.



(a)



(b)



(c)

Irudia 35 Puntu mugikorraren erantzuna: a) AMR-ren eta puntu mugikorraren ibilbidea, b) AMR-ren X eta Y posizioaren aldaketa denborarekiko, c) AMR-ren Y aldaketa denborarekiko puntu mugikorrarekiko konparatuz

Irudia 35a-n ikus daitekeen moduan, lehengo etapan puntu mugikorraren noranzkoa darama, amaierako puntua harrapatzeko. Bigarren etapa  $Y$  ardatzean puntu mugikorra jarraitu egiten du eta b - n irudian a-n interpretatutako mugimendua denborarekiko ikus daiteke. Bertan  $Y$  etengabe aldetzen dela argi eta garbi ikusten da. Seigarren segundotik aurrera AMR-ak  $X$  ardatzean posizioa mantentzen du. Emaitzak ikusita, plataforma mugikorrak puntu mugikorra jarraitzen hari dela adierazi daiteke eta puntu mugikorraren posizioa jakinda, sistema hau etengabeko prozesu batean lan egiteko aproposa da.

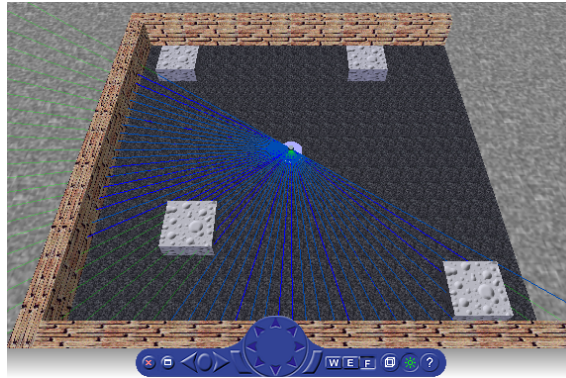
Dena den, jarraipen algoritmo honek bigarren mailako sistema bat denez, errore nulua lorrarazteko integratzaile bat behar du. Kontrol honek bigarren mailakoa izategatik aldapako kontsignak jarraitzerako orduan errore bat sortarazten du, Irudia 35c- n ikusten den moduan.

### 4.3 3D simulagailua

Ingeniariak, matematikariak, etab. grafikoak ikustera ohituta daude eta era erraz batean zer gertatzen den imajinatu dezakete. Baina erakusketa baterako, eremu birtual batean aurkeztea erakargarriago izango da. Tesi honek “*Simulink 3D animation*” bitartez lan espazio bat diseinatu du, non errepresentazio birtualez gain, Simulink barnean dauden blokeekin informazioa tartekatu dezake. Hau da, bertatik lortzen den emaitza ez da bideo bat izango, Simulink-eko blokeak eremu horrekin informazioa tartekatzen dutelako. Beraz, Simulink-eko garapena eremu birtual horrekin guztiz integratuta dago. Eremua virtual dituen oztopoak identifikatzeko zenbait sentsore ezar ahal zaizkio. Esterako, LiDAR, kamerak, etc.

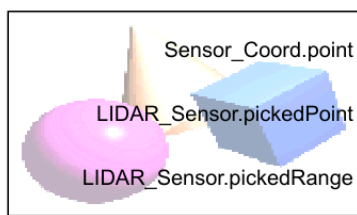
Badaude beste plataforma batzuk “*Robot Operating System*” (ROS) bateragarriak direnak, baina horrek beste software baten erabilpena suposatzen zuen. Horien artean Gazebo aurkitu daiteke.

Lan eremua diseinatu baino lehen, MatLab-en dauden ereduak landu dira eta tesiarentzako “*Differential Wheeled Robot with LiDAR Sensor*” gehien hurbiltzen dena da. Bertan robot aspiragailu forma duen plataforma mugikorra aurkezten da, non goiko eremuan LiDAR bat erabiltzen duen. Sentsore horrekin lan eremuan dauden objektuen posizio erlatiboa jakin dezake eta oztopo saihesteko algoritmo batekin haren ibilbidea aldatzen du oztopora hurbildu bezain laster.



Irudia 36 *Differential Wheeled Robot with LiDAR Sensor*-ren ereduaren lan eremua

Irudia 36-an simulazio eremua irudikatzen da, non kasu honetan LiDAR-rra  $-\pi$  rad – tik  $\pi$  rad – ra tarteko neurketak egiten ditu. Izpi urdinak luzera markatzen du eta oztopo bat aurkitzerakoan LiDAR-ak atzeko oztopoak ezin ditu ikusi. Horregatik, kolore urdinetik berdera pasatzen da. Sentsorearen informazioa bloke batetik irteten da( ikusi Irudia 37).



VR Source

(a)

```
val(:,:,1637) =
    0.1385    0    -2.2011
    0.2573    0    -2.0365
    0.3818    0    -2.0016
    0.5440    0    -2.1186
    0.7658    0.0000    -2.3570
    0.8739    0.0000    -2.2072
    0.9736    0.0000    -2.0690
    1.0666    0.0000    -1.9401
    1.1542    0.0000    -1.8187
    1.2375    0.0000    -1.7032
    1.3174    0.0000    -1.5924
    1.3947    0.0000    -1.4852
```

(b)

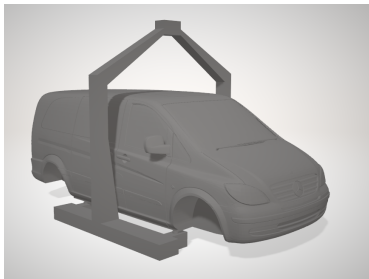
Irudia 37 LiDAR sentsorearen Simulink-eko blokea: a) blokearen errepresentazioa, b) PickedPoint-en datuen bilketa.

Informazioa “*LIDAR\_Sensor.pickedPoint*” eta “*LIDAR\_Sensor.pickedRange*” ateratzen da. Lehengoak puntuen koordinatuak ateratzen ditu, hau da, izpia mozterako orduan ze posiziotan eman diren izpi mozketak adierazten ditu.

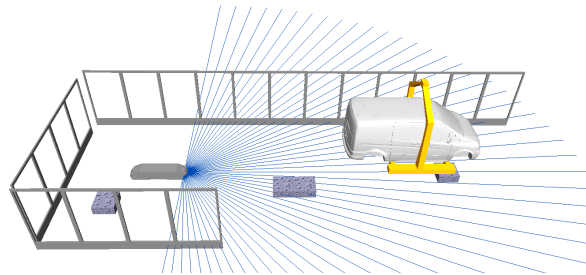
LiDAR-ak beti puntuak irudikatzen dituela, gogoratu behar da. Simulazio txiki bat eginez, irteten diren datuak ikus daitezke. Berez, datu horiek hiru zutabetan antolatzen dira eta balio horien unitatea metroa izanda. Zutabe bakoitzak X, Y edo Z koordinatuari dagokio eta ikus daitezkeen moduan, Y ardatza altuera irudikatzen du, balio guztiak nuluak direlako. Orduan, robot mugikorra X eta Z espazioan mugiaraziko da.

Bigarren kasuan aldiz, LiDAR-ak antzeman dituen tartea ikus daitezke eta lortzen diren emaitzak zutabe bakar batean irudikatzen dira. Zutabe horretako lerro bakoitzak argi izpi konkretu bati egiten dio erreferentzia eta izpi batek ezer identifikatzen ez badu, -1 balioarekin beteko da.

Lan eremu hau, proiektu beharrei moldatu behar da eta horretarako zenbait objektuen diseinua egin behar da. Horien artean, Kivnon-eko AMR-a, Mercedes – Benz-eko furgoneta eta produkzio guneetan aurki daitezkeen esekigailuak moldeatuko dira. Esekigailua amaierako muntaketan furgoneta gurpil gabe dagoenean uzten duen tresneria da (begiratu Irudia 38a ). Erremintan txertatzeko diseinu horiek STL formatuan gordeko dira.



(a)



(b)

Irudia 38 Proiektuaren simulazioaren lan eremu birtuala: a) Furgoneta eta esekigailuaren diseinua, b) lan eremu birtualaren errepresentazioa.

Irudia 38b-n Mercedes – Benz barnean planteatutako diseinua aurkitzen da eta horretarako, bai laborategiko hormak, baita kutxatxo batzuk gehitu dira eremuan, oztopoak saihesteko algoritmoak burutzeko. Egindako elementuak sartzerako orduan, haien izaera kontuan hartu behar da. Azken finean, paretak ez dute mugikortasun izaerarik eta hauek programan solido zurrunkak direla adierazi behar dira. Eragiketa hau egiten ez bada, LiDAR-ak ezarritako oztopoak ez ditu ikusiko. Furgoneta eta esekigailua aldiz, gutxienez izaera mugikorra izan behar dute. Elementu hauek espazioan zehar mugiaraziko dira, puntu mugikorra izango direlako. Simulazio honi esker eta puntu mugikor proposatzen duen algoritmoa garatuz, autorearen lehengo argitalpena burutzeko baliagarria izan da.





# IV.

## Atalburua

Gizakiaren etengabeko garapena guztiz asmaketaren menpe dago; bere adimen sortzailearen produktu inportanteena delako -Nikola Tesla-

### Aplikazioaren burutzea existitzen denarekin

---

---

Teknologiak frogatu ostean, nora eraman daitekeen eta ze aplikaziotan funtzionatu daitezkeen begiratu behar da. Horretarako, muntaketa amaierako prozesuen lan-baldintzak ezagutu behar dira. Zientzia munduan, tesi baten iraupena hiru urte ingurukoa izaten omen da eta normalean aurrerapenak pausuz pausu egiten dira. Baina eremu industrialean, batez be Mercedes –Benz barnean ikusitakoaren arabera, hiru urte horietan aldaketa asko gerta lekieke kontuan hartu behar da. Adibidez, produkzio gune berriak sortu daitezke, merkatuak behartzen diren produktuak ekoizteko, beste langune berri batzuk eraikitzen edo moldatzen dira, etab. Horrek, etengabeko aldaketak erakartzen ditu. Dena den, kasu gehienetan azterturiko lan egoerak denboran zehar mantendu egiten dira eta postuaren ergonomiaren analisia egiteko aukera ematen du, prozesu horretan garatu nahi den robota ezartzeko onurak balioetsiz.

Lan baldintza horiek jakinda eta proiektuaren betebeharrak buruan izanda, ergonomia txarra duten postuen analisia planteatu da. Bihurdura tresnerietan azaldu den moduan, halako atalak tratatzerako orduan, lankidearen konforta eta osasuna soilik ez da tratatzen. Halako postuen mugapen ondorioz, zenbait atal landu daitezke. Besteak beste: Ergonomiaren kostua, lortzen den kalitatea, sartu behar den ordezkapenak, langileen absentismoak, gizartean gertatzen hari den aldaketa demografikoa, postuaren segurtasuna eta mugaketak dituzten pertsonen eskuragarritasuna kontuan hartuko dira. Horregatik, proiektu honek ergonomiaz gain, beste aspektu hauek tratatuko dira.

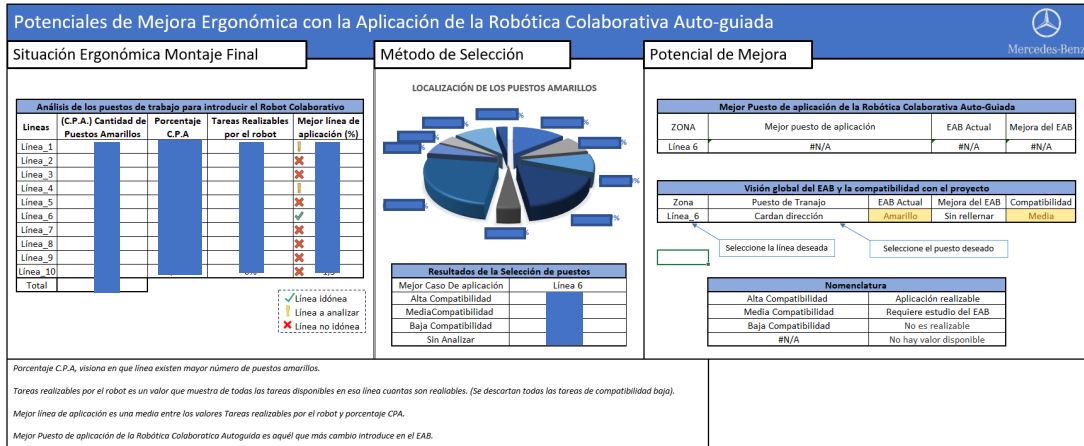
Atal honetan irudimenari eskua ematen zaio, ez soilik identifikatzeko kalteko postuak, bazik eta postu horietan plataforma nola sar litekeen aztertzeke. Gune honetan errealitatetik at lor daitekeen automatizaketaren inguruko ideiak aurkituko dira eta hurrengo atalean ideia horiek errealitatean txertatzeko landuko dira.

#### 1 Garatutako plataforma

---

Nondik hasi behar den betiko arazoa da. Kasu honetan, argi eta garbi enpresa barneko datu bilketatik hasi ber da. Eta datuak lortzeko, departamentu espezifiko batera joatea ezinbestekoa da. Berez, Mercedes – Benz barnean postuen analisiak eginak dituzte, baina datu horiek proiektuarentzako baliagarriak bilakatzeko, Excel-en bitartez programa bat burutu da. Bertan datuak sartu bezain prest, era automatiko batean enpresa osoaren egoera aztertu daiteke. Irudia 39-n aplikazioa hiru sekzioetan bananduta dagoela ikus daiteke. Lan prozedura lerrotan sailkatuta dago eta lerro bakoitzean lan-eremu ezberdinak daude. Horregatik ezkerreko sekzioan, begirada batez amaierako muntaketan lantegiaren egora osoak ikus daitezke. Gainera, robota non sar litekeen adierazten du. Lantegiko lerro bakoitzeko eremu onak, aztertzeokoak edo aproposak diren zehazten du. Erdiko sekzioan, datuak era grafiko batean aurkezten dira. Grafikoaren azpian ergonomia aspektu txarrak dituzten postuak taulan agertuko dira eta

aztertutako postuetatik tesiaren garapena sartzeko zenbat postu aproposak diren azalduko da. Eskuineko eremua, postuz postu egoerak aztertu daitezke.



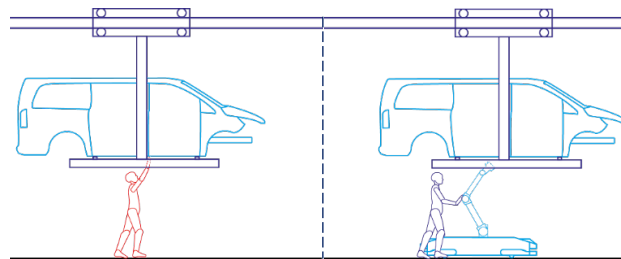
Irudia 39 Ergonomia sailkatzeko burututako Excel programa.

Irudia 39-n agertzen diren karratu urdinak, informazioa estaltzeko ezarri dira, baina plataformaren erabilpena ikusarazteko baliagarria da.

## 2 Languneen azterketa

Tesian garatutako tresneria honekin, robot kolaboratiboa nabigazio autonomoa ezartzeko bi lan-lerro identifikatu dira. Lehengoa seigarren lerroko eremua da eta bigarren laugarren lerrokoa da. Seigarren lerroa ergonomia aldetik prozesu ezeroso asko ditu. Ebaluazioa egin eta gero, programak lehengo aplikazio gunean izan liratekeela adierazi zuen. Aldiz, laugarrenean postu bakar bat identifikatu du.

Seigarren lerroan zehar, kotxearen azpitik hainbat eragiketa egiten dira, non kasu askotan bihurtura prozesuak egiten diren (begiratu Irudia 40).

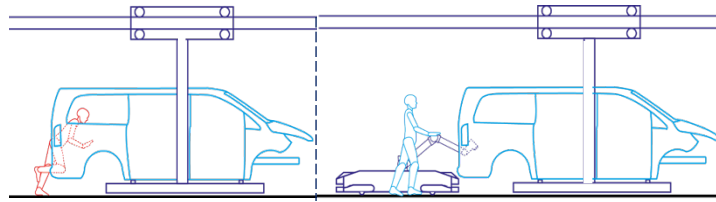


Irudia 40 Seigarren lerroaren lan egoera

Egoera ikusarazteko lan baldintzak irudikatu behar dira, batez be langileak jasaten dituen esfortzuak ulertzeko. Berez, bihurkinak 3 – 4 kg arteko samakoak dira eta eragiketak egiteko hauek beti sorbaldatik gora mantendu behar dira. Lan postu honen baldintzak sorbaldan, lepoan eta besoetan arazoak sortarazi dezake. Horregatik, langileen postu aldaketa ezinbestekoa da. Jasan behar duten masaz gain, torloju bati indar pare espezifiko bat eman beharko balitz, bihurkinak aztertzerako orduan ikusi den moduan, besoak indar bat jasango du.

Lan postu honen egoera, robotarentzako lan baldintza egokiak izan daitezke, erabilitako Kivnon-eko plataformaren tamaina handia delako eta furgoneta goiko aldetik joaten baldin bada, maniobratzeko eremu libre gehiago izango duelako.

Laugarren lerroan aldiz, eserlekuak furgonetara finkatzeko eta posizioz aldatzeko riel bat ezarri behar da, Irudia 41b-n agertzen den elementua da.



(a)



(b)

Irudia 41 Laugarren lerroaren lan egoera: a) Lan baldintzen errepresentazioa, b) Furgonetak erabiltzen dituen riel-ak (Mercedes-etik lortutako irudia)

Irudia 41a-n agertzen den moduan, langileak torlojuei indar pare konkretu bat emateko furgoneta barnean sartu behar da. Halako egoeratan bizkarrezurra eta gerrialdeko guneetan langileak gain kargak izan dezake. Bihurkinak sortzen dituen indarrak zuzenean besoek jasate dituzte. Proiektuko robota erabiltzen baldin bada eta lan postuak sortzen dituen mugapenak gainditzeko, plataforma berezi bat diseinatu beharko lirateke. Behin robota furgoneta barnean sartuta, bere eragiketak egin lezake.

Mercedes – Benz eskaintzen dituen konfigurazio ondorioz, hautatutako ekipamenduaren arabera furgonetaren sailkapena egiten da. Hau da, kasu batzuetan furgonetek bederatzi eserleku dituzte, beste zenbait kasutan eserleku gabeko furgonetak fabrikatzen dira. Ekipamendu honi esker, furgonetaren ordena eralda daiteke, lan postu bakoitzaren kargak murrizteko. Adibidez, bederatzi eserlekuko furgoneta bat etortzen baldin bada, hurrengo furgoneta eserleku gutxiago izango ditu. Horrek lanpostuaren ergonomia hobetzen du. Beraz, robota erabiltzen baldin bada, furgonetaren fabrikazioan sortzen den nahastea desagertu daiteke, logistikaren planteamendua murriztuz.

Enpresa barnean badaude beste zenbait postu ergonomia aldetik onak direnak, baina arlo ekonomiko aldetik txarrak. Batez ere, furgonetari balio erantzia ezartzen ez duten languneak direlako. Adibidez kalitate azterketak. Ibilgailuan zehar zenbait neurrien kontrola egiten dira, kotxeari ezartzen zaizkion piezak behar den eran dauden jakiteko. Momentu honetan, zenbait ikerketa postuak, lan espazio bat hartzen dute. Beraz, espazio hori era eraginkor ez da batean aprobetxatzen. Garatu nahi den erremintarekin espazio optimizatu daiteke, kalitate ikerketa muntaketak egiten diren bitartean burutu daitekeelako.

Kalitate kontrol honek, furgonetaren elementu guztiak behar bezala muntatuta dauden adierazten du eta zeozer txarto atxiki baldin bada, berrikuste gunera eramaten da. Bertan beste langile batek detektatuko akatsak zuzentzen dituzte. Robot kolaboratiboa nabigazio autonomoarekin, argazkien bitartez kontrola ezberdinak egin litzake, zer, non eta noiz eman den arazoa jakiteko. Arazo hauek guztiek furgonetaren historialean gorde litzateke, ustekabe batean kontsultatzeko. Egoera hauek ikusita, tesiaren lan-eremua seigarren lerroan ezarriko da, non lanak ibilgailuaren azpitik egiten diren. Horrekin bat, aplikaziotarako zenbait abantaila dituelako. Horien artean, AMR-rentzako esandako lan eremu gehiago dago eta altuera konstanteko prozedurak direnez, ikuspena kalibratu daiteke modu erraz batean.



# V.

## Atalburua

Gizakiak horma asko eraikitzen ditu eta ez behar diren zubiak -Isaac Newton-

### Beharren hausnarketa

---

---

Orain arte ikasitakoaren arabera eta teknologia frogatu ostean, Isaac Newton ezan zuen moduan, hormak eraikitzeari utzi behar dugu, zubiak eraikitzeke momentua delako. Atalburu honetan etengabeko lanetan robot kolaboratibo nabigazioa autonomoarekin sartzeko, landu behar diren atalak hausnartzen dira. Hau da, AMR-ekin, robotarekin, lan baldintzekin eta integrazioarekin zer ikusia duen informazioa bilduko da. Egingo diren deskribapenen ondorioz, amaierako prototipoak behar lituzkeen garapenak, egin beharrak eta integrazioa aurkituko dira.

Hau ez du esan nahi prototipoen hausnarketak egin eta gero, beste helburu batzuk suertatuko ez diren. Baizik eta, hemen tratatutako ideiak era globalean lantzen dira eta baliteke arazo baten aurrean soluzioa era lokalean landu behar izana.

Planteatzen diren hausnarketa guztiak, tresneria aldakorrei buruz, segurtasunari buruz, fidagarritasunari buruz, lortu behar diren prozedurak azaltzen dira. Azken finean atal hauek proiektuaren helburuei atxiki behar zaizkie, lortuko den tresneria industrializatzeko aukera emanez.

#### 1 AMR-rentzako puntu mugikorraren erreferentzia

---

Teknologiak aurkezten dituen AMR-ak eremu ez aldakor batean mugiarazteko pentsatuta daude. Aldiz, burutu nahi den ibilgailuak langune aldakorretan mugiarazteko ahalmena behartzen du. Horrek posizionamendu ez zehatza izatea eragin dezake eta posizionamenduaren erreferentzia ezin bestekoa da, bihurtura prozedura bat egingo delako. Edozein erreferentzia galera izanda, bihurtura prozedura behar bezala egiteko aukerak murrizten ditu. Halako egoerak ekiditeko, AMR-ak duen posizionamenduaz gain, beste sistema bat beharko du.

Bigarren erreferentzia sistema honi furgoneta jarraitzeko sentsore bat atxiki beharko lirakeke. Erreminta dela eta, bihurtura prozeduraren menpe egongo da, beraz burutzen diren lanen arabera indar-pare gehiago eragiten dituen bihurtura ezarriko da. Aldi berean erreminta honek bigarren erreferentzia sistema honekin espazioa kokatu behar du eta haren zehaztasuna zentimetro erdi baino gutxiago izan beharko da. Baliteke BlueBotics-eko nabigazio sistemak bigarren posizionamendu honekin galtzea, horregatik AMR-ren nabigazio sistemak motorren desplazamenduaren informazioa etengabe eskura izango du.

Egoeraren arabera eta merkatuak eskaintzen dituen produktuen arabera, plataforma mugikor baten diseinua egitea planteatu daiteke. Kasu horretan honako ideia hauek erabili daitezke diseinuarekin hasteko. Segurtasuna legediaren arabera, robot industrialetan LiDAR baten erabilpena gomendatzen da, batez be sistema horiek PL "D" 3-ko segurtasun dispositiboak direlako. Halako segurtasun parametroak lorrarazteko, segurtasun egoera batean errele baten bitartez motorren frenoak aktibatzen dira, kontuan izanda errele horrek bi kablez elikatuta egongo direla aktibazio agindua bi bide ezberdinetik bidaltzeko. Errele hauek, segurtasun

erreleak deritzote, eta segurtasun elektronika baino merkeagoak dira. Dena den, segurtasun LiDAR-rak zehaztasunarekin ez datoz bateratuta. Horregatik bestelako sentsoreen erabilpena planteatu daiteke, adibidez: kamerak edo 3D LiDAR-rak.

Garatu nahi den prototipoak aplikazio industrial batentzako burutu nahi denez, gune horietarako pentsatuta dauden tresneriak erabili behar dira. Horrek PLC edo Industrial-PC (IPC) erabiltzera behartzen du. Lehengo dispositiboa, Kivnon bezalako plataforma batean pentsatzen baldin bada ezarriko da, baina bigarrenak (IPC-ak) aldakortasun gehiago planteatzen ditu, bati bat prozesadore ahaltsuak eta Sistema Operatibo (SO) konbentzionalak dituztelako. Beraz, aplikazioak diseinatzerako orduan C++, Python, etab. erabili daiteke programa burutzeko. Bluebotics-en ezaugarriak ikusita, SLAM algoritmoak konplexuak dira, non optimizazio funtzioak prozesadorearen baliabide gehienak erabili behar dituen, posizionamenduaren kalkuluak egiteko. Gauza berbera gerta lekiok kameraren erabilpenarekin, batez be *“Pattern Recognition”* edo *“Deep Learning”* erabiltzen baldin badira. Halako inteligentziak erabiltzen baldin badira, PLC-ak erabilgarri diren dispositiboetatik at geldituko lirarteke.

Diseinu estetikoa bigarren plano batean gelditzen da, non mantentze prebentiboak eta prediktiboak egiteko aukera ematen duten diseinuak erabili behar dira. Horretarako, enpresentzako eskuragarri diren materialak ezarriko dira. Naiz eta diseinu konplexua badirudi, nahiko elementu sinpleak eta ohikoak erabiltzen ditu. AMR-ren gorputza egitura metaliko batekin osatu daiteke. Mugimendua sortzeko bi gurpil eta bi motor beharko dira. Prozesadore bezala IPC-a ezarriko da eta elementu guztiak elikatzeke bateriak erabiliko dira. Baina elementu hauek AMR autonomo bat bilatzeko, burutu behar diren algoritmoak eta programazioak ezin bestekoa dira eta ingeniari-tza ordu gehienak makinaren inteligentziaren garapenean inbertituko dira.

Intelligentziaren garapena zenbait zalantzak dakar, berez AMR-ei prezisio altuko posizionamenduak izatea behartzen zaizkio, baina gizakiak produktuak eskorjetan garraiatzerakoan ez dute halako zehaztasunik. Gizakiek kapazitate nahikoak dituzte gutxi gora behera korridore batetik mugiarazteko oztopoak saihestuz. Beraz, nabigazioa egiterako orduan, hau da, A puntutik B puntura abiarazterako orduan, errorea  $\pm 5\text{cm}$  izan lirarteke (lan baldintzen arabera), baina errore hori kargak uzterako orduan murriztu beharko da, gizakia egiten duen moduan. Orduan A eta B puntuetan, non lan zamak ezartzen zaizkion guneak diren, zehaztasuna handitzeko sistemak ezarri ahal zaizkio.

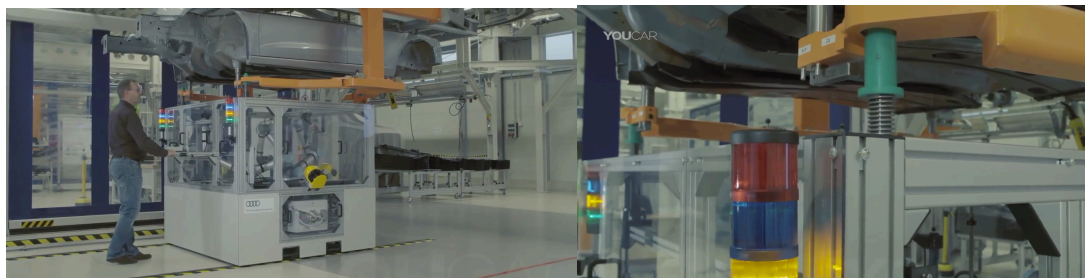
## 2 Robot Kolaboratiboa-rentzako erreferentzia

---

Edozein robota edo mekanismo produkzioko lanetan erabiltzeko, bihurkina edo beste erreminta bat eramango du. Horrek, furgonetan eragiketak egiteko zehaztasun minimo baten beharra dagoela kontuan hartu behar da. Azken finean, Mercedes – Benz ezarritako muntaketa kalitateak lortzeko asmoa dagoelako. Bi egoera planteatu daitezke. Egoera merkeena, merkatuko robot kolaboratiboa bat erabiltzea datza. Bertan AMR-ak eta robotak sistema independenteak dira. Robotak haren erreferentzia sistema independentearekin, plataforma mugikorrek sortzen dituen zehaztasuna galerak zuzenduko lituzke. Bigarren kasuan AMR-a diseinatzen baldin bada, egin daitekeen planteamendua da. Non AMR-ak eta robotaren ardatzak interpolatuta daude sistema berebean, era eraginkorrago batean erremintaren posizionamendua burutzeko. Bigarren kasu honetan, kontrolagailuak robota edo plataforma mugikorra abiarazi behar duen determinatzen du. Lehengo kasuan aldiz, AMR gune batera irizterakoan robotak lanean hasi daiteke, bien artean komunikazioa egon gabe.

Erremintaren posizionamendua furgonetarekiko burutzeko hiru aukera daude. Lehena, elementu mekanikoen erabilpenetik dator. Robotak haren posizioa furgonetarekiko jakin dezake sistema mekaniko batekin haren egiturara finkatzen baldin bada. Horrelako proposamenak beste lantegietan ikus daitezke, non bideratzaile mekaniko batzuei esker, robotak esekigailura finkatzen da. Horrela solido zurrun baten moduan jokatuko du eta

esekigailua mugiarazten baldin bada, roboten egitura ere. Zuzentasuna mantentzeko lurrean riel batzuk ezartzen dira, trenbide modura. Aplikazio konkretu honetarako, gizakiaren jokabidea ezinbestekoa da. Hasiera batean lankideak esekigailuarekin abiadura sinkronizatzen du. Esekigailua eta roboten egitura lerrokatuta duenean, eragile mekanikoak aterako behar ditu. Halako prozedurak gizakiaren inteligentzia aprobetxatzen dira, prozesuaren eragiketa desberdinak egiteko (Ikusi Irudia 42).



Irudia 42 Robot kolaboratiboen erreferentzia sistema mekanikoen bidez (Audi Smart Factory-ek aurkeztutako argazkiak)

Horrelako aplikazioak burutzen baldin badira, garestiak eta aldakortasun gutxiokoak kontsideratu daitezke. Tresnari horien diseinuan prozesu konkretu horri lotuta dagoelako eta halako egitura osatzeko, ingeniari-tza departamentuak diru interbertsioa altua egin beharko luke.

Bigarren proposamena, erreferentzia kolaboratiboaren garapenetik dator. Inteligentzia artifiziala eta ikusmen artifiziala erabili beharrez, langilearen inteligentzia erabiltzen da, furgonetarekiko erreferentzia puntu bat ezartzeko. Gizakiak robotak era kolaboratibo batean gidatu dezake bi eskuekin, maneatzailer baten moduan. Horrela robotak puntu konkretu batera eramane dezake, geroago robotak lanean jarraitzeko. Halako prozedura erabiltzeko, ARM-ak furgoneta era lerrokatu batean jarraitu beharko luke, bien arteko erreferentzia etengabe mantenduz. Ahalako batean, posizio aldaketa ematen baldin badira, robotak gizakiak ezarritako erreferentziatik at gedituko litzateke.

Azkeneko hausnarketa, ikusmen artifizialean oinarritzen da. Kamera batekin robotak furgonetarekiko posizioa etengabe jakin dezake. Ustekabe batean robotak posizio aldatzen baldin bada, robotak lan eremu berri horretara moldatuko da. Halako teknologien erabilpena software eta hardware bereziak erabiltzera behartzen dute. Dena den, aurkeztutako aukeratik eraginkorrena izango litzateke ustekabe baten aurrea. Tesian garatuko den robot + AMR lan postutik aldatzen baldin bada, era eraginkor batean erreferentzia moldatu daiteke. Beraz, sistema malgua dela adierazi daiteke.

Beste sentso motak batzuk erabiliz, planteamendu berriak sortu daitezke. Baina enpresa barneko esperientziagatik, hausnarketa nagusitik aldendu ziren. Adibidez: LiDAR baten erabilpena, furgonetaren hasierako eta bukaera non dauden zehazteko; Ultrasoinuen bitartez LiDAR-raren aplikazio berbera burutu daiteke; etab. Kasu guztiak eremu bakoitzerako Irudia 72 - an biltzen dira.

Hala ere, robot kolaboratiboa plataforma mugikor batean muntatuta dagoenez, honek erreferentzia puntualak hartzeko, robot klasikoak erabiltzen dituzten osagaiak erabili daitezke. Azken finean AMR-ak produkzio lerroak duen abiadura jarraitzen baldin badu eta sinkronizatzen baldin bada, robotak egoera kuasi-estatiko batean lan egiten du. Egoera horretan lan egiteko ustekabe gutxi egon beharko lirateke eta abiaduraren sinkronizazioa galtzen baldin bada, robotak erreferentziatik at egongo litzateke.

Errealitatean beste gauza bat gertatzen da. Erabiliko den AMR-re egituraren ondorioz, egoera kuasi-estatiko mantendu ezin da. AMR-ak planoko ardatz batean errore nulua lortzen duelako (adibidez X ardatzean). Beste ardatzean sistema azpi moteldu baten jarrera adostuko

du eta egoera kuasi-estatikora irizteko metro asko igaro beharko lirateke. Ulertarazteko, demagun Kivnon-eko AMR-ak denborarekiko mugimendu sinusoidala egiten dituela Y ardatza zuzentzeko, Irudia 19-an adierazten den moduan. Mugimendu oszilatzaile hauek biraketa puntuaren kokalekuaren arabera sortzen dira eta hauek gero eta txikiagoak izango dira, kontrolagailuak errorea murrizten duelako. Murrizketa hori eman arte Y ardatzean balore finkorik ez dira lortuko.

Merkatuan halako egoerak ekiditeko beste planteamendu batzuk daude. Kuka-k Mekanum gurpilak erabiliz, plataforma mugikorraren ardatz guztiak era erraz batean kontrolatuko ditu. Kuka - ko plataforman mugimendu sinusoidal hori ekidin egiten du. Horretarako biraketa puntua planoan zehar edozein lekutan ezar dezake. Horrek edozein biraketa egiteko aukera ematen dio, AMR-ren posizionamenduan zehaztasuna handituz. Azken finean, posizionamenduaren hobekuntzak izaera mekaniko eta fisikoengandik dator.

### 3 Sinkronizazioa

---

Sinkronizazioari buruz hitz egiterako orduan, abiaduraren sinkronizazioaz pentsatu daiteke. Baina langune bakoitzaren arabera lan-baldintza konkretu batzuk egongo dira, non elementu edota langile bakoitzarekiko komunikazioa sortu beharko den. Beraz, edozein prozesu automatiko batean gertatzen den moduan, proiektuan garatuko den robotak informazioa beste elementu edota langileekin tartekatu behar du, prozesuen egoera ezberdinak adierazteko.

Proiektutik zehar aipatu den moduan, plataforma mugikorrak robot kolaboratiboarekin produkzio abiadura eraman behar du eta prozesuen arabera abiadura hori eralda daiteke. Egoera bakoitzaren baldintzak jakiteko, lantegiko produkzio egoeraren informazioa eguneratuta izan behar du. Eskaera hau bi formaz egin daiteke, bata WiFi-ren bitartez, bestea Plug & Play izaerako sistema bat integratzea eta abiaduraren informazioa tesiaren robotak lineatik har dezan. WiFi bidezko komunikazioak erabiltzean baldin badira, haien izaeragatik abiaduraren kontrola egiteko ez dira aproposak. Informazioa tartekatzerakoan atzerapenak sor daitezkeelako, erabiltzen diren hardwareak denbora errealean ez duelako funtzionatzen. Plug & Play sistemek aldiz, sentsoreen bitartez bai posizionamendua, baita abiadura neur daiteke. Horrela tresneria zuzenean produkzio mailan jar litekeen instalazioetan aldaketa gutxi eraginez.

Proiektuaren plataforma, lankideekin komunikatu behar da, haren egoerak ezberdinak etengabe azaltzeko. Automatika munduan semaforo izaera duten argiak erabiltzen dira. Hiru kolorez baliatuz eta argi finkoak edo keinadekin zenbat kode sortaraz daiteke. Adibidez, kokapen kolaboratiboa erabiltzen baldin bada, robota modu horretan adierazteko argi berdez irudikatu dezake eta erreferentzia ematen denean, kolore gorria ezar daiteke. Horrekin bat, pantailak erabili daitezke, informazioa era erraza batean adierazteko. Zenbait kasutan, monitoretara begiratzea arriskutsua izan daiteke. Batez ere, begirada produktiotik at dagoelako eta halako jarrerak oharkabetasunak eragin ditzake. Horregatik, prototipoa aurrea doanean, elementu hauen erabilpena finkatu beharko da eta egoeraren arabera elementu ezberdinak erabiliko dira. Beraz, planteatzen diren egoeren arabera elementu ezberdinak proposatu beharko dira.

Laburbilduz, sinkronizazioa planteatzen denean, lan balditzen arabera planteatu beharko dira. Dena den, Plug & Play izaerako tresneria batera joan beharko da. Produkzioan dauden instalazioen programazioak edo sentsoreak ez aldatzeko, inteligentzia guztia plataforma mugikorrean izatera behartuz.



## 4 Integrazio digitala

---

Proiektu honek garatuko duen prototipoa, 4.0 industria arloarekin zer ikusia dauka. Berez, industria egoera honek informazioaren azterketetan oinarrituta dago eta horretarako egoeren informazioa jaso eta tartekatu egin behar da. Planteatutako tresneriak sentsore ezberdinak erabiliko dituzenez, bertatik lor daitekeen informazioa gorde eta aztertu daiteke. Lorturiko informazioa hiru eratarata erabili daiteke:

- **Kalitate arloko** datu bilketa egin daiteke, informazioa ibilgailuaren muntaketa historialean sartuz. Arazo baten aurrean, entitate ezberdinen artean informazio hori tartekatu daiteke. Horrek errorean jarraipena egiteko aukera ematen du eta furgonetak enpresaren kalitate estandarrak betetzen dituen hala ez jakin daiteke momentuan.
- **Intelligentziaren artifizialaren** erabilpena lan prozedura ezberdinetan. Ikusmen artifiziala sartuz gero, "*Pattern Recognition*" edo "*Deep Learning*" teknikak erabiliz, aldakortasunaren helburua landu daiteke. Betez be, ingeniariak klasikoak erabiltzen dituzten ebazpen matematikok ertzak, lerroak, etab. lortzeko lan eremu konkretu batzuetara baldintzatuta daude.
- **Fabrikazio prediktiborako** baliagarria izan daiteke. Batez be automatizatu gabeko bihurketa prozedurak, guztiz kontrolaturiko prozesuak bilakatzen direlako. Tesia aurkezten den huenean, bihurkinak haren kontrolagailua erabiltzen dituzte eta bihurkinak emandako indar pareak kontrolatzeko baliagarriak dira. Baina robotak tresneria hau kontrolatzerako orduan, bihurkinak prozeduran nola kokatzen den kontrola daiteke, piezen higadura berdina izateko eremu guztietan.

Ikusi den moduan Industria 4.0 barnean zenbait atalak landuko dira. Berez proiektu berri bat hasten denean, lantegiko zenbait gune modernizatzeko aukera ematen du. Sartzen diren berritzapen guztiak kalitatea, segurtasuna, kostuen murrizketa, malgutasuna, etab. lantzen dituzte.



# VI.

## Atalburua

Etorkizuna emaitzak erakutsiko ditu eta bakoitzak lortutako lorpenengatik epaituko zaio -Nikola Tesla-

### Beharrak gauzatu eta prototipoak burutu

---

---

Atal honetan proiektu osoan zehar egindako prototipoak eta aurkitutako beharrak azalduko dira. Denbora igaro hala, ikerketak egiteko Unibertsitatean AMR baten garapena proposatu da. Horregatik, atal onetan aurkeztuko diren prototipoetan bi multzoetan sailkatuta daude. Lehengo multzoan unibertsitate barneko AMR-ren garapena aurkeztuko da eta bigarreanean Mercedes – Benz-en garatutako robot kolaboratibo nabigazio autonomoaren prototipoa aurkeztuko da. Atalburu honek, Otto Scharmer-ren bi lan filosofiak bateratzen dira, azken finean beharrak eta prototipoen garapenak oso lotuta daudelako. Atalak bananduta baldin badaude, garapenaren jarraipena nahastu daiteke.

AMR-ak zientziarako planteatzen dituen erronkengatik, Gasteizko Ingeniaritza Eskolan halako prototipo baten muntaketa burutu da. Tresneriaren diseinu mekanikoa Argolabe Ingeniaritza parte hartzearekin buru da. Horretarako, bai Argolabe baita eskola dituen baliabideak erabili dira. Plataforma honi esker, zenbat ikasle barne nabigazio inguruko arloak landu dituzte, bai gradu amaierako lanen bitartez, baita master amaierako lanen bitartez. Azken finean, ahalko produktuen garapenak eskura izateak, industria barnean dauden ekipoen ezaugarriak ikasteko, diseinatze eta programatzeko aukera ematen dituzte. AMR industrial gain, merkatuan dauden beste teknologiaz baliatuz, beste robot mugikor ezberdinak planteatu dira. Hauek irudimenari eremu librea uzteko eta industriak sortarazten dituen hormak zubiak bilakatzeko erabili dira.

Mercedes – Benz-eko prototipoan aldiz, laborategiko baliabideen bitartez eraiki da, non elementu gehienak eremu industrialentzako prestatuta dauden. Kivnon-ek saldutako AMR-ren egiturari zenbat eraldaketa egin zaizkio, abiaduran eta noranzkoaren kontrola egiteko, baita roboten erreferentzia lantzeko. Konjuntu mekanikoaren zehaztasuna hobetzeko, zenbait elementu mekaniko diseinatu izan behar dira, baina malgutasunaren eta aldakortasunaren filosofiaz mantenduz. Denda den, zenbait tresneriak lan eremuen ezaugarrietara moldatu behar dira eta horrek aldakortasunaren helburutik pixka bat aldetzen da.

Lehendabizi unibertsitatean garatzen hari den prototipoari buruzko informazioa burutuko da, azken finean AMR garapenak tratatzen direlako. Aldiz, Mercedes – Benz-eko garapena AMR-ari erreminta konkretu bat atxikitzen zaio produkzio mailan bihurtura prozedurak egiteko.

# 1 Unibertsitateko prototipoak

---

Unibertsitatean Kivnon-eko AMR aztertzerako orduan nabigazio autonomoa burutzeko zenbait galdera sortu ziren. Produktu honek komertziala izateagatik eta Mercedes – Benz-eko lantegitik ateratzea ezinezkoa denez, ikerketak eta alderantzizko ingeniari-tza aplikatu ezin da. Halako egoera batean Gasteizko Ingeniaritza Eskolan eta Argolabe Ingenieria S.L.-ren laguntzarekin, AMR baten garapena burutu du. Prototipoa 2018-ko martxoan hasi zen eta hurrengo lerroetan prototipo honen deskribapena burutuko dira eta urteetan zehar izandako ondorioztapenak aipatuko dira.

## 1.1 Unibertsitatearen Prototipoaren Definizioa

Prototipoak garatzerako orduan, bere lan eremuak finkatu behar dira, lan baldintzen arabera elementu ezberdinak beharko direlako. Hala eta bezala, kasu batzuetan prototipo finalak izango dituen elementu guztiak eduki ezin dira eta beste batzuk bilatu behar dira. Kasu hauetan bilatzen diren elementuak edo tresneriak, osagai industrial bezala funtzionatuko dute, baina lan baldintza hobetuz batzuetara diseinatuta daude. Berez ahalako garapena zabal batean, prototipoak izango dituen elementu guztiak ezin dira definitu, baina horien artean zeintzuk diren susmatu behar dira.

Halako tresneriak industria eremuetarako prestatuta egon behar dira, horrek AMR-ren diseinua baztertzen du eta funtzionalitatea, osagaien eskuragarritasunaren, etab. alde balantza inklinatzen da. Prototipo honek behar baino espazio gehiago izateko diseinatuta dago, azken finean elementu elektriko, elektroniko eta mekaniko kopurua definitu gabe dagoelako. Prototipoak inteligentziak garatzen diren bitartean, hauek pixkanaka-pixkanaka definitzen joaten dira. Horrekin bat, prototipoaren fabrikazio prozesuan pentsatu egin da eta haren diseinua era eraginkor batean muntatzeko eta desmuntatzeko erraztasunari lehentasuna eman zaio ere.

Industria eremuetan lan egingo duenez eta tresneria automatikoekin komunikatzeko ahalatasuna behar duenez, IPC edo PLC baten beharra dago. Etorkizunera pentsatuz eta Bluebotics-eko informazioa ikusiz, ordenagailu baten beharra dago. Ordenagailu horren CPU -aren ahalmena momentu hasierako uneetan definitu ezin da. Batez bezala, erabiliko diren inteligentziak eta hauek burutu behar dituzten kalkuluak ezin direlako jakin. Halako egoera batean ahalmen handiko prozesadore bat erabiltzea egokiena izango lirateke, baina elementu horretan diru guztia inbertitzen baldin bada, beste ataletan diru murrizketak egon daitezke. Beraz, Beckhoff fabrikatzailearekin CPU konkretu bat adostu zen.

Merkatuen dauden aurrekariak ikusita, badirudi AMR guztiak LiDAR bat erabiltzen dutela segurtasun moduan. MiR-en kasuan, lokalizazioa egiteko LiDAR-ra erabili beharrean kamerak erabiltzen dituzte. Elikatze iturriaren definizioa ezinbestekoa da. Horretarako kontsumoen kalkulak jakin beharko lirateke, baina osagai guztiak definituta ez daudenez, baterien sistemen kapazitatea zehatz mehatz definitu ezin da. Horrek edozein elikatze iturria erabiliko denik ez du esan nahi, baizik eta parametro hori ez dela aipatuko. Baina karga nola egingo diren definitzea ezinbestekoa da, erabiltzailearen erosotasunerako. Kivnon-eko kargatzaileak lurretan ezartzen dira, baina modulu hau erosten ez bada, bateriak AMR-tik kanpo atera behar dira. 12 V-ko eta 90 Ah-ko bateria baten masa 20 Kg inguru dabil eta industriako ekipamenduek 24 V-ko elikatze iturria behar dutenez, 40 Kg-ko masa manipulatu behar da, baterien kargak egin behar den bakoitzean. AMR-ak produkzio mailan ergonomiako egoerak hobetu nahi baldin badira, baterien aldaketak manipulatu behar den masaren ondorioz, prozedura honetan ergonomiako arazoak sortuko lirateke. Horregatik, barne kargatzaile bat integratuta izan beharko luke unibertsitateko prototipoak, automobil elektrikoaren moduan.

Gurpilak direla eta, eztabaida bat sortu zen. Industria barnean, motorren erakitzeko diseinuan berrikuntza gutxi egin daitekeelako. Baina automobilgintza elektrikoak gauzak era ezberdin batean planteatzen hasi dira. Azkeneko ikerketetan motorrak gurpiletan txertatu nahi

dira, edozein patinete elektriko modura. Motor hauek bizkorragoak eta kontsumo txikikoak dira. Gainera, motorra gurpilean integratuta dagoenez, beste osagai elektroniko eta elektriko ezberdinentzako espazio gehiago uzten dute. Horrek, motorren aukeratze prozesuan iritzi desberdinak suertatzen ditu. Azkenean motor klasikoak alde batera utzi dira, gurpil motordunen alde jokatu.

## 1.2 Prototipoaren diseinua

Prototiporen definizioa aurkezterako orduan, zenbait elementu minimo ezarri dira, baina hauek zeintzuk diren ez dira aipatu. Naiz eta hasiera batean AMR-ak elementu industrialekin osatzeko asmoa izan, prototipoa aurrekontu batera mugatu behar da. Industriako ekipamenduak 24 ordu, 7 egun eta 365 egun etengabeko lanean egoteko prestatuta daude, haren salmenta prezioa igoz. AMR barnean dauden osagai industrial guztietatik IPC-ak oinarritzko dela zehaztu da, azken finean bertan intelgentzia guztiak garatuko direlako. Behin ordenagailu honetan garatu naia diren algoritmoak funtzionatzen baldin badute, beste sentsore, eragileak, etab. integrazio prozedura direla esan daiteke eta intelgentziak erabiltzen dituzten barne funtzioak ez dira aldatuko. Hau da, LiDAR industrial batek ematen duen informazioa eta LiDAR-normal batek ematen duena berdina izaten omen da.

Merkatu industrialean dauden IPC guztietatik, Beckhoff-eko ekipamendu ezberdinak erostea hautatu da. Batez be, ordenagailu hauek Windows sistema operatiboa izateaz gain, prozesadoreak denbora errealean lanean prestatuta daudelako. Horrek, EtherCat komunikazioaren bitartez edozein sarrera edo irteera industrial atxiki ahal zaio ekipamendu honi. Hala ere, Windows barnean aplikazioak denbora errealean abiarazteko, liburutegi bereziak behar dira. Horregatik, ia intelgentzia guztia Beckhoff eskaintzen duen TwinCat software bitartez, denbora errealean programatu behar dira, Structured Text (ST) erabiliz.

IPC honek C6922-0030 ezagutzen da eta katalogo barnean ahaltsun gutxiko ekipamendua da (ikusi Irudia 43a). Prozesadorea Intel Atom E3815 bat da, non haren frekuentzia 1,64 GHz den. Nukleo bakarreko CPU bat da, 512 Kb Cache memoriarekin. Hautagai moduan, 8 Gb RAM memoria, 60 GB-eko disko gogorra eta 6 USB portu ezarri zaizkio. Berez, ikusitako ezaugarriengatik, ahalmen gutxiko ordenagailua da, baina erabiltzen duen Windows 10 sistema operatiboak, soilik behar dituen driver eta programak atxikitzen ditu. Bestalde, edozein programa diseinatzerako orduan, prozesadorearen % 80-ra TwinCat edo denbora errealeko programaziorako utzi behar zaio. Beraz, Windows-en aplikazioen bat programatzerako orduan, parametro hau kontuan hartu behar da. Batez be, ekipamendu honek nukleo bakarreko CPU-a delako.



(a)



(b)

Irudia 43 Unibertsitatean erabilitako elementu industrialak: a) IPC C6925-0030 b) EtherCat EK1100 (Beckhoff - etik lortutako irudiak {8})

Ordenagailu industrial honek sarrerek eta irteerak kontrolatzeko beste txartel bat behar du eta EtherCat EK1100 bezala ezagutzen da (ikusirik irudia 43b). Modulu honek, bai sarrera eta irteera digitalak edota analogikoak kontrolatzeko baliagarria da, baina beste komunikazio busekin ere. Adibidez: CanOpen-ekin, Profibus, etab. Azken finean, fabrikatzaile honek, industria barnean dauden beste komunikazio protokoloen harremana eskaintzen du.

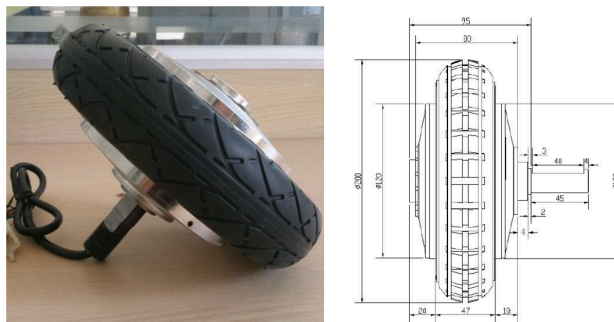
AMR-ren periferiarentzako, EK1100 burutik beste zenbait elementu eskegi behar dira. Horien artean: lizentzia gordetzen duen modulua, sarrera/irteera analogikoak, sarrera/irteera digitalak eta zabalera modulatuak (PWM) irteera. (begiratu Taula 5)

Taula 5. EK1100 moduluak eskegita dituen elementuak

Elementuak	Deskribapena	Sarrera/Irteera kopurua
EL6070-0033	Twicat lizentzia ezberdinak dituzte eta egindako programazioaren arabera, lizentzia ezberdinak erabili behar dira. Modulu honetan lizentziak gordeko dira.	-
EL3104	Sarrera analogikok eskaintzen duen modulua. Sarrera hauek -10 V eta 10 V artekoak izan behar dira	4
EL4134	Irteera analogikok eskaintzen duen modulua. Irteera hauek -10 V eta 10 V artekoak izan behar dira	4
EL1008	Sarrera digitalak ematen dituen modulua. 0 – 5 V arteko sarrerek, Low balioa emango du. 11 – 30 V arteko sarrerek High balioa itzuliko du.	8
EL2008	Irteera digitalak ematen dituen modulua. 24 V eta 0,5 A irteerak eskaintzen ditu	8
EL2502	Pultsu zabalera modulatuak (PWM) irteera. 0 – 24 V bitarteko pultsuen zabalera aldatu daiteke. Kontrola egiteko seinaleak izantzen dira eta frekuentzia 20 – 200 KHz alda daiteke.	2

Behin ekipamendu industrial hau definituta, beste elementu ezberdinen deskribapena egin behar da. AMR-ak mugiaraziko, ezaugarrietan gurrpil motor txertatuarekin erabiliko direla adostu da. Baina egitura mekanikoaren arabera, zenbat gurrpil motordunak behar diren ezarri behar da. Merkatuak ebazpen ezberdinak eskaintzen ditu eta ebazpen sinpleena triziklo baten egitura erabiltzea da. Beraz plataforma mugikor honek, 2 gurrpil elektriko erabiliko ditu eta hirugarren gurrpil horrek euskarri modukoa izango da.

Gurrpilak aukeratzeko orduan eta lehengo prototipoa denez, elikatze iturriak baldintza bakarra ezartzen du. Kasu honetan gurrpilak 24 V tentsioa jasan behar dute eta UUMotro 200 mm-tako eta 180 W gurrpilak hautatu dira.



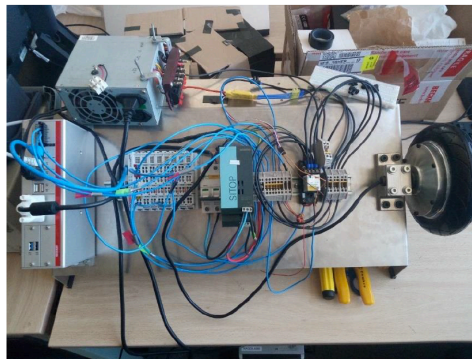
Irudia 44 Unibertsitateko AMR-ak erabiliko dituen gurrpil motordunak

Gurpilek erabiltzen duten motorrak, “Brushless Direct Current Motor” (BDCM) izaerakoak dira, hau da, eskuilak erabiltzen ez dituzten korrante zuzeneko makinak dira. Motor hauen estatorea izaera trifasikoa daukate, fase bakoitza polo batera konektatuta. Errotorea aldiz, iman permanentez osatuta dago. Motor hauen izaeragatik, faseen kontrola ezin bestekoa da. Horretarako, akoplamendu trifasikoa eta sei pausutako bihurgailu bat erabiltzen da. Makina hauek indar-pare altuak lortzeko ahaltasuna daukate eta beste DC motorrekin konparatuta mantentze prozedurarik ez daukate. Tresnari honek oso eraginkorra da, bai efizientzia aldetik, baita abiadura aldetik. Lortzen diren zehaztasunak, erabiltzen den kontrolagailuaren arabera izango da. Faseen kontrola egiteko, errotorearen posizionamendua ezin bestekoa da eta sensore gabeko teknikek gurpila abiarazteko zailtasunak sortzen ditu.

Behin motorra aukeratuta eta haren dimentsioak ezagututa, trizikloaren hirugarren gurpila aukeratu daiteke. Gurpilak 100 mm-ko erradioa dutenez, gurpil honen tamaina totala 100 mm baino txikiagoa ezin da izan, batez be egitura guztia orekatuta mantentzeko. Merkatuan dauden elementuen artean LKRA – TPA 101G – GS12 erabili da, non altuera 133 mm denez, gurpil motordunen egituran atxikitzeko behar den elementua diseinatzerako orduan kontuan hartu behar da.

Bateriak direla eta, bi konposizioak izan daitezke. Alde batetik, automobilgintzan konbustioko motorrak abiarazteko erabiltzen diren azido – berun bateriak daude. Bestetik, mugikorrek edo ibilgailu elektrikoetan erabiltzen diren LiFePO4 (litioa) bateriak aurkitu daitezke. Aurrekontuari estutzeko lehengo konposizio mota erabiliko da. Halako izaera dituzten bateria gehienak 12 V-ekoak dira. Beraz, osagaiak elikatzeko seriean konektaturiko 2 bateria beharko dira, 24 V-era tentsioa igotzeko. Fiamm FG23505 bateriak erabiliko dira eta hauek 35 Ah eskaintzen dituzte. Birkargatzeko, Victron Smart IP65 kargatzailea erabiltzen da eta AMR barnean ezarriko da. Horrela langileak soilik entxufe bat manipulatu behar du.

Elementu hauek guztiak izanda, egitura diseinatu baino lehen, diseinaturiko entsegu banku batean ezarri dira. Berez bi helburu zeuden: TwiCant softwarearen testa egitea eta gurpilen motorren kontroladoreak frogatzea. Bigarren kasuan prototipoaren garapenerako ezinbestekoa da.



Irudia 45 Entsegu bankua

Lehengo helburua, langilearen esku utzi da, baina bigarrenak, AMR-ren garapenerako ezin bestekoa denez, zenbait testetan parte hartzea ezinbestekoa zen. Hasiera batean 24 V-eko ESC bat erabili zen. ESC hauek motorraren abiadura aldaketa laguntzen duten dispositibo elektronikoak dira, non PWM izaerazko seinale bati ezker abiadura hori aldatzen den.



(a)

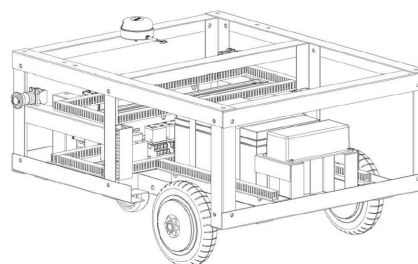
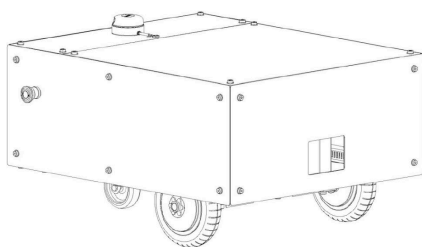


(b)

Irudia 46 BDCM motorren kontrolen aukeraketa: a) ESC kontrol modulua b) Patinete elektriko batek erabiltzen duen kontrola

Irudia 75 agertzen diren dispositiboen arteko diferentzia, gurpilaren posizioa kontrola egin hala es hautatu behar da. Eskuineko sisteman posizio sentsorerik ez dauka eta horrek gurpila abiarazteko arazoak sortzen zituen. Hazken finean sortu behar den konmutazioa desfasatua egiten delako eta errotoreak sortzen duen eremu magnetikoa, estatoreak ez du jarraitzen. Horregatik eskuin aldeko bezalako sistema bat erabili behar dela ondorioztatu zen. Bertan Hall bitartez gurpilaren abiadura eta norabidea ebaluatu daiteke. Neurgailu honek, eremu magnetikoen edo korronteen neurketa eginez, posizioa antzeman dezake. Hall bidezko zunda erabiliz, gune magnetikoa hurbiltzerakoan tentsio proportzional bat sortarazten du.

Behin gurpilaren mugimendua lortu ostean, beste produktuen erosketekin jarraitu da. Erosi behar den azkeneko sentsorea RPLiDAR A2M8 da. Berez, LiDAR honek izaera ez industrialia dauka, gainera segurtasunarentzako funtziorik ez dauka. Baina, garapen proiektuetan behar den material guztia hasiera batean ezin da eduki. Dena den, LiDAR-ra lan eremuko neurketak hartzeko erabili nahi da eta horretarako A2M8 LiDAR-ra egokia da. 8 metrotako luzera duen Laser Image Detection and Rangin (LiDAR) da, non motor bati esker sentsorea 360º bira dezake, gune guztiak aztertuz. Berez, halako dispositiboaren funtzionamendua laser argi baten islapen denboran oinarrituta dago. Informazioa serie bitartez komunikatzen da, 115200 bps-etan. LiDAR-rak duen motorraren abiadura aldatzeko, PWM motatako sarrera bat erabiltzen du. Abiadura eraldatuz, eremutik lortzen den informazioaren zehaztasuna eralda daiteke.



Irudia 47 Unibertsitateko prototipoaren diseinua

Osagaiak definituta izanda, AMR-ren diseinuarekin eta fabrikazioarekin hasten da, non Argolabe Ingenieria S.L. laguntzarekin burutzen den (ikusi Irudia 47). Behin prototipoa eskolan egonda, haren inteligentziaren garapenarekin has daiteke.



### 1.3 Inteligentziak garatzen

Prototiporen diseinua eta fabrikazioa amaitu ostean lortutako ekipamenduaren ahalmenak ezagutu behar dira eta horretarako inteligentzia ezberdinak garatu behar dira. Azken finean plataforma mugikor hau, era autonomoan mugiarazteko sentsoreak informazioa jaso behar dute, geroago garatutako algoritmoak prozesatzeko eta motorrei abiadura kontsignak bidaltzeko

#### 1.3.1 LiDAR-raren integrazioa

Edozein mugimendua egin aurretik, eremutik informazioa jaso behar da. Azaldu den moduan, LiDAR-ra espazioaren informazioa ematen duen dispositiboa da eta horretarako haren funtzionamenduz ikasi behar da. Berez, A2M8 programa batzuk eskaintzen ditu, baina inteligentziak garatzeko eta tartekatzen den informazioa zehatz mehatz jakiteko, tesian zehar zenbait programa burutu dira. Sentsoreak eskaintzen dituen 64-biteko liburutegiak erabiliz, C++ hiru programa prestatu dira, ondoren mex bitartez MatLab-erako konpilatzeke.

Lehengo programa LiDAR-raren driver-ararekin komunikazioa sortzen du, behin programa abiarazita, MatLab-en driver-entzako informazioa erakusle batean gordeko da (begiratu Kodea 3, or.- 138 -). Ordenagailuaren arabera COM portuaren zenbakia alda daiteke, horregatik Kodea 3-n hogeita zortzigarren lerroan "COM3" agertzen den tokian, ordenagailuaren baldintzatuta dagoen parametroa da. Dena den, Kodea 3 programaturiko informazioa *Pseudokodea 1* biltzen da.

*Pseudokodea 1* MatLab eta LiDAR arteko komunikazioa sortzeko

```
1 mexFuntion()
2 Sarrerak:
3     nlhs Espero den irteera kopurua
4     nrhs Espero den sarrera kopurua
5     prhs Sarrera bektorea
6
7 Irteera:
8     plhs Irteera bektorea
9
10
11 Prozedura:
12     Komunikazioa sortu COM portuarekin eta ezarri komunikazio abiadura
13     If (komunikazioan arazoak daude):
14         Errorea adierazi
15     Else ():
16         LiDAR-ren motorra abiarazi eta lehengo neurketa egin
17         COM komunikazioan sorturiko informazioa plhs bitartez kanporatu beste prozesu batzuetan
18     erabiltzeko
19     End if
```

Bigarren programak lehengo programa sortzen duen driver-araren informazioa hartu eta LiDAR - ra abiarazten du (begiratu Kodea 4 or.- 138 -). Abiarazterako orduan, MatLab-en 360 graduko buelta bakoitzeko, informazioa hiru bektoreetan gordetzen du. Programa honek, a *Pseudokodea 2*-ren bitartez aurre ikusi daiteke.

*Pseudokodea 2* MatLab eta LiDAR artean informazioa tartekatzeke

```
1 mexFuntion()
2 Sarrerak:
3     nlhs Espero den irteera kopurua
4     nrhs Espero den sarrera kopurua
5     prhs Sarrera bektorea
6
7 Irteera:
8     plhs Irteera bektorea
9
10 Prozedura:
11     LiDAR-rari adierazi Neurketa bat egin behar dela. (Pseudokea 1-n sorturiko komunikazio
12     informazioa erabili behar da).
13     LiDAR-rak bildutako informazioa bildu (Distantzia, Posizioa eta kalitatea)
14     Bildutako informazioa plhs bektorean gorde
```

Bereziki *Pseudokodea 2*-a abiaraztean, islapen puntu bakoitzaren distantzia, LiDAR-raren angeluaren posizioa, eta kalitatea itzultzen ditu. GrabScanData funtzioa erabiltzea gomendagarria da. Berez, LiDAR-raren datuak Buffer batean gordetzen dituelako eta beste edozein datu eskuratze funtzioa erabiliz, Buffer hori dagoen moduan informazioa bidaltzen du. Aldiz, proposatzen den funtzioarekin, informazioa buelta bat eman duenean tartekatzen da. Horrela, Matlab-en gordetzen dituen datuak, azkeneko biraketaren datuak direla ziurtatzen du. Informazio bilketa amaitzeko beste C++ programa baten beharra dago. Komunikazioa eteten eta LiDAR-ra gelditzen duen funtzioa da, *Pseudokodea 3* agertzen den moduan (Kodea 5 or.- 139 -).

*Pseudokodea 3* MatLab eta LiDAR arteko komunikazioa eteteko

---

```

1  mexFuntion()
2  Sarrerak:
3      nlhs Espero den irteera kopurua
4      nrhs Espero den sarrera kopurua
5      prhs Sarrera bektorea
6
7  Irteera:
8      plhs Irteera bektorea
9
10 Prozedura:
11  LiDAR-raren mugimendua eten. (Pseudokodea 1 emandako komunikazio informazioa erabiliz)
12  Komunikazio portua eten.
```

---

Behin 3 programa hauek C++ bitartez burututa izanda, mex komando bitartez MatLab-erako programa konpilatzen da. LiDAR-raren libreak erabiltzen direnez, honako hau exekutatu behar da: mex <Programaren izena> RPLidar.lib. Konpiladoreak zein liburutegi atxiki behar duen adosten zaio. Behin MatLab-eko funtzioak burutuak izanda, mapa bat burutzeko programa bat garatu behar da. Lehendabizi, egindako programak LiDAR-rak duen desplazamendua ez du kontuan hartzen. Berez datu bilketa nola egiten den antzemateko erabili daiteke( ikusi Kodea 6, or. - 139 -).

*Pseudokodea 4* LiDAR-reko informazioa MatLab-en gorde

---

```

1  Sortu LiDAR-raren komunikazioa (Pseudokodea 1 erabiliz)
2
3  While (Stop botoia aktibatu gabe):
4
5      LiDAR-rekin neurketak egin (Pseudokodea 2 erabiliz), distantzia, angeluak eta kalitate
6      aldagaietan gordez
7      Distantzien balioak milimetrotik metrotara pasa
8      Angeluen balioak gradutik radianetara pasa
9      Distantzia eta angeluak koordinatu polarretara pasa X eta Y aldagaietan informazioa bilduz
10     Lorturiko emaitzak erakutsi eta datuak bildu egitura batean geroago informazioa gordetzeko
11  End While
12
13  Gorde MatLab-eko informazioa
14  LiDAR-aren komunikazioa itxi (Pseudokode 3)
```

---

*Pseudokodea 4* programa stop botoi baten bidez geldiarazi daiteke. Horretarako bederatzigarren lerroan botoiaren konfigurazioa egiten da. Ordenagailuaren arabera, botoiaren errepresentazioa ikusteko, geldiune bat jarri behar da. Horrela MatLab-eri behartzen diogu, gune horretan gelditzera eta egindako prozedura guztiak amaitzea. Behin leihoa irekita, LiDAR-rarekin komunikazioa bidea sortzen da (ikusi Kodea 6 or. - 139 -, 9-garren eta 14-garren lerroa). Behin komunikazioa sortuta, datu bilketa hasi daiteke, Kodea 6-n hogeita laugarren lerroan agertzen den moduan. Bilketa honek 0,5 segundora emango da, ezarritako geldiunearen ondorioz. Programa amaitzeko, botoia sakatzen da eta RPLDIAR\_STOP funtzioa abiarazten du. Horrek LiDAR-raren driver eta ordenagailu arteko informazio tartekatzea eteten du. Gainera, egitura batean gordetako datuak, disko gogorrean grabatuko dira.

Mapa eraikitzeko, Kodea 6 baliatuz, programaturiko egiturari robotaren posizioa atxiki behar zaio, irakurketa bakoitzeko ( Kodea 7 or.- 140 -, 23-garren lerroan agertzen den moduan). Posizio hauek lurrean finkatutako gune konkrituak dira, momentu honetan AMR-ren gurgil kontrola burututa ez dagoelako. Gordetzen diren datuen kantitatearen ondorioz (2 Gb baino gehiago kasu batzuetan), Kodea 7-n berrogeita zortzigarren lerroan erabiltzen den “save”

funtzioa ezarri ber da, bertsio berezi bat behar delako tamaina handiko datuak direlako eta “-v 7.3” adierazten zaio.

*Pseudokodea 5* Mapa sortzeko MatLab-eko programa

```

1 Sortu LiDAR-raren komunikazioa (Pseudokodea 1 erabiliz)
2
3 While (Stop botoia aktibatu gabe):
4
5     LiDAR-raren koka puntua edo lokalizazioa sartu
6     LiDAR-rekin neurketak egin (Pseudokodea 2 erabiliz), distantzia, angeluak eta kalitate
7     aldagaietan gordez
8     Distantzien balioak milimetrotik metrotara pasa
9     Angeluen balioak gradutik radianetara pasa
10    Distantzia eta angeluak koordenatu polarretara pasa X eta Y aldagaietan informazioa bilduz
11    Lorturiko emaitzak erakutsi eta datuak bildu egitura batean geroago informazioa gordetzeko
12 End While
13
14 Gorde MatLab-eko informazioa
15 LiDAR-aren komunikazioa itxi (Pseudokode 3)

```

Funtzio hau abiarazi ondoren, mapa eraiki behar da, LiDAR-rak hartutako oztopen posizio erlatiboak, posizio absolutura bilakatuz. Kodea 7-n hogeita hemezortzigarren lerroan agertzen den posrobot datuak erabiliz, Z ardatzaren biraketa matrizea eta translazio funtzioak aplikatu behar zaie. Funtzio hauekin posizio eraldaketa egiteko ezin bestekoa dira, ekuazio ( 19 )-an aurkezten duen moduan.

$$\text{Mapa} = \begin{bmatrix} X_{\text{mapa}} \\ Y_{\text{mapa}} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} X_{\text{LiDAR}} \\ Y_{\text{LiDAR}} \end{bmatrix} + \begin{bmatrix} X_{\text{AMR}} \\ Y_{\text{AMR}} \end{bmatrix} \quad (19)$$

Mapa bilakatzen duen proposaturiko programa *Pseudokodea 6* da (Kodea 8 or.- 141 -). Kodea 7-n gordetako datuak kargatu egiten ditu lehendabizi. Ondoren, datuak posizio erlatibotik absolutura bilakatzen ditu. Behi datu absolutuak izanda, maparen errepresentazioa egiten du. Datu garbiketa ez denez egiten, puntu batzuk zapaltzen egiten dira. Beraz, etorkizunerako, mapan zapaltzen diren datu berdinak agertzen baldin badira, egoera hau ekidin duen funtzioaren bat planteatu beharko da. Horrek, maparen tamaina murriztu lezake eta edozein lokalizazio algoritmo abiarazterako orduan, era eraginkorrago batean egingo luke.

*Pseudokodea 6* Mapa eraikitzeko MatLab-eko programa

```

1 Kargatu PseudoKodea 5 sorturiko datuak
2 Laginketa_Kopurua = Jakin zenbat laginketa dauden gordetako informazioan
3
4 For (i=1-etik Laginketa_Kopurua-ra):
5     Posizio_Datuak = Laginketa bakoitzeko zenbat posizio dauden
6     Lidar_Datuak = Posizio bakoitzeko zenbat LiDAR datu dauden
7
8     For (j=1-etik Posizio_Datuak-era): // AGV-ren posizioa koordenatu absolutuetan bilakatzeko
9         theta = AgvThetaPose(j) + AgvThetaPose(j-1)
10        XAMR(j) = AgvXpose(j) * cos(theta) - AgvYpose(j) * sin(theta) + XAMR (j-1)
11        YAMR(j) = AgvXpose(j) * sin(theta) + AgvYpose(j) * cos(theta) + YAMR (j-1)
12
13        For(w=1-etik Lidar_Datuak-era) // LiDAR Puntuak koordenatu absolutuetara bilakatzeko
14            Xmapa(w) = XLiDAR(w) * cos(theta) - YLiDAR(w) * sin(theta) + XAMR(j)
15            Ymapa(w) = XLiDAR(w) * sin(theta) + YLiDAR(w) * cos(theta) + YAMR(j)
16
17        End For
18    End For
19 End For
20
21 Gorde MatLab-eko informazioa

```

Kodea 8 garatutako programak LiDAR-rak bildutako informazio guztia irudikatzen du. Zenbait kasutan, irudikatu behar ez diren elementuak mapan ager daitezke, Adibidez: bide erdian dagoen pertsona bat, etc. Halako egoeretan gordetako mapa irudi batera pasatzen duen beste garapena bat osatu nahi da. Horrekin bat, simulaziozko LiDAR funtzioa garatu nahi da, *Pseudokodea 7* aurkezten duen moduan (ikusi Kodea 9 or. - 142 -). Lokalizazio inteligentziak garatzerako orduan, LiDAR-ra simulazio hau erabili daiteke, ordenagailuan dauden mapak erabiliz. Funtzio honek, maparen informazioa eta AMR edo AGV-ren posizioa pasata, LiDAR-rak ikusi ahal dituen puntuak itzultzen ditu. Berez, funtzio honek A2M8-ren baldintzetara egokitu

behar da, non gehienez 8 metrotako luzera duen. Horrekin bat,  $-\pi$  eta  $\pi$  harteko baldintza ezarri da, batez be mapa eraikitzerako orduan, 360 graduak ez direlako erabiltzen, ordenagailuak LiDAR - raren atzeko partean dagoelako. Orduan, garapena egiten den bitartean, datuen bilketa angelu baldintza horietara egokitu da.

#### Pseudokodea 7 LiDAR funtzioa

```

1 LidarMap()
2   Sarrerak:
3   Mapa eta AMR-ak duen lokalizazioa
4
5   Irteerak:
6   AMR-aren posizioaren arabera LiDAR-ak ikusten dituen puntuak
7
8   LiDAR-rak ikusten dituen puntuak lortzeko prozedura:
9   For (i=1-etik Mapa dituen puntuak arte):
10  //LiDAR-raren ezaugarri fisikoak sortzen dituen mugapenak arabera puntu ezberdinak ikus dezake
11  If (Xmapa(i) <= XAMR + MaxLidar eta Xmapa(i) > XAMR - MinLidar):
12  If (Ymapa(i) <= YAMR + MaxLidar eta Ymapa(i) > YAMR - MinLidar):
13  PointsX(cont) = (Xmapa(i)-XAMR) * cos(theta) - (Ymapa(i)-YAMR) * sin(theta)
14  PointsY(cont) = (Xmapa(i)-XAMR) * sin(theta) + (Ymapa(i)-YAMR) * cos(theta)
15  cont = cont + 1
16  End If
17  End If
18  End For
19
20  LiDAR-rak objektuen atzetik ikusten ez dituen puntuak kentzeko prozedura:
21  If (Puntuak aurkitu baldin badituz):
22  PointsX eta PointsY koordenatu polarretara pasa alpha eta R parametroak lortuz
23  For (h=1-etik pi-ra):
24  Bilatu alpha eta h balio berdinak duten datu guztiak
25  If (alpha balio berdinak badaude):
26  R balio minimoa bilatu
27  R_minimoa eta alpha balioak koordenatu XLiDAR YLiDAR Kartesiarretara pasa
28  End If
29  End For
30  End If

```

*Pseudokodea 8-ak* (Kodea 10 or.- 143 -) planteatzen duen funtzioak, MatLab-en lortutako mapa .mat formatua argazkietako formatura egokitzen du. Aipatu den moduan, kasu batzuetan zenbait mapa gune zuzendu behar dira eta irudi batean datuak baldin badaude, hauek manipulatzeko edozein irudien software erabili daiteke. Dena den, kode honek *Pseudokodea 7* - an egindako programazioa frogatzeko ere erabili da. Azken finean, mapa osoa irudian bilakatu beharrean, maparen laginketa bat egiten delako.

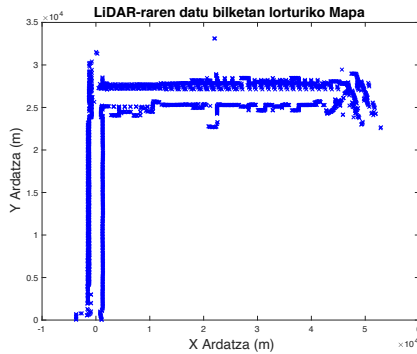
#### Pseudokodea 8 Mapa irudi batean bilakatu

```

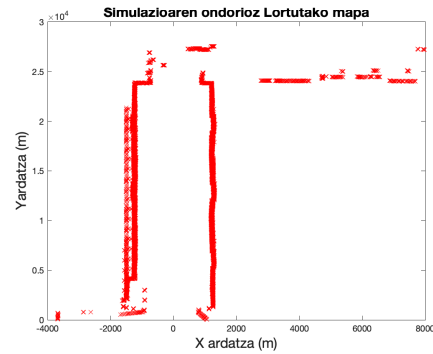
1 Pseudokodea 6-k lorturiko mapa kargatu
2 Definitu AMR-ren hasierako posizioa, daraman abiadura eta noranzkoa.
3 For (v=1-etik Laginketa kopuru arte):
4   Pseudokodea 7 abiarazi, LiDAR puntuak lortzeko
5   For (g=1-etik Aurkitutako LiDAR puntuak-arte) // Parametrotik at dauden puntuak
6   If (XLiDAR > 0 eta YLiDAR > 0)
7     Xargazki(cont) = XLiDAR
8     Yargazki(cont) = YLiDAR
9     cont = cont + 1
10  End If
11  End For
12  End For
13
14 // Estimatu irudiaren tamaina kontuan izanda metro bateko espazio emango zaiola
15 Xmin = min(Xargazki) + 1000
16 Xmax = max(Xargazki) + 1000
17 Ymin = min(Yargazki) + 1000
18 Ymax = max(Yargazki) + 1000
19 Xirudia = (Xmax - Xmin)/10 // mm-etik cm-tara pasa
20 Yirudia = (Ymax - Ymin)/10 // mm-etik cm-tara pasa
21
22 //Sortu irudi beltza
23 Irudia = zero(Xirudia,Yirudia) //zeroz beteriko matrize bat, definituriko irudiaren tamaina erabiliz
24
25 //Irudi beltzean LiDAR-ak ikusitako objektuak zuriz identifikatuz
26 For(g=1-etik Xargazki kopurua bete arte):
27   Xims = Xargazkia(g) - Xmin
28   Yims = Yargazkia(g) - Ymin
29   Irudia (Xims,Yims) = 255
30 End For
31 Gorde Xmin eta Ymin
32 Gorde Irudia

```

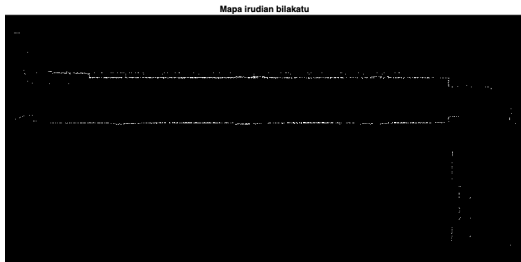
Kodea 10-k aurkezten den programazioa bi ebazpenetik lortu da. Alde batetik, Kodea 9 -n garatutako funtzioaren frogapena egiten da eta bestetik maparen irudia sortzen da. Zazpigarren lerrotik hogeita hamazazpi lerroa, LiDAR-ra Kodea 8-n lorturiko mapatik laginketa bat egiten du. Horretarako, hamargarren lerrotik hamabosgarren lerroa, mugimendu baldintzak ezartzen zaizkio simulatutako AMR-ri. Kasu partikular honentzako, plataforma mugikorra Y ardatzean zehar soilik mugiaraziko da. Berrogei lerrotik, hirurogeita laugarren lerroa, mapa irudia bilakatzeko erabiltzen den programazioa aurkitu daiteke. Berez, irudia bilakatzerako orduan, fitxategi batean  $X = 0$  m,  $Y = 0$  m posizioa pixeletan non dagoen gordetzen da. Horrela irudia irekitzerako orduan, fitxategi horrekin AMR-ren hasierako posizioa zehaztu daiteke.



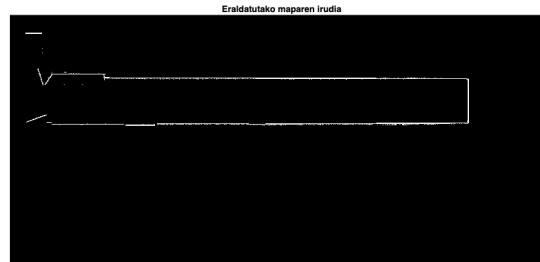
(a)



(b)



(c)



(d)

Irudia 48 Eskolako korridorearen maparen errepresentazioa: a) Kodea 8-k sortzen duten mapa, b) Kodea 9 funtzioak emandako mapa, c) Kodea 10 mapa irudian bilakatzearen emaitza, d) Lortutako irudiaren maparen eraldaketa

Irudia 48d-an mapa moldatutako irudi bat erabili daiteke, bertatik informazioa ateratzeko. Horretarako, Kodea 9-an erabiltzen den programazioaz baliatuz, irudiekin funtzionatzen duen LiDAR simulagailu bat planteatzen da, *Pseudokodea 9* aurkezten duen moduan (begiratu Kodea 11 or.- 143 -). Kontuan izanda LiDAR-rak paretan edo bestelako objektuen atzetik dauden elementuak ezin dituela ikuzi, beraz if baten bidez halako kasuak baztertzen dira. Gainera, funtzio honek edozein LiDAR-ren parametroak sar daitezke sarrera moduan simulagailuan. Orduan funtzio honen sarrerak honako hauek dira: Maparen irudia, AMR-ak mapa absolutuan duen posizioa, LiDAR-rak neurtu dezakeen distantzia maximoa, LiDAR-rak neurtu dezakeen distantzia minimoa, sentsore honen graduak nondik nora doazen eta MapData fitxategia. MapData mapatik irudira bilakatzeko zenbait informazio gordetzen du eta parametro hauen arabera px-ak eta maparen neurriak zehazten ditu.

### Pseudokodea 9 Iruditik maparen informazioa lortu.

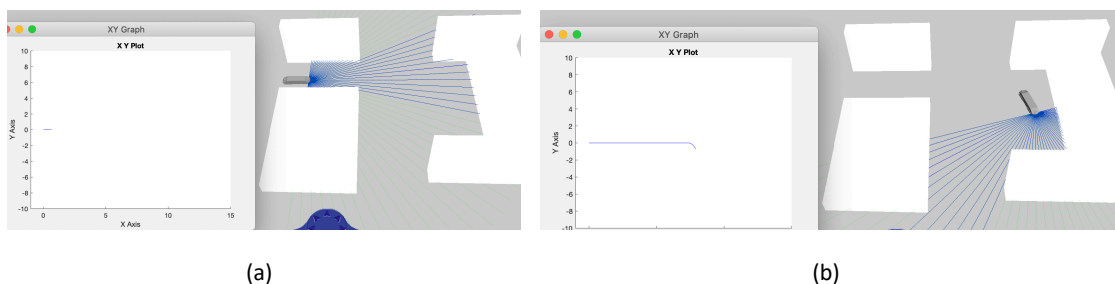
```

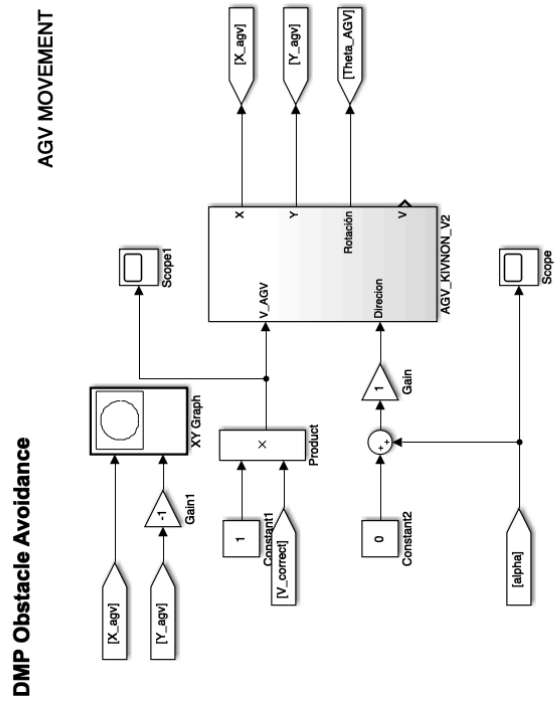
1 ImageLidar()
2   Sarrerak:
3     Maparen Irudia eta MapData, AMR posizioa, LiDAR-raren neur dezakeen distantzia maximoa eta
    minimoa, baita LiDAR-rran angelu mugak.
4
5   Irteerak:
6     LiDAR-rak ikus dezakeen puntuak bektore moduan aterako dira
7
8   For(ImgX=1-etik Irudiak duen X tamaina arte):
9     For(ImgY = 1-etik Irudien duen Y tamaina arte):
10      If (Irudia(ImgX,ImgY) ezberdin 0):
11        Xpoint = ImgX * 10 + MapaData(1) // MapaData Pseudokodea 8-tik datorren informazioa da
    eta bider hamar egiten da, mm-tara pasatzeko.
12        Ypoint = ImgY * 10 + MapaData(2)
13
14        XLiDAR = Xpoint * cos(-θ) - Ypoint * sin(-θ) // θ -> AMR-ak duen rotazioa plano
    absolutuan
15        YLiDAR = Xpoint * sin(-θ) + Ypoint * cos(-θ)
16
17        If(XLiDAR <= AMRX + LidarDistMax eta XLiDAR >= AMRX + LidarDistMin):
18          If (YLiDAR <= AMRY + LidarThetaMax eta YLiDAR >= AMRY + LidarThetaMin):
19            XLiDAR_fin = XLiDAR * cos(θ) - YLiDAR * sin(θ)
20            YLiDAR_fin = XLiDAR * sin(θ) + YLiDAR * cos(θ)
21            Puntuak(cont,:) = [XLiDAR_fin,YLiDAR_fin]
22            cont = cont + 1
23          End If
24        End If
25      End For
26    End For
27  End For
28
29 // Lorturiko emaitzak iragazi egin behar dira, LiDAR-rak objektu atzean dagoen informazioa
    ezin duelako ikusi
30
31 Puntuak koordenatu kartesiarretik koordenatu polarretara igaro behar dira R eta Theta
    lorrarazteko
32 For(t=1-etik R kopurua bete arte):
33   Bilatu Theta(t) beste Theta balioekin berdintasuna dagoenik
34   If (Theta berberak daudenean):
35     R_min = min(R(Aurkitutako Balioak)) // R Balio minimoarekin geratu
36   Else
37     R_min = R(t)
38   End If
39
40   Zx(t) et Zy(t) lortuko dira, R_min balioa eta Theta balioa koordenatu kartesiarretara pasaz
41 End For

```

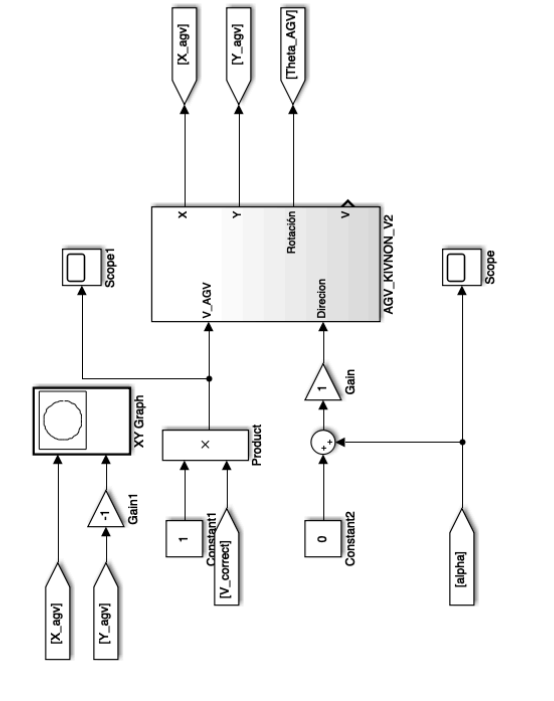
#### 1.3.2 Roomba Mode

Atal honetan, unibertsitateko lehenengo prototipoa abiarazteko eta edozein eremutara mugiarazteko, garatutako programazioa aurkeztuko da. Robot aspiragailuetan oinarritutako mugimendua da, non talka baten ondorioz robotak haren norabidea aldatzen duen. Prototipoak talkak neurtzeko sensorerik ez duenez, LiDAR-rak hartzen dituen datuez baliatuz, objektuak zein distantziatara dauden jakin daiteke. Oztopo hauek maila batera hurbiltzen direnean, AMR-aren noranzkoa aldatuko da. Laborategiko teknologia suspertzeko simulagailu bat garatu da, beraz proposaturiko programa AMR-an frogatu baino lehen, simulagailuaren bitartez, lehenengo frogapenak planteatu ziren (ikusi Irudia 49)

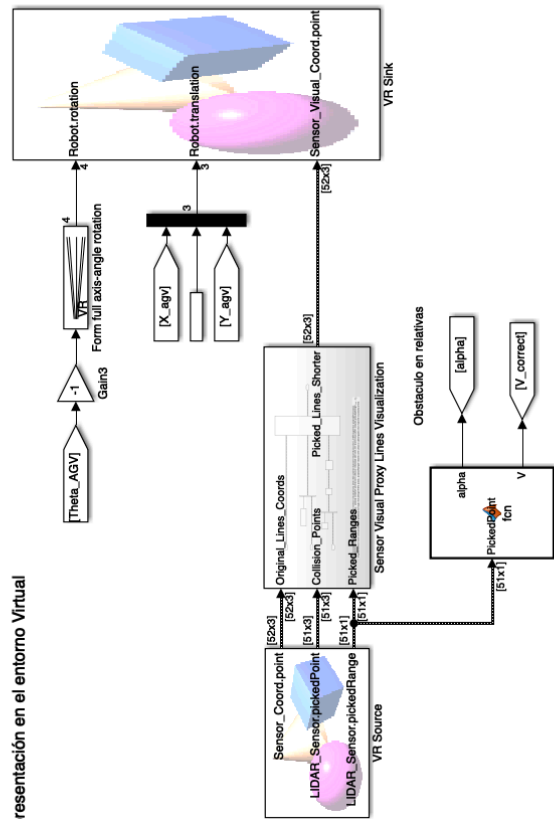




### AGV MOVEMENT



### AGV REPRESENTATION Representación en el entorno Virtual



(c)

Irudia 49 Roomba mode-ren simulazioa, a) hasierako aldiuneko egoera, b) objektuak saihesten duen egoera, c) erabilitako Simulink-eko programazioa

Simulagailu honetan planteatutako programazioa oso sinplea da, non objektu bat oso gertu dagoenean AMR-ak eskuinera biratzen duen. Berez, simulagailu honek buruan zegoen ideia funtzionatzen zuen balioztatzeko erabili da. Programazio hau ikusi beharrean, AMR-an programatutakoa erakutsiko da. Berez Roomba programazioa bi esteketan landu da. Lehenkoa, MatLab-ren bitartez AMR-ren kontrola egiten dena da eta bigarrena, LiDAR-ra eta Roomba mode AMR-ren IPC-an integratu dira, MatLab duen ordenagailu eramangarri baten menpe ez egoteko.

Lehenko programazioa MatLab bidez burutu da(ikusi Kodea 12 or. - 144 -), *Pseudokodea 10*-a jarraituz. Horretarako Beckhoff-eko IPC-a eta MatLab dagoen eramangarria komunikatu behar dira, ADS izeneko portu batetik. ADS portu honek, Beckhoff-ek eskaintzen duen komunikazio asinkronoa da. Portu honen funtzionamendua honelakoa da. Lehendabizi TwinCat garatu behar dituen operazio guztiak abiaraziko ditu. Behin prozesu horiek amaitu ostean, unitate ziklo batean denbora baldin badago, ADS portua irekiko da. Horrela TwinCat periferiako elementuekin harremanetan jartze da. Portu honek ematen dituen aukerak mugagabeak dira, azken finean edozein osagai komunikatzeko erabili daiteke. Kontuan izanda, sistema honek denbora errealetik at gelditzen dela, sistema determinista ez delako. Denbora errealean egin behar bada, orduan TwinCat barnean programatu beharko da. Komunikazio hau egiteko, Beckhoff-eko departamenduan C++ programa bat eskaini du, non ADS protokoloa programatuta dagoen. Programa hori mex bitartez konpilatuz, MatLab barnean abiarazteko aukera dago.

#### *Pseudokodea 10* MatLab bidezko Roomba Mode

---

```

1  Sortu LiDAR-raren Komunikazioa COM informazioa LiDAR aldagaian gorde
2
3  While(n=1)
4      Irakurri nReal balioa ADS portutik
5      LiDARTime(LiDAR)
6      If (cont = 1) // Lehenko aldiunean cont = 1 balio du
7          Galdetu erabiltzaileari zer egin nahi duen Start edo Exit, erantzuna Var gorde
8          cont = 0
9      End If
10
11     Switch Var:
12         Case "Start":
13             ADS-ren bitartez bidali -> bBool = True // Motor Kontrolagailuak gelditu
14             ADS-ren bitartez bidali -> nByte = 0 // Balioa bidali
15             ADS-ren bitartez bidali -> nByte = 255 // Balioa berriro bere izaerara itzuli
16             Var = "Move"
17             Speed = 0.2 // Abiadura ezarri
18             Steer = 0 // Noranzko jarri
19         End Case
20
21         Case "Move":
22             ADS-ren bitartez bidali -> nUInt = Speed
23             ADS-ren bitartez bidali -> nInt = Steer
24             ADS-ren bitartez bidali -> nByte = 1 // Balioa bidali
25             ADS-ren bitartez bidali -> nByte = 255 // Balioa berriro bere izaerara itzuli
26         End Case
27
28         Case "Stop":
29             ADS-ren bitartez bidali -> bBool = False
30             ADS-ren bitartez bidali -> nByte = 0 // Balioa bidali
31             ADS-ren bitartez bidali -> nByte = 255 // Balioa berriro bere izaerara itzuli
32         End Case
33
34         Case "Exit":
35             n = 0
36         End Case
37     End Switch
38 End While
39
40 Geldiarazi LiDAR-raren komunikazioa
41
42 /// Bestelako funtzioak
43 LiDARTime():
44     Sarrera:
45     LiDAR-raren komunikazioa egiterako orduan sorturiko datuak
46
47     Irteera:
48     LiDAR-rak ikusten dituen puntuen distantziak eta angeluak
49
50     Prozesua:
51     LiDAR-raren irakurtze funtzio abiarazi R-ak eta Thetak lortzeko
52     [Distantziak, angeluak]Security(R,Theta) // LiDAR-raren seguritate funtzioa abiarazi
53
54 SecurityLiDAR():
55     Sarrera:
56     LiDAR-rak emandako datuak, bai Radioak, baita angeluak

```

---



```

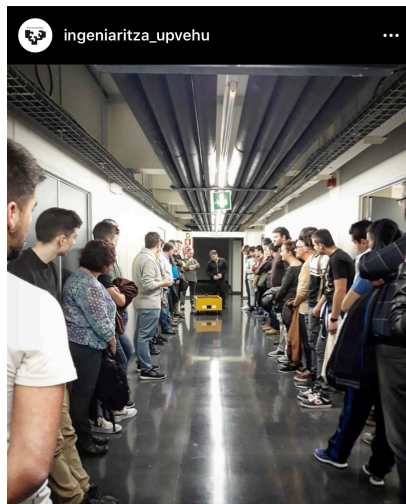
58
59 Irteera:
60 Mugimenduarentzako aldagai Var eta angelu bat
61
62 Prozedura:
63 For (h=1 Radio guztiak aztertu arte): // Baztertu zenbait datu okerrak
64   If(Radioa(h) > 0):
65     Erradio(cont) = Radio(h)
66     Ang(cont) = Angeluak(h)
67     cont = cont + 1
68   End If
69 End For
70
71 Bilatu erradio minimoa
72 If (Erradio minimoa <= 400):
73   Var = "Stop"
74 Else():
75   If(Erradio minimoa <= 700):
76     If(ang(Aurkitutako erradio minimoaren lekuan)<90):
77       Var = "Move"
78       Steer = 90
79     Else If(ang(Aurkitutako erradio minimoaren lekuan)>270):
80       Var = "Move"
81       Steer = 90
82     End If
83   Else
84     mugitu
85     biraketa gabe
86   End If
87 End If

```

---

SecurityLiDAR funtzioaren bitartez, LiDAR-retik jasotako informazio tratatzen da, non oztopo bat 400 mm baino gertuago baldin badago, zuzenean AMR-ak geldiarazten den. Horrek AMR-ak izango duen segurtasun funtzioaren lehengo urratsak dira. Oztopoa 700 mm-tara baldin badago, orduan AMR-aren noranzkoa eralda daiteke, bai eskuinera, baita ezkerretara. Horretarako, LiDAR-rak ematen dituen erradioaz gain, angeluak aztertu behar dira eta oztopoa aurkitutako angeluaren arabera, AMR norabidea eraldatzen du. Adibidez: oztopoa eskuineko aldean baldin badago, orduan eskuineko gurpilaren abiadura handituko da, ezkerretara biratzen hasteko.

Garatutako programazio eta ADS-ren bitartezko komunikazioari esker, AMR-ak 2019-ko eskolako ate irekietan parte hartu zuen. Bertan bisitatzailleak korridore bat osatzen zuten eta plataforma Roomba Mode-ren bitartez mugiarazten zen inor harrapatu gabe (ikusi Irudia 50).



Irudia 50 Ate ireki eguna Gasteizko Ingeniaritza eskolan egindako argazkia (ingeniaritza\_upvehu instagram-etik lortutako irudia)

Roomba Mode MatLab-ren bitartez funtzionatzen duela ikusita, bai LiDAR-ra, baita Roomba Mode TwinCat barnean programatu nahi dira. LiDAR-ra Beckhoff-ekin bateragarria ez denez, ADS-ren bitartez komunikazioa sortu da. Berez, C++-eko programa batek IPC barnean abiaraziz, TwinCat eta programa hori informazioa tartekatu dezakete. TwinCat-ek naiz eta IPC-an abiarazi, denbora errealeko sistema bat denez, aplikazioak abiarazteko denbora errealeko funtzioak behar ditu. Horregatik, Beckhoff-ekok ADS portu honekin IPC-an edo haren periferian dagoen edozein elementuen artean informazioa tartekatu daiteke. LiDAR-raren kasuan C++ eta ADS-ren bitartez egingo da, LiDAR-raren funtzioak RT-ekin bateragarriak ez direlako.

Horretarako Beckhoff C++ txantilo batzuk eskaintzen ditu eta hauek nahi den informazioaz betetzen dira. LiDAR-aren integrazioa soilik egin beharrean, teklatu bidez zenbait komando bidaltzeko programa bat burutu da ere, batez be AMR bultzatu gabe mugiarazteko. C++ programa honek Windows-en konektatzen diren periferikoen datu bilketak TwinCat barnean gordetzen ditu, gero denbora errealean abiarazten diren programatan erabiltzeko (ikusitako Kodea 13 or.- 145 -). Datu hauek GVL\_Lidar izeneko aldagai orokorren zerrendan gordetzen dira. Bertan bai LiDAR-ak ematen dituen luzerak, baita angeluak, arrDato eta arrPose - n aldagaian gordeko dira. Azken finean GVL\_Lidar izeneko zerrendan, LiDAR inguruko ezinbesteko informazioa gordetzen du eta arrDato edo arrPose, zerrenda horretako Scan bateko datuak gordeko dira. 600 datuzko bektoreak dira eta Kodea 13 abiarazten den bakoitzean, bektore horiek berriztuko dira. Teklatutik bidaltzen den informazioa, GVL\_DigitalIO zerrendaren nTeclado aldagaian gordetzen dira. Dena den, funtzionamendua era erraz batean ikustarazteko, *Pseudokodea 11*-an beharrezko diren funtzioak azaltzen dira.

#### *Pseudokodea 11* C++ LiDAR eta TwinCat komunikazioa

---

```

1  Ireki ADS portua
2  If (ADS portua errore gabe):
3  Sortu LiDAR Komunikazioa
4  If(LiDAR erroreak baditu):
5      Break
6  Else
7      Abiarazi LiDAR motorra
8      Itxaron 3 segundo LiDAR-rak hartu beharreko abiadura hartzeko
9      While():
10         If(Teklatuak daturen bat badauka):
11             Teklatua = ASCII Kodearen balioa gorde
12         End If
13         LiDAR irakurketa bat egin
14         LiDAR datuak hartu
15         ADS-ren bitartez, bai Teklatuaren datuak, baita LiDAR-rarenak bidali
16         If(ADS erroreirik baldin badu):
17             ADS portua itxi
18             Gelditu LiDAR-ra
19             Itxi LiDAR-raren komunikazioa
20             break
21         End If
22     End If
23 End If

```

---

Programa hau Windows hasi bezain prest abiarazteko, exe fitxategi bat sortzen da eta exe fitxategi honek Windows-ari eginkizun moduan ezartzen zaio. Bestalde, TwinCat programatzeko era desberdinak daude. Hizkuntz hauek industria barnean erabiltzen direnak dira, Adibidez: Kontaktore bidezko programaketa, Testu Egituratua, etab. Programatzailearen edota enpresak adostu dituen baldintzen arabera, hizkuntz ezberdinak erabiltzen dira. Beckhoff - en Roomba Mode funtzioa garatzeko testu egituratua (ST) erabili da (begiratu Kodea 14 or.- 147 -) eta *Pseudokodea 12*-an oinarrituta dago. Programa honekin AMR-ren mugikortasun autonomoa bermatzen da eta nabigazio autonomorentzako LiDAR-raren erabilpena ulertarazten da. Hemendik aurrera beste inteligentzien garapenekin jarraituko da, AMR-ak gero eta ahaltsuagoa bilakatuz.

*Pseudokodea 12 Roomba Mode funtzioa TwinCat bidez programatuta*

```
1 RombaModeST():
2   Sarrera:
3     LiDAR-rak emandako angeluak eta erradioak
4
5   Irteera:
6     Abiadura eta noranzkoa AMR-rentzako
7
8   Prozedura:
9     If(Seguratasun Distantzia < Erradio Minimoa):
10      Aurkitu Erradio Minimoa eman den angelua eta ThetaLidar aldagaian gorde
11      If(ThetaLidar > 360):
12        ThetaLidar = ThetaLidar - 360
13      End If
14      If(ThetaLidar >= 0 eta ThetaLidar <= 120): // Eskerretara biratu
15        AMR abiadura ezarri
16        AMR noranzkoa ezarri
17      Else If(ThetaLidar >= 250 eta ThetaLidar < 360): // Eskuinetara biratu
18        AMR abiadura ezarri
19        AMR norantza ezarri
20      Else // Zuzen joateko
21        AMR abiadura ezarri
22        AMR noranzkoa nulua ezarri
23      End If
24      Else // Zuzen joateko, naiz eta ezarritako tartean ez egon
25        AMR abiadura ezarri
26        AMR noranzkoa nulua ezarri
27      End If
```

### 1.3.3 Seguratasun funtzioa TwinCat barnean

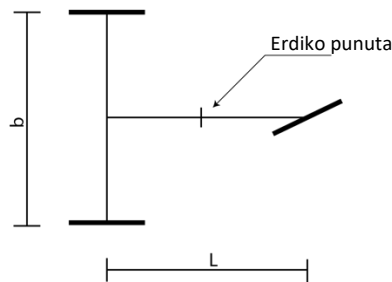
Unibertsitatean garatutako prototipoak seguratasun falta dauka, batez ere dituen osagaiengatik. Horrek seguratasun funtzio bat burutzea behartzen du. LiDAR-rak ematen dituen datuei ezker, oztopo bat seguratasun eremu barnean dagoen determinatu daiteke. Muga hori gainditzen baldin badu plataforma mugikorra geldiarazi behar da eta oztopoa desagertu arte AMR-ak ezin da abiarazi. Dena den, erabiltzaileak teklatua erabiliz, seguratasun baldintza kendu dezake eta erantzukizuna langilearen esku jarriko da. Seguratasun funtzio honek edozein programekin exekutatu da, seguratasuna bermatzeko denbora osoan (ikusitako Kodea 15 or. - 147 -). Roomba mode bezala, testu egituratua programatu da. Berez funtzioa oso sinplea da eta MatLab barnean erabiltzeko bezalakoa da. Non baldintza egituren bitartez, oztopoa seguratasun gunean sartu den hala ez kontrolatzen da.

*Pseudokodea 13 Seguratasun funtzioa TwinCat bidez programatuta*

```
1 SeguratasunaST():
2   Sarrera:
3     LiDAR-rak emandako angeluak eta erradioak
4
5   Irteera:
6     Abiadura AMR-rentzako
7
8   Prozedura:
9     For(q=1-etik LiDAR datu guztiak aztertu arte): // Bilatu balore minimoak
10      If(LiDAR-rak emandako erradioak(g) > 0):
11        If(LiDAR-rak emandako erradioak(g) < BaloreMinimoa):
12          BaloreMinimoa = LiDAR-rak emandako erradioa(g)
13          Aurkitutako Gunea = g
14        End If
15      End If
16    End For
17
18    If(BalorMinimoa < Seguratasun Tartea):
19      If(BalorMinimoa < 0): // LiDAR-ra txarto konektatuta baldin badago
20        Seguratasun modua aktibatuta
21      End If
22      If(Seguratasun Modua Aktibatuta baldin badago):
23        AMR-aren motorrak geldiarazi
24
25      Else If(Seguratasun Modua Aktibatuta baldin badago eta Teklatutik seguratasuna kodea
26      ezarrit):
27        AMR-ren motorrak abiarazi eta mugitzeko aukera ematen du
28      End If
29      Else
30        AMR-a abiarazita utzi
31      End If
```

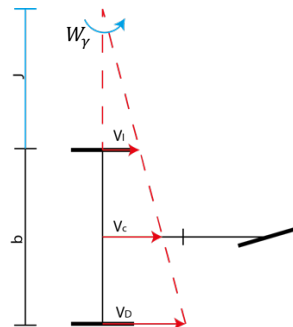
### 1.3.4 Unibertsitateko AMR-ren modelo matematikoa

Lokalizatorako edozein algoritmo abiarazteko, plataformaren egitura mekanikoaren ebazpen matematiko bat behar da, non abiadura eta noranzko kontsignak sarrera moduan ezarriko diren eta irteera motor bakoitzaren abiadura izango da. Unibertsitateak garatu duen plataformaren arabera, triziklo moduko egitura dauka (ikusi Irudia 51). Umeen trizikloetan abiadura eta noranzkoa aurreko gurpilean ematen da, baina kasu honetarako atzeko bi gurpilak abiadura ezarriko dute eta haien abiadura ezberdintasunen ondorioz biraketa sortzen da.



Irudia 51 Unibertsitateko AMR egituraren errepresentazioa

Edozein modelo matematikoa ezaugarri fisikoen arabera, parametro finko batzuk izango ditu. Hauek diseinu mekanikoa egiterako orduan ezartzen dira.  $L$  parametroa aurreko eta atzeko gurpilen ardatzen arteko luzera adierazten du eta  $b$  parametroak aldiz, atzeko ardatzaren zabalera. Modeloaren ekuazioak lortzeko, kasu hipotetikoa planteatzen da, non AMR-ak biraketa batean dagoela irudikatzen den (begiratu Irudia 52).



Irudia 52 Unibertsitateko AMR-ak biraketa egoeran

$V_D$  eskuineko motorraren abiadura,  $V_I$  ezkerreko motorraren abiadura eta  $V_C$  modeloaren sarreraren abiadura direla kontuan hartuz, biraketa puntuaren abiadura angeluarra kalkulatu daiteke (ikusi (20) eta (21) ekuazioak). Hauen konbinaketa eginez, biraketa abiadura  $j$  parametrotik islatu daiteke eta ezagun diren parametro arabera definitu, aztertu (22) ekuazioa.

$$W_y \cdot J = V_I \quad (20)$$

$$W_y \cdot (b + J) = V_D \quad (21)$$

$$W_y = \frac{V_D - V_I}{b} \quad (22)$$

$V_C$  edo Erdialdeko abiadurak biraketa abiadura angeluarrarekin erlazionatu daiteke (23) ekuazioan bezala, eta ekuazio hori (20) eta (22)-rekin lortzen da.

$$W_\gamma \cdot (b/2 + J) = V_c \quad (23)$$

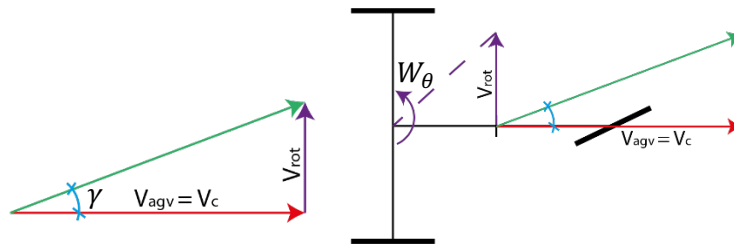
$$V_c = \frac{V_D + V_I}{2} \quad (24)$$

Erdialdeko abiadurak mapa absolutuaren bi ardatzetan deskonposatu daiteke. Hauek AMR mapan duen biraketarekin erlazionatuta dago, ikusi ( 25 ) eta ( 26 ).

$$V_x = V_c \cdot \cos(\theta) \quad (25)$$

$$V_y = V_c \cdot \sin(\theta) \quad (26)$$

$\theta$  angelua lortzeko, modeloan sartzen den abiadura eta norantzarekin lor daiteke. Horretarako, lehendabizi biraketazko abiadura deritzon parametro bat lortu behar da. Abiadurak AMR-an zehar deskonposatuz, sarrerako datuak eta biraketa abiadura erlazionatu daiteke (begiratu Irudia 53).



Irudia 53 Unibertsitateko AMR-ak biraketa abiaduraren kalkuluen errepresentazioa

Abiaduraren deskonposizioa garatzen duen irudiari esker, ( 27 )-tik ( 30 )-era biraketa abiadura horrek egitura mekaniko zehar nola deskonposatzen den irudikatzen dituen ekuazioak planteatu daitezke.  $W_\theta$  edo biraketa abiadura angeluarraren kokapena diseinuaren menpe dago eta kasu honetan trizikloaren diseinua jarraitzen duenez, bi motorren ardatz arteko erdigunean ezartzen da.

$$V_{rot} = V_c \cdot \tan(\gamma) \quad (27)$$

$$\frac{W_\theta \cdot L}{2} = V_{rot} \quad (28)$$

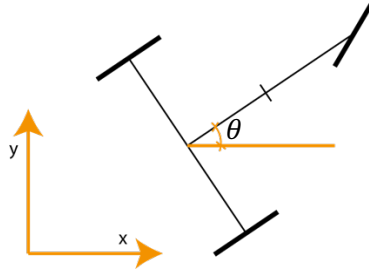
$$\frac{W_\theta \cdot L}{2} = V_c \cdot \tan(\gamma) \quad (29)$$

$$W_\theta = \frac{V_c \cdot 2 \cdot \tan(\gamma)}{L} \quad (30)$$

Unibertsitateko AMR-ak definitzen dituen ekuazioak ( 25 ),( 26 ) eta ( 30 ) dira. Hauek matrize batean bildu daitezke laburbiltzeko garapena, ikusi ( 31 ).

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \\ \frac{2 \cdot \tan(\gamma)}{L} \end{bmatrix} \cdot V_c \quad (31)$$

Berez, lortutako ekuazioak abiadura irudikatzen dituzte eta balioen integrazioa eginez, AMR-ren posizioa mapa absolutu batean zein den jakin daiteke.



Irudia 54 Unibertsitateko AMR-ak kokapena mapa absolutu batean

### 1.3.5 Barne lokalizazioa

LiDAR-raren integrazioa burututako simulagailua prestatu ostean eta Unibertsitateko AMR-ren modelo matematikoa garatu ostean, barne nabigazioa planteatzeko unea da. Aurrekarietan ikusitako arabera EKF-SLAM sistema linealekin soilik funtzionatzen du. Horrek plataforma mugikorretan lokalizazioa burutzeko zenbait baldintza finkatu behar dira. Aldakortasun guneetan sistema linealak huts egiten dute. Horregatik, partikulen iragazkian oinarritutako algoritmo bat garatzea proposatu da. Baita sare neuronal batean oinarritutako lokalizazio bat ere.

Partikulen iragazki batean, posizioa estimatzeko eta gurpilek oinarritutako lokalizazioaren zehaztapena handitzeko, partikulak erabiltzen dira. Berez, partikulei mapa barnean posizio bat ematen zaie eta partikula bakoitzak posizio horren arabera mapako elementu desberdinak ikus dezakete. Partikula bakoitza ikusten duena, LiDAR-rak une finko batean ikusten denarekin konparatzen da eta partikula bakoitzari probabilitate balio bat ematen zaio. Garatu beharreko funtzioak aurre ikusteko, hurrengo lerroetan hasierako uneetan pentsaturiko urratzen aztertzen dira.

Lehendabiziko pausua AMR-ren posizioaren berritze funtzioa dator, ikusi ( 32 ). Non  $\vec{X}_i(t - 1)$  aurreko aldiunean definituriko AMR-ren posizioa den eta  $\vec{U}(t)$  motorrek sentsoeren bitartez neurtutako desplazamendua den. Funtzioa garatu oztean,  $\vec{X}_{partikula}(t)$  posizio berria lortzen da.

$$\vec{X}_{partikula}(t) = \vec{f}(\vec{X}_i(t - 1), \vec{U}(t)) \quad (32)$$

Aipatu den moduan, maparen azterketa partikulen bitartez egiten da eta hauen posizioaren arabera, gune ezberdinetan mapa osoan zehar oztopo ezberdinak aurkituko dituzte. Horretarako, LiDAR-ra simulatzen duen kodea erabiltzen da (begiratu Kodea 8 edota Kodea 11). Partikula hauek  $\vec{X}_{partikula}(t)$  egindako maparen informazioa beharko dute, mapatik informazio laginketak egiteko.

$$[\vec{Z}(t)] = \text{Mapa}_{agv}(\vec{X}_{partikula}(t), \text{Obj}) \quad (33)$$

( 33 ) ekuaziotik  $\vec{Z}(t)$  lortzen da, non simulatutako LiDAR-rak ikusten dituen oztopoen posizioak gordetzen diren. LiDAR-raren benetako posizioa eta simulatutako LiDAR-raren arteko ezberdintasunen arabera, partikula bakoitzari pisu bat egokituko zaio emandako tartearen arabera, begiratu ( 34 ) ekuazioa. Lorturiko pisuen arabera,  $\hat{X}_q(t)$  AMR-ren posizio berria estimatu daiteke, emandako probabilitatearen arabera.

$$w_i = G(\vec{Z}_{Lidar}(t) - \vec{Z}(t)) \quad (34)$$

$$\hat{X}_q(t) = \text{Arg}_{min} \|\vec{Z}_{Lidar}(t) - \vec{Z}(t)\| \quad (35)$$

Behin estimatutako posizioa lortuz, konparatu daiteke ea maparen objektuak lekuz aldatu diren. Azken finean, SLAM batek lokalizazioa egiten duen bitartean, mapak dituen objektuak berriro lokalizatzen ditu. Horretarako, AMR-ren estimatutako posizioaz baliatuz, LiDAR-ra simulatzen duen funtzioa abiarazten da, objektu berriak antzemateko. Azkenik,  $\alpha$  konstante baten arabera, berritze prozeduraren mugatu egiten da. Mapa bizkorregi aldatzen baldin bada, partikulen iragazkiaren SLAM-ak iragarpen txarrak emango ditu. Berez, tesi hau aurkezten den unean, industriako AMR-ak ( 35 ), ( 36 ) eta ( 37 ) ekuazioak ez dituzte erabiltzen, soilik lokalizazioa egiten delako.

$$[\overrightarrow{Obj}_q(t)] = Map_{AMR}(\hat{X}_q(t), Obj) \quad (36)$$

$$Obj = Obj + \alpha \cdot (\overrightarrow{Obj}_q(t) - Obj) \quad (37)$$

Behin partikula iragazkien teoriako funtzioak ikusita, MatLab-en bitartez programazioa burutuko da. Non hasiera batean lokalizazioa soilik tratatuko den, merkatuan dauden AMR-ak bezala. Programazioa garatzerako orduan, ( 32 ) funtzioak aztertzen dituen sarrerak moldatu dira eta AMR aurreko aldiuneaz eta motorren sentsoeren desplazamenduz gain, zenbat partikula nahi diren ezarri dira (ikusi Kodea 16 or. - 147 -).

*Pseudokodea 14 ( 32 ) ekuazioaren programazioa*

---

```

1  PartikulaSortzailea():
2  Sarrera:
3  AMR-ren azkeneko Posizioa
4  Partikula Kopurua
5  AMR-ak daraman desplazamendua (Motorretik neurtuta)
6
7  Irteera:
8  Partikulak
9
10 Prozedura:
11 For(q=1-etik Npartikula arte):
12   X_rand(q) = Abiarazi AMR-ren X posizioaren inguruan zenbaki aleatorio bat
13   Y_rand(q) = Abiarazi AMR-ren Y posizioaren inguruan zenbaki aleatorio bat
14   Z_rand(q) = Abiarazi AMR-ren Z posizioaren inguruan zenbaki aleatorio bat
15   Partikularen X posizioa lortzeko, X_rand + AMR-ak eragindako desplazamendua
16   Partikularen Y posizioa lortzeko, Y_rand + AMR-ak eragindako desplazamendua
17   Partikularen Z posizioa lortzeko, Z_rand + AMR-ak eragindako desplazamendua
18 End For

```

---

Behin, partikulak abiarazita eta hauek AMR barnean dagoen mapaz baliatuz, partikula bakoitzeko informazioa biltzen da. Informazio honek errealitatean LiDAR-rak zer ikusten denarekin konprobatu behar da, posizio estimatu bat lorrarazteko, begiratu ( 34 ) eta ( 35 ) ekuazioak. Beraz, partikula batek simulatutako LiDAR-raren informazioa errealitatearekiko zenbat aldentzen den aztertu behar da eta horretarako algoritmo baten beharra dago.

Lehendabiziko frogetan, bi aldiune desberdinetako argazkiak erabiliz, desplazamendua eta biraketa kalkulatzeko duen algoritmo bat proposatzen da. Argazki hauetan puntu esanguratsuetako irudikatzen dira eta bi aldiuneko puntuen arteko bateragarritasuna bilatu behar da. Algoritmo honek Iterative Closed Point (ICP) deritzo. Lokalizatorako, aldiz, ICP-ri emango zaion informazioa, LiDAR-rak identifikatutako oztopoetatik dator. Hau da, argazkietako puntu esanguratsuak izan beharrean, oztopoen lokalizazioa ematen zaio. (begiratu Kodea 17, or. - 148 -).

*Pseudokodea 15* LiDAR-ren informazioaz baliatuz, desplazamendu eta biraketaren kalkulua

```
1 ICP Erregresioa():
2 Sarrera:
3 Data1 -> Lehengo Aldiuneko datuak
4 Data2 -> Bigarren Aldiuneko datuak
5 Erabil nahi den metodologia
6
7 Irteera:
8 R errotazio matrizea eta T translazio matrizea
9
10 Prozedura:
11 While(Errotazio eta Translazio errorea < Nahi den errorea):
12 Bilatu bi aldiunetik zein ordena errore minimoa sortzen duen
13 If (Iterative Closed Point erabiltzen baldin bada):
14 SVD funtzioa aplikatu -> R1 eta t1 lortzeko
15 Else If(Erregresio Lineala erabiltzen baldin bada):
16 LR funtzioa aplikatu -> R1 eta t1 lortzeko
17 End If
18 Data2 = R1 * Data2 // Data2 biratu
19 Data2 = Data2 + t1 // Data2 mugitu
20 R = R1*R -> Errotazioaren minimizazioa lortzen da
21 T = R1*T + t1 -> Translazioaren minimizazioa lortu
22 Errorrea ebaluatu minimizazioa lortu den jakiteko
23 End While
```

Garatutako ICP kodean optimizazio bat garatzen da, non *data2* informazioa eraldatzen doa minimo batera iritzi arte, ikusi Kodea 17-an hogeita seigarren lerrotik hogeita hamarrera. Hala eta be, beste zenbait prozedura egin behar dira optimizazio horretara iritzi baino lehen. Besteak beste, bi datu laginketek dituzten puntuak elkartasuna bilatu behar dira eta datuak ordean konkretu batean ezarriz, *Pseudokodea 16* abiarazten duen moduan. (Begiratu Kodea 18, or. - 148 -).

*Pseudokodea 16* Bi laginketen elkartasun kodea.

```
1 Bilatu Puntuen lokalizazioa ():
2 Sarrera:
3 Data1 -> Lehengo Aldiuneko datuak
4 Data2 -> Bigarren Aldiuneko datuak
5
6 Irteera:
7 Emandako ordena eta lortutako errorea
8
9 Prozedura:
10 For (q=1-etik Data1 bete arte):
11 Aurkitu Data1 ordean hoberena Data2-rekiko. Bi laginketa arteko errorea kalkulatu
12 Puntuz puntu errore minimoa sortzen duen ordena hautatzen da
13 Ordean horren errorea ebaluatu
14 End For
```

Kodea 17-aren hurrengo urratsa, desplazamenduaren eta biraketaren kalkulua egitea da. Bi metodo desberdin aurkezten diren. Alde batetik balio singularren deskonposizioaren (SVD) erabilpena (aztertu *Pseudokodea 17*). Bestetik, autoreak egindako erakarpen batetik erakargarritako funtzioa, hau da, erregresio lineal baten erabilpena (begiratu *Pseudokodea 18*).

*Pseudokodea 17* Singular Value Descomposition kodea

```
1 SVD Funtzioa():
2 Sarrera:
3 M -> Lehengo Aldiuneko datuak
4 Data2 -> Bigarren Aldiuneko datuak
5 Index-ren ordenaren balioa
6
7 Irteera:
8 Errotazio eta Translazio matrizeak
9
10 Prozedura:
11
12 mm = Data1 media
13 S = Data2 index arabera ordenatuta
14 ms = S-ren media
15 Sshifted = [S(1,:)-ms(1); S(2,:)-ms(2);]
16 Mshifted = [M(1,:)-mm(1); M(2,:)-mm(2);]
17
18 W = Sshifted*Mshifted';
19
20 Abiarazi Matlabeko svd funtzioa U,A eta V matrizeak lortzeko
21
22 Errotazioa lortzeko ekuazio aplikatu (U*V')'
23 Translazioa lortzeko ekuazioa aplikatu mm - R*ms;
```



*Pseudokodea 18 Erregresio linealaren kodea*

```

1 LR Funtzioa():
2   Sarrera:
3     Data1 -> Lehengo Aldiuneko datuak
4     Data2 -> Bigarren Aldiuneko datuak
5     Data2-ren ordenaren balioa
6
7   Irteera:
8     Errotazio eta Translazio matrizeak
9
10  Prozedura:
11  For(g=1-etik Data2 datu kantitate arte):
12    φ-ren matrizea osatu,  $\begin{bmatrix} -Y_{1i} & X_{1i} & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ X_{1i} & Y_{1i} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$  kontuan hartuz.
13    Y-ren matrizea osatu,  $\begin{bmatrix} X_{2i} \\ \vdots \\ Y_{2i} \\ \vdots \end{bmatrix}$  kontuan artuz.
14  End For
15  Erregresioaren ekuazioa planteatu  $\begin{bmatrix} \widehat{\sin(\alpha)} \\ \widehat{\cos(\alpha)} \\ \widehat{c}_x \\ \widehat{c}_y \end{bmatrix} = (\phi^t \cdot \phi)^{-1} \cdot \phi^t \cdot \bar{y}$ 
16  Lortu erregresiotik R eta T matrizeak

```

Behin biraketa eta translazio datuak lortuta, partikularen posizioaren baliagarritasuna kalkulatu behar da. Horretarako, Kodea 17-ek emandako emaitzak baliatuz probabilitate funtzio bat planteatzen da eta partikula guztien balio horren arabera posizioaren estimazioa eragiten da. Translazio eta biraketa balioak oso altuak baldin badira, probabilitate txikiak ezartzen dira (ikusi Pseudokodea 19).

*Pseudokodea 19 Probabilitate funtzioa*

```

1 Probabilitate Funtzioa():
2   Sarrera:
3     Partikularen posizioa
4     Errotazio eta translazio funtzioa
5
6   Irteera:
7     Posizio estimatua
8     Probabilitatea
9
10  Prozedura:
11  X,Y eta Thetaren posizio bakoitzeko 1/exp() oinarritutako ekuazio batekin posizio bakoitzaren probabilitatea estimatu egiten da.
12  Lortutako probabilitatea partikularen posizio bakoitzarekiko biderkatzen da, estimatutako posizio bat lortzeko

```

Barne funtzioak azalduta, partikula iragazkiaren funtzio nagusia Pseudokodea 20-n aurkitu daiteke (aztertu Kodea 22, or.- 149 -). Berez, simulatutako kasua denez, bai AMR-ren mugimendua, baita AMR-ren LiDAR-ra funtzioaren barnean kalkulatzen dira. Errealitatean, datu hauek sentsoretik hartuko lidakete.

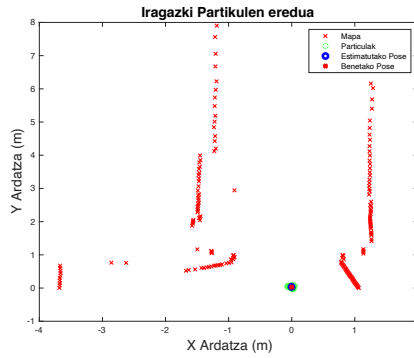
*Pseudokodea 20 Partikula iragazkiaren funtzio nagusia*

```

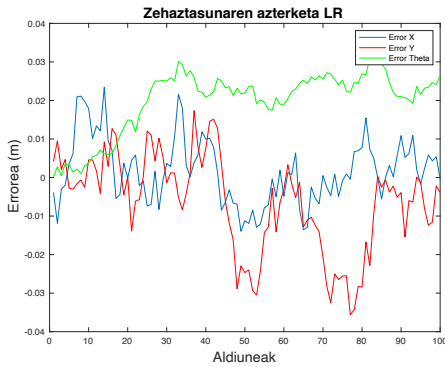
1 Mapa kargatu eta adierazi partikula kantitatea
2 AMR-ren hasierako posizioa definitu eta eramango duen abiadura eta norantza
3
4 Pseudokodea 14 abiarazi partikula sortzeko
5 PseudoKodea 7 abiarazi LiDAR-ra simulatzeko, AMR-ren posiziotik
6 Koordenatu Kartesiarretara pasa lorturiko LiDAR datuak
7
8 For(q=1-etik Partikula kantitate arte):
9   PseudoKodea 7 abiarazi, Partikula posizio bakoitzeko
10  Koordenatu Polarretik Kartesiarretara eraldatu LiDAR informazioa
11  PseudoKodea 15 abiarazi, translazioa eta errotazioa lortzeko
12  PseudoKodea 19 abiarazi, translazio informazioa metrotatik milimetrotara pasaz
13 End For
14
15 Estimaturako posea kalkulatu Estimaturako posizio guztien batuketa zatituz probabilitate guztien batuketarekin
16 Posizio datuak eguneratu lortutako estimazioaren arabera

```

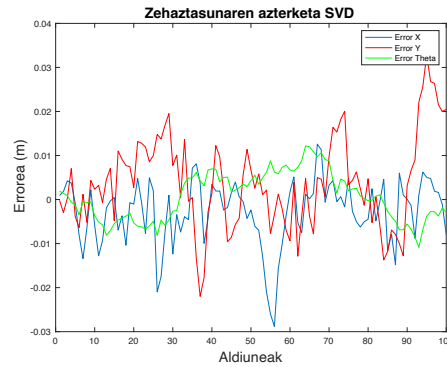
Egindako programa 100 laginketa abiaraziz datuak bildu dira. Bi simulazio egin dira, bata erregresio lineala erabiliz, bestea SVD funtzioa aplikatuz. Horrela bien arteko ezberdintasunak ikusiko egingo dira eta lokalizazioa estimatzeko zehaztasuna konparatzeko baliogarria izango da. Irudia 55a partikulen iragazkiaren funtzionamendua ikus daiteke. Frogapen honetarako eskolako korridorearen mapa erabili da.



(a)



(b)



(c)

Irudia 55 Partikulen iragazkiaren frogapena: a) Simulazio bitarteko leihoa, b) Erregresio linealaren erantzuna 100 laginketa ondorioz, c) SVD-ren erantzuna 100 laginketarako.

Erabilitako korridoreak berrogei metroko luzera eta hiru metroko zabalera ditu. Horrek posizioa interpretatzeko arazoak dakar, batez be luzeran. Gune batzuetan ezkerreko horma, baita eskuinekoa informazio gutxi daukate eta beraien artean simetrikoak dira. Horrek, Y ardatzaren lokalizazioan zehaztasuna galaraztea dakar, informazio eskasa dagoelako. Naiz eta abiarazitako ereduan gune simetrikora ez iritsi, fenomeno au ikus daiteke Taula 6-n biltzen dituen datuetan. Y Ardatzeko balioak X ardatzean lortutakoak baino zehaztasun gutxikoak dira. Ardatzen mediak aztertuta, errore altuena SVD-n 32 mm-tan eta erregresio linealean 4 mm-tan finkatzen dira. Beraz, erregresioak translazioak kalkulatzeko zehatzago da, aldiz errotazioak ikusterako orduan, balioak aldatzen dira. Laginketa honetan, erregresio linealak 0,0224<sup>o</sup>-ko mediana dauka eta SVD-rekin konparatuta, balio oso altua dela adieraz daiteke. Baina, 0,02<sup>o</sup>-ko zehaztasuna izatea barne nabigazio baterako, balio onargarri bezala har daiteke. Dena den, simulaturiko kasuak direnez, Unibertsitateko AMR-an frogatzerako orduan emaitzak eralda daitezke, AMR gordeta duen mapa errealitateetik alden du daitezkeelako.

Taula 6. Partikula Iragazkiaren konparaketaren erantzunak

Errore Mota	SVD erabiliz	Erregresio lineala erabiliz
X Errore media (m)	- 0,0028	0,0016
X Errore mediana (m)	- 0,011	0,0009
Y Errore media (m)	0,0006	- 0,0065
Y Errore mediana (m)	0,032	- 0,0039
$\theta$ Errore media (gradu)	0,0038	0,0198
$\theta$ Errore mediana (gradu)	0,0009	0,0224

### 1.3.6 Sare neuronalen bitartezko lokalizazioa

Behin LiDAR-raren funtzioa garatuta, lokalizazioa sare neuronal batekin ordezkatu daiteke. Posizionamenduan zehaztasuna lortzeko sare neuronal bakarra garatu beharrez, hiru sare proposatzen dira, bakoitza  $X$ ,  $Y$  edo  $\theta$  -ren iragarria egiteko. Non sarearen sarrera LiDAR-rak ematen duen informazioa izango den eta irteera posizioen iragarpena izango den.

Sare neuronal bat denez, lehengo jarduera datuen prestaketa da. Horretarako, Kodea 9-n burututako LiDAR funtzioa eta Kodea 8-n eraikitako mapaz baliatuz, 3.000 laginketa lortu dira, posizio aleatorioak erabiliz (aztertu Pseudokodea 21 eta Kodea 23, or. - 150 -).

Pseudokodea 21 Laginketak sortzeko kodea

```

1  Nahi diren laginketak ezarri
2  If(Laginketa Datuak baldin badaude):
3      Kargatu gordetako datuak
4  Else
5      For(q=1-etik Ezarritako Laginketa kopuru arte):
6          X_rand(q) = zenbaki aleatorio bat
7          Y_rand(q) = zenbaki aleatorio bat
8          Z_rand(q) = zenbaki aleatorio bat
9          Abiarazi Pseudokodea 7 LiDAR datuak izateko
10         If(LiDAR informazioa hutsik badago):
11             Cont + 1
12         Else
13             LiDAR informazioa gorde
14         End If
15     End For
16     Gorde lorturiko informazioa
17 End If

```

Sarearen sarrera 720 balioko bektore bat izango da. Lehengo 360 datuak LiDAR-rak emandako erradioen balioak izango dira eta hurrengo 360-ak angeluak dira. Sareak koordinatu polarrak erabiltzen ditu. Behin datuak prestatuta sarea entrenatzeko unea da. Fitnet izaerako sarea aukeratzen da. Geruzak eta neuronak kopuru finko aukeratu beharrez, gutxienez  $Sare_{laginketa} = Neurona^{Geruza}$  frogapenak ebaluatu egiten dira. Aukera honekin zenbat geruza behar diren eta geruza bakoitzak zenbat neurona ezarri behar diren ebaluatu egiten du, sare egitura egokiena lortuz. Egiturak hori aukeratzeko, sareak 100 posizio aleatorioekin ebaluatzen da (ikus (38) ekuazioa). Errore txikiena lortzen duen sareak fitxategi batean gordeko da, Pseudokodea 22 aurkezten duen moduan.

$$Errorea = \frac{\sum \sqrt{(R_i - R_{sarea})^2}}{100} \quad (38)$$

Pseudokodea 22 Sare egokiena aukeratu

```

1  Sarea Entrenatzeko Kodea():
2
3  Sarrerak
4  Sareak behar dituen Sarrera datuak, Sareak behar dituen Irteera datuak, Geruza Kopurua,
   Neurona Kantitatea, entrenamendu funtzioa, Zenbat Aldiz frogatu nahi den sarea, Mapa, LiDAR-
   raren distantzia maximoa, (X;Y;Theta)Sarea aukeratu
5
6  Irteerak:
7  Entrenaturiko Sarea eta sorturiko errorea
8
9  Prozedurak:
10 Sortu konbinazioak Combinator funtzioarekin
11 Combinator planteatzen ez dituzten beste aukerak sortu geruzak moldatuz, edo neuronak
   moldatuz
12 For(g=1-etik sorturiko konbinazio guztiak arte):
13   Prestatu sarearen antolaketa, konbinazio arabera.
14   Prestatu datuak, sailkatuz entrenatzeko datuak, ebaluatzeko datuak eta frogatzeko datuak
   bananduz
15   Sarea entrenatu
16   Lorturiko sarea frogatu laginketa berriekin, PseudoKode 23
17   If(Lorturiko sarea orain arte lortutako baino errore txikiagoa lortu badu):
18     Gorde sarea
19   End If
20 End For

```

Kodea 24-an ezinbesteko diren 2 funtzio behar dira. Lehengoa “Combinator” du izena, eta berrogeita hamabigarren lerroan agertzen da. Matlab-eko laguntza erabiliz lorrarazi den funtzioan da, dena den Kodea 40-an atxikita agertzen da. Berez, prozedura honek *Neurona<sup>Geruza</sup>* posible diren kasuak ateratzen ditu. Demagun bi neurona eta lau geruza daudela. Horrek, sarearen entrenamenduan, hamasei posible kasu aztertu beharko dira(ikusirudia 56a). Dena den, funtzio honek lau geruza iritzi arte sor daitezkeen konbinaketak baztertzen ditu. Adibidez, geruza bat bi neuronekin. Horregatik, kasu guztiak lorrarazteko berrogeita hamaseigarren lerrotik berrogeita hamazortzigarrenera beste zenbait eragiketak egin dira. Erabilitako adibidearentzako 30 kasu aztertu beharko lirateke, sarearen egitura determinatzeko (begiratu Irudia 56b).

>> combinator(2,4,'p','r')

```

ans =
   1   1   1   1
   1   1   1   2
   1   1   2   1
   1   1   2   2
   1   2   1   1
   1   2   1   2
   1   2   2   1
   1   2   2   2
   2   1   1   1
   2   1   1   2
   2   1   2   1
   2   1   2   2
   2   2   1   1
   2   2   1   2
   2   2   2   1
   2   2   2   2

```

(a)

Konbinazioak =

```

   2   0   0   0
   1   0   0   0
   1   1   1   1
   1   1   1   2
   1   1   2   1
   1   1   2   2
   1   2   1   1
   1   2   1   2
   1   2   2   1
   1   2   2   2
   2   1   1   1
   2   1   1   2
   2   1   2   2
   2   2   1   1
   2   2   1   2
   2   2   2   1
   2   2   2   2
   1   1   0   0
   1   2   0   0
   2   1   0   0
   2   2   0   0
   1   1   1   0
   1   1   2   0
   1   2   1   0
   1   2   2   0
   2   1   1   0
   2   1   2   0
   2   2   1   0
   2   2   2   0

```

(b)

Irudia 56 Combinator funtzioaren frogapena, bi neurona eta lau geruza erabiliz: a) 52. lerroaren funtzioa abiarazita, b) 52. lerrotik 68-ra kodea abiarazita

Bigarren funtzioa “TestNet” du izena eta ekuazio ( 38 )-a burutzen du. Zazpigarren lerrotik hamargarrenera 100 posizio aleatorio osatzen dira. Baliteke posizioen bat mapatik at egotea, “MedidasLIDAR” funtzioak MaxLidar limitetik at dagoen balioa emanaz. Horregatik, posizio horretan objekturik ikusten ez bada, zuzenean posizio hori baztertzen da, posizio berri bat kalkulatu.

Pseudokodea 23 Errorearen kalkuluaren funtzioa

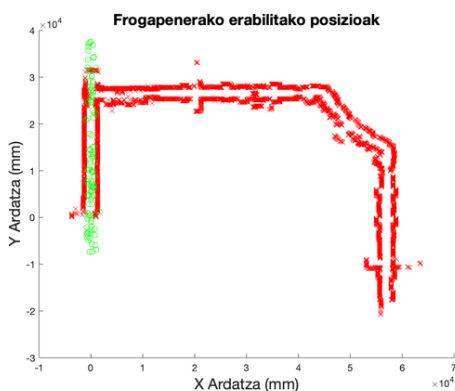
```

1  Sarea frogatzeko kodea Kodea():
2
3  Sarrerak
4  Zenbat posizio konprobatu nahi diren, Mapa, LiDAR-aren distantzia maximoa, sarea eta
   zein(X,Y,Theta) sareak frogatu nahi diren
5
6  Irteerak:
7  Errorea
8
9  Prozedurak:
10 For(q=1-etik frogapen kopuru arte):
11   While(Mapa Barnean dagoen posizio bat aurkitu arte):
12     XPosizioa = Zenbaki aleatorioa
13     YPosizioa = Zenbaki aleatorioa
14     ThetaPosizioa = Zenbaki aleatorioa
15   End While
16   Lorturiko posizio aleatoriotik PseudoKode 7 erabili LiDAR-retik datuak lortzeko
17   Koordenatu polarretik kartesiarretara pasa
18   Abiarazi sarea
19   Errorea = Errorea + sqrt(Sarearen posizio estimazioa - Posizio Aleatorioa)^2
20 End For
21 Errorea = Errorea/frogapen kopurua

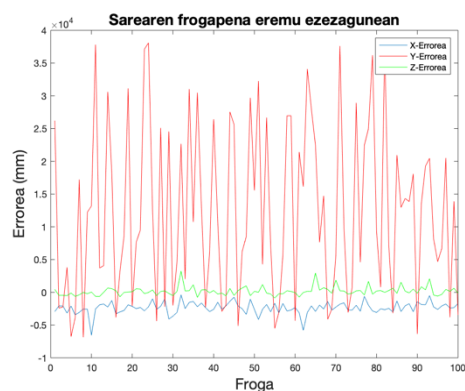
```

Laburbilduz, kode honek lokalizatorako sare neuronalen entrenamenduak burutzeko baliagarria da, non neuronen eta geruzen arteko konbinazio ezberdinak sortzen diren, errore minimoa sortzen duen sarea bilakatzeko. Sare egokiena hautatzeko, sareak 100 posizio aleatorio ezarriz frogatzen da eta ( 38 ) ekuazioaren arabera, errore minimoa sortzen duen sarea hautatzen da.

Kontuan izanda posizio bakoitzarentzako sare bat sortzen dela, lau geruza eta hogei neurona ezarri direla balio maximo bezala eta 168.420 kasu aztertu hoztean lortutako emaitzak honako hauek dira. X posizio absolutuan ebaluatzen duen sareak, 29 mm errorea lortu du. Kasu txarrean 2.947,4 mm (ikusirudia 57). Errore minimoa sortzen duen sareak, lehenengo geruzan bi neurona ditu, bigarrenean zortzi, hirugarrenean hamabost eta laugarrenean zortzi. Beraz, ikusten den moduan, geruza bakoitzak haren neurona kopurua fintzen du, aukeratuz egitura egokiena. Y posizio absolutua kalkulatzeko duen sareak, lehenengo geruzan bederatzi neuronak ditu, bigarrenak hamabost, hirugarrena bederatzi eta laugarrena hemezortzi. Sareak sortzen duen errorea 0.0792 mm-takoa da. Azkenik errotazioa kontrolatzen duen sareak da eta entrenamendua amaitu ondoren, errore minimoa lortu duen sareak 0,0067 gradu dira. Egitura honako konposizioa dauka: Lehenengo eta hirugarren geruzetan hemeretzi neurona, bigarren geruzan hemezortzi neurona eta azkeneko geruzan hamar neurona.



(a)



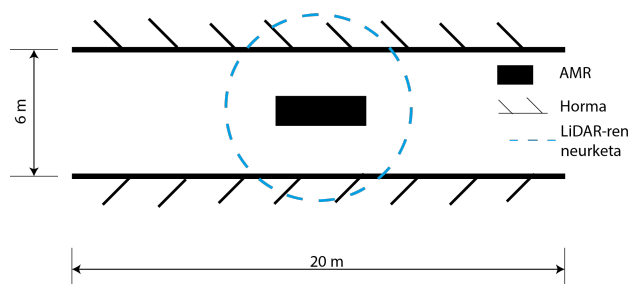
(b)

Irudia 57 Sare neuronalen entrenamenduaren ondorioz egindako frogapena: a) Entrenamenduaren mapa eta entrenatzeko posizioen errepresentazioa, b) frogapenaren erroreak erantzuna

Irudia 57a-n ibilbide baten froga puntuak dira eta Irudia 57b lorturiko emaitzak. Naiz eta sareak balio minimo batzuk lortu, nabigazioa hasten denengan, Y ardatzaren iragarpenak txarto egiten ditu. Horrek LiDAR-rrak duen luzeraren ondorioz dator, non korridoreren hasierako eta amaierako hormak ikusten ez dituzenez eta paretak simetrikoak direnez, lokalizazioan zehaztasuna galtzen da. X ardatzean, zenbait unetan berdin gerta lekiok, baina sortzen duen errorea Y-rekin konparatuz askoz txikiagoa da. Tesia entregatzerako unean, sarearen frogapena eremu errealean ez da frogatu.

### 1.3.7 Sare Neuronalak Nabigazio algoritmoentzako

LiDAR-rak dituen luzera mugengatik eta egindako frogapen ondorioz, AMR-ak korridore luze eta simetrikoetan galdu daiteke. Irudia 58-an adierazten den moduan, LiDAR-rak informazio esanguratsua ezin du lortu, bertatik lokalizatzeko edo mugimendua antzemateko. Bertan AMR -ren gurrupilak labaintzen hasten baldin badira, AMR-ren lokalizazioa guztiz galduko litzateke.



Irudia 58 LiDAR-rak dituen limitazioak korridore luze eta simetrikoetan

Horrekin bat, industrian erabiltzen diren LiDAR gehienak bi dimentsiokoak dira, beraz ingurumenean zehar informazio galerak daude. Horrek sentsoreaz aldatzeko aukera bat suertatu du, ikusmen artifizialaren erabilpena proposatuz.

Inteligentzia honen arloko lehengo pausua Mercedes – Benz barnean eman dira (aztertu 2.3.1 atala). Berez, atal horretan sare neuronalak irudian objektu bat identifikatzen du eta lokalizazio horren arabera, robotaren eta AMR-ren koordenadak zuzendu daitezke. Prozesu konkretu horretatik baztertuz, halako objektuen identifikapena, sailkapena, segmentazioa, etab. baliatuz AMR-arentzako nabigazio algoritmo baten garapena burutu daiteke. Sekzio honetan, plataforma mugikorrek edozein eremutik mugiarazteko, segmentazio sare baten garapenez hitz egiten da.

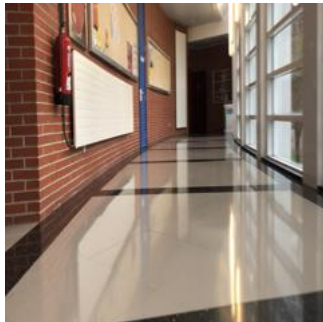
Segmentazio sare batek irudian dauden objektuen sailkapena egiten du, horrela irudian zehar objektuak banandu egiten ditu. Banaketa hori egiten denean, irudia iragazi egiten dela eza daiteke. Adibidez, sareak lur zorua identifikatuz gero, lur zorua ez den beste guztia baztertuko luke (begiratu Irudia 59). Filtro honek, PixelLabel bezala ezagutzen da.



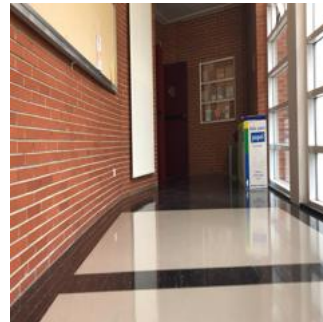
Irudia 59 Sare neuronalak egiten duen filtroa, objektu ezberdinak sailkatzeko

Edozein gainbegiratu gabeko teknikak garatzeko, datuak bildu behar dira. Segmentazio sarea entrenatzeko, argazki multzo bat behar du eta multzo horren irudi bakoitzaren pixel bakoitzari zein taldeari egokitzen zaion adierazi behar zaio. MatLab-aren kasuan, “*Image Labeler*” izeneko erreminta bat eskaintzen du, prozedura honen lan zama murrizteko. Gainera, halako erremintak automatizatu daiteke. Adibidez, jadanik entrenaturiko sareak erabiliz, pixel bakoitza sailkatu daiteke era automatiko batean. Erabiltzaileak sailkapen hori ondo dagoen adieraziko du soilik.

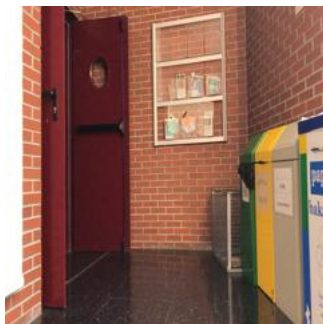
Datu bilketa dela eta, hasiera batean Vitoria – Gasteizko Ingeniaritza eskolaren korridoreetan oinarritu zen eta bi eremu identifikatzen ziren. Alde batetik lurzorua eta bestetik lurzorua ez den beste guztia. Nabigazio algoritmoa garatzeko lurzoruaren informazioa erabiliko da soilik, horrela robot mugikorrek lurzurutik zehar duen espazioa interpretatu dezake eta nabigazio teknika bat erabiliz, edozein eskolako eremutik mugiarazi daiteke, ikusi Irudia 60.



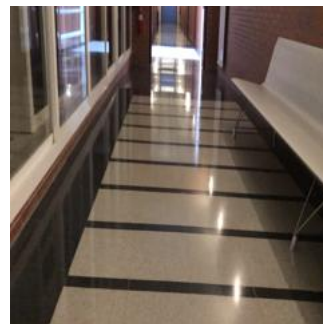
(a)



(b)



(c)



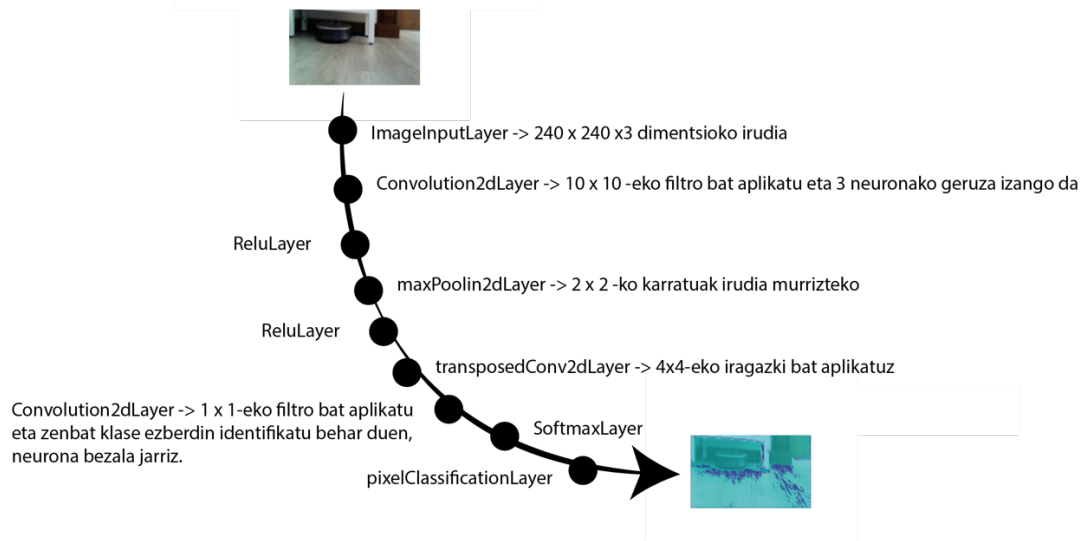
(d)

Irudia 60 Vitoria – Gasteizko Ingeniaritza Eskolaren korridoreen argazkiak: (a)(b)(c)(d) sare neuronalaren entrenatzeko erabilitako ereduak

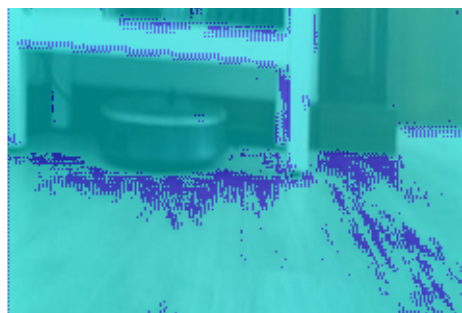
Behin datuak prestatuta izanda, sare neuronalaren egituraren determinatu behar da (aztertu 2.3.1 atala). Informazioa ez biderkatzeko, atal honetan segmentazioari buruz hitz egingo da soilik. Segmentazioa egiten duen sare neuronal baten egitura sinplea Taula 6-n biltzen da. Bertan, sareak behar dituen geruzak aipatzen dira eta Irudia 61a-n proposaturiko egitura bat planteatzen da.

Taula 7. Segmentazio Sare Batek dituen geruzak

Geruzak	Deskribapena
ImageInputLayer	Irudien sarrera egiten duen geruza. Bi dimentsiotako irudiak izango dira eta datuak normalizatuak izango dira.
Convolution2dLayer	Konboluzio geruzak bi dimentsiotako filtro labaiagarri bat aplikatzen du, haren sarreran. Mugimendu horrek bai horizontalik baita bertikalki aplikatzen du. Prozedura honi esker, produktu baten kalkulua egiten da, ezarritako pisuen eta sarreraren arabera, baita ezarritako alborapen bat kontuan hartuz
reluLayer	Atarizko eragiketa bat egiten du, non haren sarreran elementu bakoitzaren balio aztertzen du eta zero baino txikiago diren balioak, zero bezala ezartzen ditu.
maxPooling2dLayer	Geruza honek sarrerako datuak sekzio karratuetan elkartzen ditu datuak, gune bakoitzaren balio maximoa kalkulatu. Hau da, irudia txikitu egiten da, informazio esanguratsuekin geratuz
transposedConv2dLayer	Konboluzioaren iraulketa aplikatzen duen geruza
softmaxLayer	Sarrerako datuak, 0 eta 1 bitarteko datuetan banatzen ditu
pixelClassificationLayer	Sarreraren amaieran ezartzen den geruza. Pixel bakoitzaren bereizketa egiten duen geruza da.



(a)



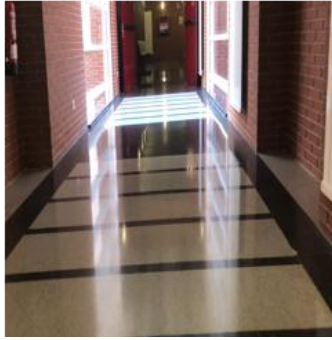
(b)

Irudia 61 Egitura sinpleko Sare neuronal bat: a) Sarearen egitura, b) Sareak entrenatu ostean emandako erantzuna, jakinda urdina lurzorua adierazten duela

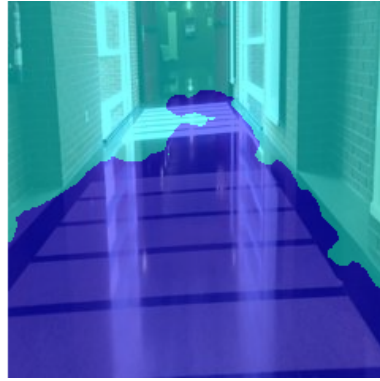
Halako sare sinpleekin zoru beste denetik bereizteko zailtasunak ditu, sare sinpleak argazki konplexu batean agertzen diren parametro guztiak ikasteko zailtasunak dituelako. Horretarako beste teknika batzuk aurki daitezke. Adibidez, “*Transfer Learning*” erabili daiteke. Teknika honek beste proposamen baterako entrenatuko sare bat erabili daiteke, irudi bat segmentatzeko. Hala ere, proposamen desberdinetarako planteatutako sareak direnez eraldaketak egin beharko dira. Entrenaturiko sare batek geruzen pisuak abiarazita dituzte eta







(a)



(b)

Irudia 63 Lehengo ResNet – 18 Segmentazio sarean lorturiko emaitzak: a) Sareari emandako ez ikasitako irudia, b) Sareak sorturiko erantzuna, urdinez lurzorua adieraziz

Behin ikusita halako sare batek nabigazio gunea interpretatu dezakeela, sareak emandako informazioaz baliatuz nabigazio algoritmo bat planteatu daiteke. Azken finean, sareak maskara bitar batzuk ematen ditu. Kasu partikular honentzako, bi maskara daude (Lurzorua eta lurzorua ez den beste guztia).

Lurzoruaren maskara erabiliz, erdiko ibilbide bat interpretatu daiteke, AMR-ak ibilbide hori jarraitzeko. Ekuazio ( 39 )-an agertzen den moduan, irudiak lerroz lerro datuak horizontalik ateratzen dira. Demagun irudia 10 x 20-eko dimentsioa duela, orduan i-ren balioa batetik hamarrera izango lirake.  $Hasi_{lokalizazioa}$  eta  $Bukatu_{lokalizazioa}$  aldagaiak espazio librea nondik nora doan markatzen du. Bi aldagai hauei esker, erdiko puntuak estimatu daiteke. Planteaturiko adibidean, hamar erdiko puntu egongo dira guztira. Erdiko puntu hauek argazki zentroarekin konparatuko dira eta kasu txarrean AM-REN noranzkoa adieraziko du (aztertu ( 40 ) ekuazioa). ( 40 ) ekuazioan 120 balioak argazkiaren erdiko balioa zehazten du. Berez RestNet 240x240-ko argazkiak erabiltzen dituelako.

$$Erdiko_1 = \frac{Bukatu_{lokalizazioa} - Hasi_{lokalizazioa}}{2} + Hasi_{lokalizazioa} \quad (39)$$

$$[Biraketa, lekua] = \max(abs(120 - Erdiko)), \quad (40)$$

Programa nagusia *Pseudokodea 25* arabera garatu da (aztertu Kodea 26, or.- 152 -). Kode horren exekuzio ondorioz, Irudia 64 lorturiko erantzunak irudikatu daiteke. Irudi horiek sareak ez ikasitakoak dira, horregatik argi aldatetekin ibilbidearen kalkulua egiteko arazoak ditu, Irudia 63-an gertatzen den moduan.

*Pseudokodea 25 Segmentazio sareak emandako erantzunetik erdiko puntu atera*

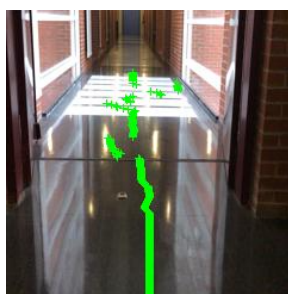
---

```

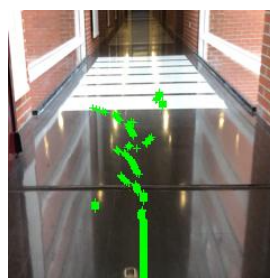
1 Kargatu entrenatutako sarea
2 For(q=1-etik aztertu nahi diren argazki kopuru arte):
3   Ireki aztertu nahi den irudia
4   Sematicseg(Irudia(q)) MatLab-eko funtzioa abiarazi sareak argazkia segmentatzeko
5   Atera lurzoruko maskara
6   Kalkulatu erdiko puntuak, ekuazio ( 39 ) oinarritutako funtzioa erabiliz
7   If(Aurkitu baditu erdiko puntuak):
8     Adierazi puntuak irudietan
9   End If
10 End For

```

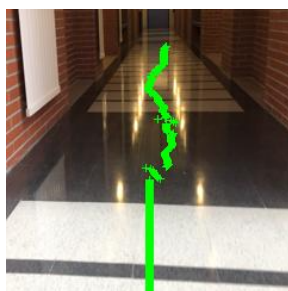
---



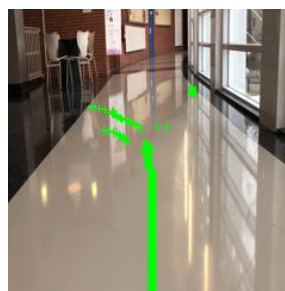
(a)



(b)



(c)



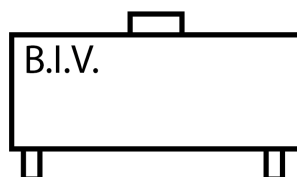
(d)

Irudia 64 ResNet – 18 segmentazio sareak emandako emaitzak erdiko punturen kalkulua aplikatuta

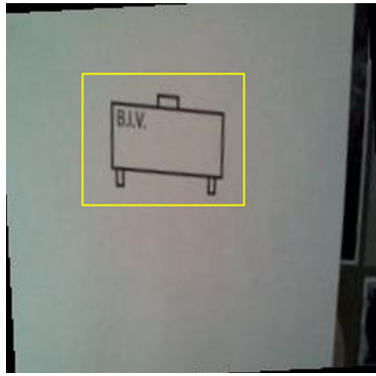
Halako erantzunak ikusita, AMR-ak edozein ikasitako eremutik mugiarazi daitekeela ikus daiteke. Horretarako abiadura eraldatzen duen beste funtzio bat planteatu da. Berez, motorrei kontsigna finko batekin mugituko dira, adibidez 10 mm/s. Biraketa bat egin behar badu, orduan motor bati kontsigna horri murriztuko litzateke.

(40) ekuazioan *Lekua* parametroak, hautatutako erdiko puntuaren altuera ezartzen du. Altuera horren arabera abiadura ere eralda daiteke. Berez erdiko puntua oso urrun baldin badago, abiadura maximoarekin mugiarazi daiteke. Aldiz, erdiko puntu hori, AMR-retik oso gertu dagoenean, abiadura murriztuko da. Programaturiko segmentazio funtzioei ezker, Mathematics barnean argitalpena lorrarazi zen, tesi honek lantzen dituen gaien buruzko argitalpen inportanteena. Segmentazioaz gain, sare neuronaletan oinarrituko beste nabigazio algoritmo bat lor daiteke. Eremuko objektuak identifikatuz, AMR-ren kokapena determinatu daiteke, agindu desberdinak bidali ahal zaizkio, etab.

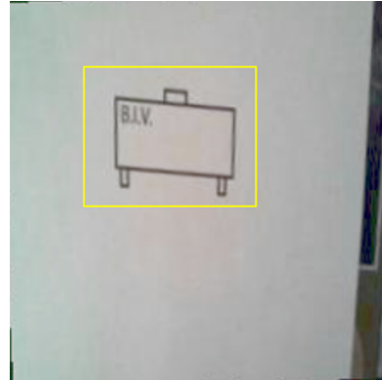
Barneko plataforma mugikorrek trafikoa arautzen duten legedirik ez dagoenez, eraikuntza barneetan inolako seinalerik ez dago. Horregatik, sare neuronal konboluzionalak erabiliz, irudi bateko objektuan identifikapena eta kokapena determinatu daiteke. Seigarren atalburuko 2.3.1 atalean oinarritutako sareak erabili daiteke, baina furgonetaren pieza bat identifikatu beharrean, sinbolo konkretu bat identifikatu dezake (begiratu Irudia 66). Sarea entrenatzeko, "*Bounding Box (BBox)*" izeneko karratuak irudiz irudi prestatu behar dira. Prozedura honek segmentaziotik bereizten da, bereziki planteatutako bigarren inteligentzia honek, karratuaren kokapen koordinatuak emanda, identifikatu nahi den objektua non dagoen adierazten zaiolako sareari. Aldiz segmentazioan pixel bakoitzari, kategoria bat ezartzen zaio.



(a)



(b)



(c)

Irudia 65 Sare Neuronal Konbuluzionalak aplikatuz, logo konkretu baten identifikapena: a) Erabilitako logoa, b)c) lorturiko emaitzak

Kasu konkretu honetarako MatLab 2019b erabili da, sareen entrenamendua errazteko bertsio horretan zenbait funtzio berri inplementatu dira. Horien artean, edozein sare neuronal konbuluzionala objektu identifikatzaile bihurtzeko funtzioak. Horrek seigarren atalburuko 2.3.1 atalean aurkeztutako programazioa, erabat eraldatzen du. Bertan sarearen geruzak eskuz moldatu behar dira, aldez gaur egungo MatLab-en funtzio berezi horrekin era automatiko batez moldatzen dira. Pseudokodea 26 (ikusi Kodea 27, or. - 153 -) erabiliz Irudia 65b eta c agerturiko erantzunak lorrarazi daitezke.

#### Pseudokodea 26 Sare Konbuluzional batek objektuak identifikatzeko

---

1	Kargatu entrenamendurako prestatutako datuak
2	Banandu entrenatzeko datuak eta frogatzeko datuak.
3	Estimatu AnchorBoxes parametroa, estimateAnchorBoxes Matlab-eko funtzioarekin
4	Ireki erabiliko den sare neuronala
5	Adierazi zein geruza alda behar da, sareak objektuak identifikatzeko
6	Abiarazi yolov2Layers funtzioa sarearen egitura moldatzeko
7	Entrenamendua egin
8	Frogatu sarean

---

Tesia aurkeztzen den unean, nabigazio algoritmoa garapen prozeduran dago. Dena den, segmentazio sareak funtzionatzen duen modua ikusita, objektuen identifikapena AMR-ri agindu bereziak emateko erabiliko litzateke. Horrez gain, objektuen identifikapena lokalizatorako erabili daiteke, azken finean sareak identifikatutako objektuak landmark bezala funtzionatzen dutelako. Adibidez, logo barnean kodifikazio berezi bat ezarriz, AMR-ak non dagoen jakin dezake eta horren arabera erabakiak hartu. Bi sare erabiltzen baldin badira aldi berean, prozesadorearen etekina murriztu daiteke. Horren ondorioz, ekipamendu ahaltsuagoak beharko dira.

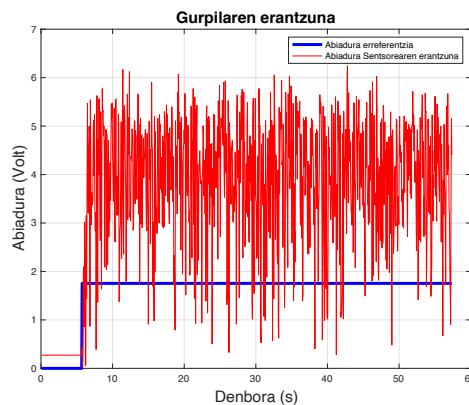
#### 1.4 Lehen Prototipoaren Gurpilen kontrola

Edozein lokalizazioa egiteko, bai gurpilen kontrola, baita abiadura neurtzen duten sentsoreen irakurketa egin behar da. Gurpil hauen integrazioak oso bizkorak izaten omen dira, batez ere beharrezko informazioa parametrizatuta dagoelako. Gasteizko Ingeniaritza Eskolan planteatutako lehengo prototipoan aldez, patinete elektrikoak erabiltzen dituzten gurpil modukoak ezartzea erabaki da. Motor hauek, espazio gutxi behar dute, bati-bat motorra gurpilean integratuta dagoelako. Efizientzia aldetik, eraginkorrak dira ere. Beraz, AMR - rentzako egokiak direla pentsatu daiteke, baina saltzailetik informazio faltaren ondorioz arazoak etor daitezke.

Gurpilaren izaeragatik arazo gutxi daude, problema guztiak gurpilak behar dituzten kontrolagailuetatik dator. Kontrolagailu hauek korrante zuzena korrante trifasiko batean bilakatzen dute, motorra abiarazteko. Motorren abiadura seinale baten bidez egiten da, baina abiaduraren seinalearen izaeraren informazioa ez dago.

Bi motatako seinaleak konektatu ahal dira. Alde batetik zabalera modulatuak pultsua (PWM) dago, bestetik seinale analogiko bat. Bi kasuetarako seinalearen tentsio maximoaren muga ez dago zehaztuta. BLDC motorretan ikusitako arabera, abiadura PWM-ren bitartez finkatzen da. Horregatik lehengo pausuak halako seinale motarekin abiarazi dira.

AMR-ak dituen PWM seinaleak 24 V-ko tentsio bitartez osatzen dira. Multimetro bat seinale irteera horretara konektatzen baldin bada eta PWM % 50 ezartzen baldin bada, multimetroan 12 V adieraziko ditu. Baina seinale hori 24 V duen seinale batez osatuta dago, beraz zenbait elementu elektronikoa matxuratu daitezke. Halako egoera arazoei aurre egiteko Argolabe Ingenierian S.L. zirkuitu elektronikoa planteatu du, 0 V – 24 V bitarteko tentsioa izan beharrean, 0 V – 5 V izateko. Bestalde, kontrolagailuak motorrak duen abiadura neurtzeko beste seinale bat dauka eta hasieratik seinale analogiko bat dela pentsatu da. Horregatik zuzenean Beckhoff-eko EL3104-ra konektatu da.

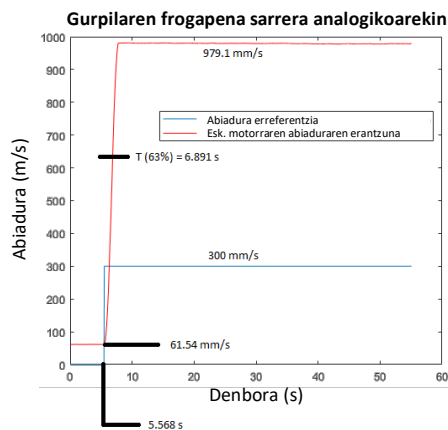


Irudia 66 Eskuineko gurpilaren lehengo frogapena

Lehengo frogapena egin ondoren, lortutako datuen arabera, gurpilaren kontrola ezin da egin (ikusi Irudia 66). Seinale hori era eraginkor batean irakurtzeko 0,5 segundoko atzera sartzen duen iragazki bat sartu behar da. Beraz, lehengo frogapen hauen ondorioz, sistemaren kontrola egiteko egoera hau eraldatu behar da. Dena den, ondorio batzuk lortu ziren. Adibidez, abiadurako sentsoreak gurpila geldituta dagoenean, 0,9 V-eko tentsioa itzultzen du. Horrek abiadura geldoekin informazio galerak egongo direla adierazten du.

Gurpilaren izaera berriz pentsatuz, patin elektriko batean muntatzeko prestatuta dagoenez, tresneria hauetan abiadura kontsignak nola emetan diren aztertu zen. Patinetetan, motorretan bezala, manillarrean dagoen katuaz baliatuz, abiadura eraldatzen da. Katu horiek potentziometro batean oinarrituta daude. Orduan AMR-aren abiadura kontrolatzeko, PWM izan beharrean, seinale analogikoa izango da.

Horren ondorioz abiadura kontsigna EL4134 bitartez kontrolatuko da. Horrez gain, bi konstante kalkulatu dira, Voltekoetatik abiadurera bilakatzeko. Lehengo konstantea lortzeko, takometro bat erabili da. Esperimentuan EL4134 Volt ezberdinak ezarriz, Volt bakoitzeko biraketa minutuko (RPM) neurtu dira,  $21,6 \frac{RPM}{Volt}$  konstantea lortuz. Bigarren konstantea gurpilaren ezaugarri mekanikoei lotuta dago. Azken finean, biraketa bat zenbat metro desplazatu den kalkulatzeko, zirkunferentzia baten luzeraren funtzioa erabili behar da. Ebazpen honen emaitza  $10,46 \frac{M/s}{RPM}$  da.



Irudia 67 Eskuineko gurpilaren frogapena kontrolagailuaren sarrerak ezagututa

Motorraren erantzuna guztiz eraldatu da, behi sarrerako era analogiko batean tratatzerako orduan. Horregatik PI izaerako kontrolagailu bat burutu da, batez ere Irudia 67 agertzen den erantzunaren ondorioz. Sistema irekian gurpilak kontsigna baino hiru aldiz abiadura gehiago darama eta hori onartezina da. Kontrolagailuaren parametroak hautatzeko sistemaren denboraren erantzuna erabiltzen da.

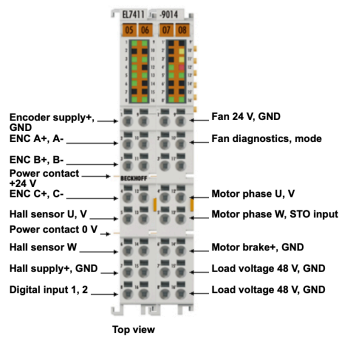
Dena den, naiz eta gurpilen kontrola hobetu, abiaduraren kalkuluan zehaztasun falta dago. Gurpil hauek 20 Km/h-ko abiadurak hartzeko prestatuta daude, baina AMR-ak gehienez 1 m/s-ko abiadura doa, legedia betetzeko. Hau da, AMR-ak 3,6 Km/h-ko abiadurarekin mugiarazten da. Abiadura sentsoreen ondorioz, motorrak aldatzeko erabakia hartu da.

### 1.5 Prototipoaren zehaztasunaren hobekuntza

Zehaztasuna lortzeko, bi prozedura mota egin ziren. Alde batetik BLDC motatako gurpilen kontrolak aldatu ziren. Bestetik, gurpil berrien fabrikazioa edota erostea proposatu zen. Dena den, zehaztasunaren hobekuntza nola garatu den ikusteko, pausuz-pausu egindako moldaketak aztertuko dira.

Gurpil motordunak baztertu beharrean, kontrolagailu industrialen baten erabilpena proposatu zen. Horretarako, Luis Manuel Martinez-ekin, Iparraldeko gunearen Beckhoff-eko arduradunarekin elkarrizketa bat antolatu zen. Bertan, motor hauentzako konponbide bat proposatu zen. EL – 7411 – 9014|BLDC terminala erabili daiteke, halako izaera duten motorrak kontrolatzeko, ikusi Irudia 68a. Halako txartel batekin, AMR-ak gero eta industrialagoak izango litzateke eta soilik motorrak industrial produktuetatik at geldituko lirateke.

Behin elementu hau laborategian izanda, Irudia 45-an agertzen den entsegu bankua muntatu zen. Baina oraingo honetan, beharrezkoak den hardwarea ezberdina da, non EK – 1100, EL – 7411 – 9014|BLDC eta EL – 2008 erabili diren. Behin entsegu bankua muntatuta Twincat barnean zenbait parametro konfiguratu behar dira. Parametro hauek EL – 7411 – 9014|BLDC-ren erantzuna eraldatzen dute, motorren ezaugarrietara egokitzeko (begiratu Irudia 68b). Informazio honek fabrikatzaileak eskaini behar ditu, baina gurpil motordunen fabrikatzaile txinatarrak, halako informazioa ez zuen ematen.



(a)

8011:0	DRV Motor Settings	RW	> 49 <	
8011:11	Max current	RW	0x00001770 (6000)	mA
8011:12	Rated current	RW	0x00003E8 (1000)	mA
8011:13	Motor pole pairs	RW	0x01 (1)	
8011:16	Torque constant	RW	0x00000032 (50)	mNm/A
8011:18	Rotor moment of inertia	RW	0x00000064 (100)	g cm <sup>2</sup>
8011:19	Winding inductance	RW	0x0064 (100)	0.01 mH
8011:1B	Motor speed limitation	RW	0x000186A0 (100000)	1/min
8011:29	I2T warn level	RW	0x50 (80)	%
8011:2A	I2T error level	RW	0x69 (105)	%
8011:2D	Motor thermal time constant	RW	0x0028 (40)	0.1 s
8011:2E	Rated speed	RW	0x00003E8 (1000)	1/min
8011:2F	Rated voltage	RW	0x0000BB80 (48000)	mV
8011:30	Winding resistance	RW	0x00003E8 (1000)	mOhm
8011:31	Voltage constant	RW	0x0000BB80 (48000)	μV/(1/min)

(b)

Irudia 68 EL – 7411 – 9014 | BLDC: a) Terminalaren egituraren errepresentazioa, b) konfiguratu behar diren parametroak.

Halako egoera ikusita, gurpilari entsegu ezberdin egiteko aukera dago, datu horiek lorrarazteko (ikusi Taula 8).

$$Kt = Ke/0,001 \quad (41)$$

$$Ke = \frac{\text{Volts}}{\text{abiadura (rpm)}} \quad (42)$$

Taula 8. Gurpil Motordunen ezaugarrien laburpena

Ezaugarria	Balioa
Korrante Maximoa (mA)	11471
Korrante Nominala (mA)	10471
Polo Kopurua (m)	16
Indar Parearen konstantea [Kt] $\left(\frac{mNm}{A}\right)$	36346821
Motorraren Voltaje konstantea [Ke] $\left(\frac{\mu V}{min}\right)$	36347
Erresistentzia $m\Omega$	1920

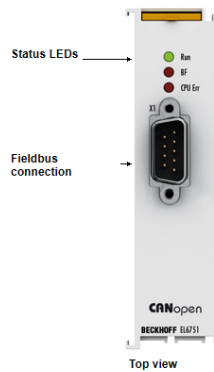
Naiz eta parametroak definitu, gurpilaren mugimendua ez zen egokia. Beraz, gurpil hauen erabiltzearen ideia baztertu zen eta gurpil industrialen bilaketa hasi zen. Merkatuan zenbait aukera ezberdinak aurkitu daitezke, azken finean tesi hau aurkezten den unean edozein plataforma mugikorrek gurpilen beharra daukate. Taula 9-an agertzen diren balioak merkatutik aukeratutako motorrak ikus daitezke. Datu horiek, AMR-ak dituen motor gurpildunekin konparatzen dira.

Gemmotors, motor gurpildunak garatzen dituen enpresa berria da. Fabrikatzaile honek, motor hauen abantailak ikusi dituzte eta teknologia hau enpresetara hurbildu nahi du. Tesi hau aurkezten den unean, gurpil hauen fabrikazioarekin ez dira hasi, beraz etorkizunerako produktu bat da. Bigarren aukera aldiz, Nanotec fabrikatzailearen baliabideak erabiltzea da. Berez, motorren hauen izaera ezberdina da. Pausuz – pausuzko teknologia erabiltzen dute errotorera mugitzeko. Kontrol aldetik, motor sinpleagoak dira eta 3D inprimagailuetan aurkitu daitekeen motorrak dira. BLDC-ekin konparatuz abiadura geldoagoak daramatzate, baina posizionamendu aplikazioetarako zehaztasuna eman dezakete.

Taula 9. Plataforma Mugikorrentzako aukeratutako motorrak

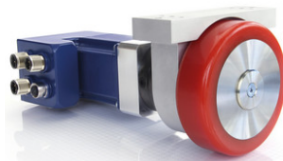
Ezaugarria	Ditugun gurpilak	Gemmotors	Nanotec
Motorraren izaera	BLDC	BLDC	Step by Step
Motorraren abiadura maximoa (rpm)	100-600	100-400	10-3.500 (motorrak bakarrik)
Indar Pare Nominalak (Nm)	10-35	30	12,6
Frenoa/Tentsio beharra	0/Ez	40 Nm/Bai	1,87 Nm/Ez
Behar duten tentsioa (V)	24	24	24
Potentzia (W)	180	700	100
Korrante Nominala (A)	7,5	29,2	4,2
IP babesa	Ez	IP54	IP65
Komunikazioa	Analogikoa	CanOpen	CanOpen

Merkatuaren egoera ikusita eta proiektuak dituen baliabideak ikusita, Nanotec gurpilak aukeratu dira. Eragileaz eraldatzerako orduan, Beckhoff-eko hardwarean eraldaketak planteatu behar dira. Batez ere CanOpen komunikazio Bus-a erabiliko delako, gurpilekin informazioa tartekatzeke. Informazioa Bus batetik igarotzerako orduan, AMR-ren eskema elektrikoa eraldatzen da, egitura sinplifikatuz (ikus Eskema elektrikoa, orria - 137 -). Motorrez gain, CanOpen motako txartel baten beharra dago, Beckhoff eta motorrak komunikatzeko (begiratu Irudia 69).



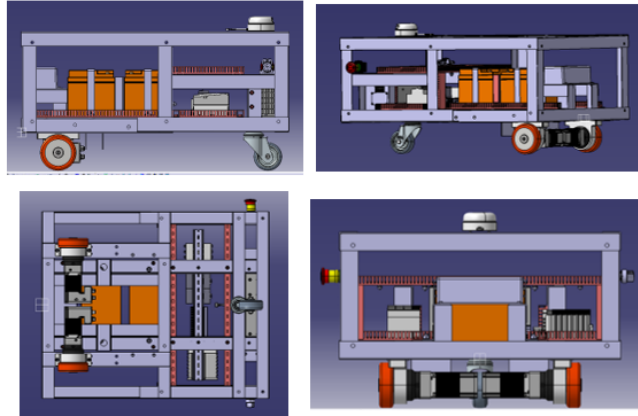
Irudia 69 EL – 6751, Beckhoff-en CanOpen-eko modulua

EL – 6751 moduluak D9 motako konektore bat behar du, gurpilak aldiz M12-ko izaerako konektorea. Komunikazioa sortarazteko, M12-ko kableak D9 batean soldatu behar dira, komunikazio Bus-ak ezartzen dituen arauen arabera. Irudia 70a-an Nanotec saltzen duen gurpilaren errepresentazioa ikus daiteke eta gurpilak AMR-an ezartzeko, haren egitura moldatu behar da. Irudia 70b-n unibertitatearen plataforma mugikorraren errepresentazioa ikus daiteke, non motor hauek ezartzekoan aurrealdeko gurpilaren tamaina eraldatu behar da, Nanotec gurpilek erradioa txikiagoa duelako.



(a)





(b)

Irudia 70 Unibertsitateko AMR-ren motor berriak: a) Nanotec motorren irudia, b) AMR-ren muntaketaren errepresentazioa.

Hardwarearen aldaketaren ondorioz, Twincat programazioa eraldatu behar da, gurpilek abiarazteko komunikazioa eraldatu delako. Motor hauek, Twincat Motion Control barnean konfiguratuta behar dira, hau da, atal konkretu eta bereiz batean programatzen dira, denbora errealean kontrolatzeko. Denbora errealez hitz egiten denean, gurpil inguruko kontrolagailu guztiak 10 ms abiarazten direla esan nahi du. Horrelako exekuzio abiadurekin zehaztasuna lorrarazi daiteke.

Hasierako gurpilekin konparatuz, abiadura aginduak eta kontrolagailua PLC izaerako proiektu batean atxikita zegoen eta AMR-ren beste logikarekin konpartitzen zituen errekurtsioak. Nanotec gurpilekin aldiz, Twincat motorren programazioa eta PLC programazioa bereizten ditu. Gainera, motorrak kontrolatzeko funtzio bereziak ezartzen ditu, bai azelerazioak baita dezelerazioak kontrolatzeko, non motorren jarrera guztiz kontrolatu daitekeen. Beraz, halako aldaketarekin AMR-ren mugimendua era eraginkor eta zehatzago batean abiarazi daiteke. Naiz eta motorren kontrola PLC-etik bereizita egon harren, PLC proiektuan abiadura kontsignak, motorren aktibazio kontsignak, etab. ezar daitezke. Horrek, PLC barnean makinaren egoera ezberdinak definitzera behartzen du (ikusi *Pseudokodea 27* eta *Kodea 28*, or. - 155 -). Hasierako uneetan funtzio ezberdinen irakurketa egiten da, berez funtzio hauek aldagai ezberdinak lista global batean gordetzen dituzte parametroak, horregatik gakotze artean elementurik ez dago. Hamazazpigarren lerroan, makinaren egoera ezberdinak aztertzen dira. Lehengo kasua "*Paro*" du izena, non motorrak tentsio gabeko egoera batean daude eta komunikazioa berriro sortarazi behar den. Behin egoera hau gaindituta, motorrak aktibatzen dira. Horretarako motorren sarrera aktibatzen da, kontrolagailuak martxan jartzeko. Egoera honetatik irteteko hiru gauza gerta daitezke: roomba botoia sakatu dela, teklatu baten bidez AMR - ak mugiarazi nahi dela adierazi zaiola edo geldiarazi programa agindua eman duelako. Akzio bakoitzak bere egoera izango du, baina lehengo biak motorrei abiadura kontsignak ezarriko dituzte eta azkenekoa makina lehengo fasera itzuliko da.

### Pseudokodea 27 Nanotec gurpilentzako Main programa

```
9  Abiarazi Sarrera eta Irteera Digitalen funtzioa
10 Abiarazi Bestelako irakurketak, beteak bete teklatur
11 Abiarazi Segurtasun funtzioa
12 Abiarazi CanOpen-eko irakurketa prozedura
13
14 Case(Egoera):
15   Egoera = 0-> Makina guztiz geldituta Motorrak tentsio gabe
16   If(Start Botoia sakatuz):
17     Motoreen errelea itxi egiten da, Motorrak tentsioa izateko
18   Else
19     Motorrak geldiarazita utzi, errelea irekita utzita
20   End If
21   Egoera = 10-> Start Moduan
22   If(Roomba botoia = ON):
23     Egoera = 20
24     Motorren kontrola aktibatu
25   End If
26   If(Stop botoia sakatuta):
27     Egoera = 0
28   End If
29   If(Erabilerara Manuala ezarriz):
30     Egoera = 30
31   End If
32   Egoera = 20 -> Roomba Modea
33   If(Roba Botoia = OFF):
34     Egoera = 10
35   Else
36     Roomba Funtzioa Abiarazi
37     Motorrei Aginduak bidali
38   End If
39   Egoera = 30 -> Erabilera Manuala(Teklatutik AMR-a mugitu)
40   If(Erabilerara Manuala = OFF):
41     Egoera = 10
42   Else
43     Teklatuaren funtzioa abiarazi
44     Motorrei Aginduak bidali
45   End If
46 End Case
```

Naiz eta gurpilen zehaztasuna hobetu, AMR-entzako gurpilen ikerketa proiektu bat hasi da. Non Centro Stirling, Azaro Fundazioa eta Gasteizko Ingeniaritza Eskola parte hartzen dute. Bertan motorrei buruzko izaerak, kontrolak, etab. ikertzen hasi dira. Proiektua “Motor de Accionamiento para Robot Guiado Automáticamente”, KK-2019/00099, Programa ELKARTEK du izena.

## 1.6 Prototipoaren Posizionamendua

AMR-an lokalizazioa burutzeko, lehenbizi Kodea 22-ak proposatzen dituen funtzioak testu egituratu hizkuntzara pasatu behar dira. Lehengo lokalizazio mota ICP-ren bitartez burutu da, bigarrena aldiz autoren erakarpenetik dator. *Pseudokodea 28*-an (aztertu Kodea 29, or. - 156 -) eskolako AMR-an programaturiko funtzioa ikus daiteke eta *Pseudokodea 15*-an oinarritutako programazioa. *Pseudokodea 28*-an beste bi azpi funtzioen beharra ditu. Bi funtzio hauek iterazio kopuru konkretu bat arte exekututzen dira. Dena den, erroreak balio minimo bat lortzen baldin badu, orduan kiribil horretatik irteten da,  $n_{\max}$  iterazio gabe. *Pseudokodea 28*-an ikusten den moduan, matrize edo bektore baten osagaiak banan-banan betetzen dira. Horrela programazioan sorturiko erroreak modu erraz batean jarraitu daiteke.

### Pseudokodea 28 Iterative Closed Point funtzioa testu egituratuan

```
1 // Aldagaiak abiarazi
2 Errotazio Matrizearen lehengo zutabea eta lehengo errenkada (Ricp11) = 1
3 Errotazio Matrizearen bigarren zutabea eta lehengo errenkada (Ricp12) = 0
4 Errotazio Matrizearen lehengo zutabea eta bigarren errenkada (Ricp21) = 0
5 Errotazio Matrizearen bigarren zutabea eta bigarren errenkada (Ricp22) = 1
6 Translazio bektorearen lehengo errenkada (Ticp1) = 0
7 Translazio bektorearen bigarren errenkada (Ticp2) = 0
8
9 For(ncountn=1-etik nmaxIter arte):
10 FindNearestPoint() funtzioa abiarazi -> Lortu LiDAR-eko bi laginketen puntuen ordena eta
   sorturiko errorea
11 SVDMotionEstimatio() funtzioa abiarazi -> Lortu Errotazio matrizea(R11,R12,R21,R22) eta
   Translazio bektorea(T1,T2)
1 Ricp11 := R11*Ricp11 + R21*Ricp12;
2 Ricp12 := R12*Ricp11 + R22*Ricp12;
3 Ricp21 := R11*Ricp21 + R21*Ricp22;
4 Ricp22 := R12*Ricp21 + R22*Ricp22;
5 Ticp1 := R11*Ticp1 + R12*Ticp2 + T1;
6 Ticp2 := R21*Ticp1 + R22*Ticp2 + T2;
7 Prozeduraren errorea metatu
8 If(Prozeduraren errore metatua < Ezarritako muga):
9   Break
10  End If
11
12 End For
```

fbFindNearestPoint funtzioak *Pseudokodea 29* (aztertu Kodea 30, or. - 157 -) jarraituz programatu da eta *Pseudokodea 16*-an oinarrituta dago. Non bi LiDAR laginketa artean puntuen arteko konkordantzia bilatzen dituen. Testu egituratuan (ST), bektoreekin lan egiten dituzte eta bektore bakoitzaren luzera ezarri behar da. Luzera honek behar baino datu gehiago gordetzen dira, baldintza zabalgoetan lan egiten delako. Hau da, LiDAR-rrak 600 datuko bi bektoretan puntuak gordetzen ditu. Baina tesiak aurkezteko unean, LiDAR-rrak daraman abiadurarekin 360-362 datu gordetzen ditu. LiDAR honek 360º-etan datuak hartzen dituela, sentsoreak motor batek biratzen duelako. LiDAR-raren datu bilketaren zehaztasuna eraldatzeko motorraren abiadura eraldatzen da. Zehaztasun handiko aplikazioetan, motorraren abiadura txikitzen da, datuak gradura gorde beharrean gradu erdian gordetzeko. Beraz irteerako bektorearen tamaina biderkatzen da. MatLab-ek datuekin eraginkorragoa da, bektorearen tamaina finkorik ez duelako. Baina, TwinCat barnean LiDAR-raren funtzionamendu baldintzak ezar behar dira.

### Pseudokodea 29 Bi laginketa arteko korrelazioa bilatu ST hizkuntzarako

```
1 For(ncount=1-etik 500 arte):
2   For(ncounter=1-etik 500 arte):
3     Errorea X(ncounter) = LiDAR-eko data2-tik X(ncounter) datua ken LiDAR-eko data1-tik
   X(ncount) datua
4     Errorea Y(ncounter) = LiDAR-eko data2-tik Y(ncounter) datua ken LiDAR-eko data1-tik
   Y(ncount) datua
5     Distantzia(ncounter) = erro karratua (Errorea X * Errorea X) + (Errorea Y * Errorea Y)
6     If(ezarritako muga > Distantzia(ncounter)):
7       Ezarritako muga = distantzia(ncounter)
8       Index(ncount) = ncounter
9     End If
10  End For
11  Errorea = errorea + Ezarritako muga
12 End For
```

Behin laginketak korrelazioa izanda, SVD-ren bitartez bien arteko translazioa eta errotazioa kalkulatu behar dira, *Pseudokodea 30* erabiliz (ikusi *Pseudokodea 17* eta Kodea 31 or.- 157 -). Berez, MatLab-ek matrizeekin lan egiteko prestatuta dago, baina TwinCat ez. Horregatik SVD funtzioa programa berri batean idatzi behar da. Non MatLab programazio sinbolikoaz baliatuz, SVD barnean erabiltzen diren funtzio eta matrizeak errepikatze aukera eman duen.

### Pseudokodea 30 SVD programazioa ST hizkuntzarako

```
1 For(ncount=1-etik 500 arte):
2   S Matrizean X balioak(n_SX) bildu LiDAR-eko data2-tik index parametroaren arabera
3   S Matrizean Y balioak(n_SY) bildu LiDAR-eko data2-tik index parametroaren arabera
4 End For
5 For(cont=1-etik 500 arte):
6   Batuketen lehengo batukaria(sum1) = LiDAR-eko data1-etik X datuak + sum1
7   Batuketen bigarren batukaria (sum2) = LiDAR-eko data1-etik y datuak + sum2
8   Batuketen hirugarren batukaria(sum3) = S Matrizetik X datuak + sum3
9   Batuketen laugarren batukaria (sum4) = S Matrizetik Y datuak + sum4
10 End For
11   Batuketa batukari bakoitzaren media lortu
12 For(cont=1-etik 500 arte):
13   Arr_SshiftX bektorea bete n_SX ken sum3-ren mediaren balioa aplikatuz
14   Arr_SshiftY bektorea bete n_SY ken sum4-ren mediaren balioa aplikatuz
15   Arr_MshiftedX bektorea bete Data1-etik X datuak ken sum1-ren mediaren balioa aplikatuz
16   Arr_MshiftedY bektorea bete Data1-etik Y datuak ken sum2-ren mediaren balioa aplikatuz
17
18   W matrizearen lehengo zutabearen eta lehengo errenkadaren balioa (W11) kalkulatu
19   W matrizearen bigarren zutabearen eta lehengo errenkadaren balioa (W12) kalkulatu
20   W matrizearen lehengo zutabearen eta bigarren errenkadaren balioa (W21) kalkulatu
21   W matrizearen bigarren zutabearen eta bigarren errenkadaren balioa (W22) kalkulatu
22 End For
23
24   C matrizearen lehengo zutabea eta lehengo errenkadaren balioa (C11) kalkulatu
25   C matrizearen bigarren zutabea eta lehengo errenkadaren balioa (C12) kalkulatu
26   C matrizearen lehengo zutabea eta bigarren errenkadaren balioa (C21) kalkulatu
27   C matrizearen bigarren zutabea eta bigarren errenkadaren balioa (C22) kalkulatu
28
29   V matrizearen kalkulua burutu baino lehen X1 balioa kalkulatu
30   V matrizearen kalkulua burutu baino lehen X2 balioa kalkulatu
31
32   If(X1 edo X2 balioak nuluak baldin badira):
33     V matrizearen lehengo zutabea eta lehengo errenkada (V11) = 0
34     V matrizearen lehengo zutabea eta bigarren errenkada (V12) = 1
35     V matrizearen bigarren zutabea eta lehengo errenkada (V21) = 1
36     V matrizearen bigarren zutabea eta bigarren errenkada (V22) = 0
37
38     S matrizearen lehengo zutabea eta lehengo errenkada = X1-en erro karratua
39     S matrizearen lehengo zutabea eta bigarren errenkada = 0
40     S matrizearen bigarren zutabea eta lehengo errenkada = 0
41     S matrizearen bigarren zutabea eta bigarren errenkada = X2-en erro karratua
42
43     U matrizearen lehengo zutabea eta lehengo errenkada (U11) = V11
44     U matrizearen lehengo zutabea eta bigarren errenkada (U12) = V12
45     U matrizearen bigarren zutabea eta lehengo errenkada (U21) = V21
46     U matrizearen bigarren zutabea eta bigarren errenkada (U22) = V22
47
48     Errotazio eta Translazio matrizeak kalkulatu, matrize bakoitzaren elementu bakoitza
    bektoreetan antolatuta dagoela kontuan hartuz.
49
50   Else:
51     V, S, U matrizeen kalkulua eragin kontuan izanda matrizearen konposizioa bektoreen bidez
    egin dela
52   End If
```

MatLab kode guztiak ST-ra bilakatu dira, TwinCat PLC barnean programatzeko, bertan egingizun bakoitzari lan denbora bat ezar daitekeelako. Horrek, programazioaren kontrola ematen du eta unibertsitateko AMR-ren kasuan erabiltzen duen IPC-a dela eta, denboraren kontrola ezinbestekoa da, haren baliabideak era eraginkor batean erabiltzeko. Beraz, programa bakoitzaren denbora kontrola ezarriz, CPU-aren kalkuluen prozesamenduak kontrolatzen dira, non kalkulu astunei denbora gehiago emanez, prozesadorea ez da beteko kalkulu horiekin.

Autoreen argitalpenetan erregresio lineala bitartez lokalizazio erlatibo bat burutu da (ikus *Pseudokodea 18* eta Kodea 20 or. - 149 -). Bertan MatLab erabiliz, zenbait simulazio burutu ziren eta haren eraginkortasuna ikustekoan, ST hizkuntzara bilakatu dira. Horrela, AMR-ak eremu errealean lokalizatzeko bi funtzio ezberdinekin froga daiteke eta *Pseudokodea 31*-ean oinarrituta dago (begiratu Kodea 32 or. - 159 -).

### Pseudokodea 31 Erregresio linealaren funtzio nagusia ST hizkuntzarako

```
1 Abiarazi datuak prestatzen duen PreparaDatos funtzioa
2 Abiarazi Phi eta Phi iraulia biderkatzen duen funtzioa
3 Exekutatu matrizearen alderantzizkoa egiten duen funtzioa
4 Biderkatu matrizearen alderantzizkoa eta Phi iraulia
5 Lortu biraketa eta translazio emaitzak
```

Kodea 32-ak Kodea 31 funtzioa ordezkatzeko du. ST erregresioan bost azpi funtzio planteatzen dira. Laugarren lerroan agertzen dena, erregresio lineala egin baino lehen datuen prestaketa planteatzen du. Hemeretzigarren lerroko funtzioak matrizearen iraulia kalkulatu du eta bi matrizeen biderketa planteatzen ditu. Berrogeita bosgarren lerroan dagoen funtzioak, matrizearen alderantzizkoa kalkulatu du. Hirurogeita zazpigarren lerroan, matrizearen alderantzizkoa eta irauliaren arteko biderketa egiten du. Azkenik laurogeita batgarren lerroan emaitzak adierazten ditu. Kodea 32-a abiarazi baino lehen, LiDAR-rak ematen dituen datuak koordinatu kartesiarretara pasatu behar dira.

*Pseudokodea 32 Datuen prestaketa funtzioa ST hizkuntzarako*

```

1  gg = 1
2  For(q=1-etik 600 datu arte):
3    Y matrizearen lehengo datuak bete gg zenbatzailea arabera eta Data2-tik Y balioak (Yact)
   erabiliz
4    Phi matrizearen lehengo zutabea (PhiC1[gg]) Data1-tik -Y balioarekin (-Yant[q]) bete
5    Phi matrizearen bigarren zutabea (PhiC2[gg]) Data1-tik X balioarekin (Xant[q]) bete
6    Phi matrizearen hirugarren zutabea (PhiC3[gg]) bat balioarekin bete
7    Phi matrizearen hirugarren zutabea (PhiC3[gg]) zero balioarekin bete
8
9    Phiren iraulia egiteko 4 bektore prestatu zutabeak errenkada bezala kontsideratuz, adib.
   PhiF1 = PhiC1
10   gg = gg + 1
11   End For
12
13   For(q=1-etik 600 datu arte):
14     Y matrizearen lehengo datuak bete gg zenbatzailea arabera eta Data2-tik X balioak (Xact)
   erabiliz
15     Phi matrizearen lehengo zutabea (PhiC1[gg]) Data1-tik X balioarekin (Xant[q]) bete
16     Phi matrizearen bigarren zutabea (PhiC2[gg]) Data1-tik Y balioarekin (Yant[q]) bete
17     Phi matrizearen hirugarren zutabea (PhiC3[gg]) zero balioarekin bete
18     Phi matrizearen hirugarren zutabea (PhiC3[gg]) bat balioarekin bete
19     gg = gg + 1
20   End For

```

*Pseudokodea 32-n* (aztertu Kodea 33 or.- 159 -) erregresio lineala burutzeko behar diren eragiketak guztiak aurkeztu dira. Matrizearen irauliaren operazioa erraz identifikatu daiteke, berez zutabeak lerroetan bilakatzen direlako (hamahirugarren lerroetik hamaseigarren arte). Kode honetan burutzen diren beste operazioak autorearen argitalpenean azaldutako ekuazioen ondorioz datoz.

*Pseudokodea 33 Bi matrize arteko biderketa ST hizkuntzarako*

```

1  Bi 4 x 4-ko matrize biderketa egiteko 16 for planteatzen dira hurrengo kodifikazioa erabiliz:
2  For(i=1-etik 1200-ra):
3    Batukaria = PhiF** * PhiC* + Batukaria -> Non * zutabe eta ** errenka konkretu bati
   egokitzen zaion. 1 <= * <= 4 eta 1 <= ** <= 4
4    End For
5    Emaitzaren matrizea bektoreetan gorde * eta ** arabera (F**C*)

```

*Pseudokodea 33-an* (ikus Kodea 34 or.- 159 -) matrize arteko biderketa nola burutu behar den adierazten da. TwinCat matrize arteko biderketak ez ditu burutzen. Gainera, bi bektoreen arteko biderketak burutzeko balioz balio biderkatu behar dira. Horren ondorioz, 1200 x 4 matrize bat beste 4 x 1200 matrize batekin biderkatzerako, hamasei "for" eragiketa behar dira. Azken finean, funtzio honek eskuzko eragiketak eginko balitz programatuta dago.

*Pseudokodea 34 Matrizearen alderantzizkoaren kalkulua ST hizkuntzarako*

```

1  4 x 4-ko matrize iraulia prestatzeko, 16 ekuazio planteatzen dira
2  Ekuazio bakoitzak, alderantzizko matrize bat lortzeko F**C* arteko biderketak, kenketak,
   batuketak eta zatiketa proposatzen dira.
3  Adibidez: Alderantzizko matrizearen lehengo zutabe eta lehengo erronkaren balioa lortzeko
   halako biderketa bat egin behar da.
4  InvF1[1]:= (F2C2*F3C3*F4C4 - F2C2*F3C4*F4C3 - F2C3*F3C2*F4C4 + F2C3*F3C4*F4C2 + F2C4*F3C2*F4C3
   - F2C4*F3C3*F4C2)/(F1C1*F2C2*F3C3*F4C4 - F1C1*F2C2*F3C4*F4C3 - F1C1*F2C3*F3C2*F4C4 +
   F1C1*F2C3*F3C4*F4C2 + F1C1*F2C4*F3C2*F4C3 - F1C1*F2C4*F3C3*F4C2 - F1C2*F2C1*F3C3*F4C4 +
   F1C2*F2C1*F3C4*F4C3 + F1C2*F2C3*F3C1*F4C4 - F1C2*F2C3*F3C4*F4C1 - F1C2*F2C4*F3C1*F4C3 +
   F1C2*F2C4*F3C3*F4C1 + F1C3*F2C1*F3C2*F4C4 - F1C3*F2C1*F3C4*F4C2 - F1C3*F2C2*F3C1*F4C4 +
   F1C3*F2C2*F3C4*F4C1 + F1C3*F2C4*F3C1*F4C2 - F1C3*F2C4*F3C2*F4C1 - F1C4*F2C1*F3C2*F4C3 +
   F1C4*F2C1*F3C3*F4C2 + F1C4*F2C2*F3C1*F4C3 - F1C4*F2C2*F3C3*F4C1 - F1C4*F2C3*F3C1*F4C2 +
   F1C4*F2C3*F3C2*F4C1);
5  Non F1-> lehengo errenkada adierazten duen eta [1] lehengo zutabea.

```

*Pseudokodea 33-ak* (aztertu Kodea 35 or.- 161 -) matrizearen alderantzizkoa kalkulatu du. Bertan ikus daitezkeen moduan eragiketak oso luzeak dira eta termino askoren biderketak eta zatiketak daude. MatLab-eko programazio sinbolikoari ezker, funtzio honen programazioa sinplifikatu zen.

Kodea 1 Alderantzizko matrizearen eta matrize irauliaren arteko biderketa ST hizkuntzan

```

1 FOR i:=1 TO 1200 DO
2   D1Filas[i]:= InvF1[1]*PhiF1[i]+InvF1[2]*PhiF2[i]+InvF1[3]*PhiF3[i]+InvF1[4]*PhiF4[i];
3   D2Filas[i]:= InvF2[1]*PhiF1[i]+InvF2[2]*PhiF2[i]+InvF2[3]*PhiF3[i]+InvF2[4]*PhiF4[i];
4   D3Filas[i]:= InvF3[1]*PhiF1[i]+InvF3[2]*PhiF2[i]+InvF3[3]*PhiF3[i]+InvF3[4]*PhiF4[i];
5   D3Filas[i]:= InvF4[1]*PhiF1[i]+InvF4[2]*PhiF2[i]+InvF4[3]*PhiF3[i]+InvF4[4]*PhiF4[i];
6 END FOR

```

Kodea 2 Datuen errepresentazioa ST hizkuntzan

```

1 sinu := 0;
2 koseno := 0;
3 tx := 0;
4 ty := 0;
5
6 FOR i:=1 TO 1200 DO
7   sinu := D1Filas[i]*Y[i]+sinu;
8   koseno := D2Filas[i]*Y[i]+sinu;
9   tx := D3Filas[i]*Y[i]+tx;
10  ty := D4Filas[i]*Y[i]+ty;
11 END FOR

```

Kodea 1-an eta Kodea 2-an azkeneko operazioak biltzen dituzte, non emaitzak AMR-ren translazioa eta biraketa kalkulatu diren. Puntu honetan maparen gordetzea eta irakurketa nola egin behar den planteatu behar da.

## 1.7 Ondorioak eta hurrengo pausuak

Lorturiko urratzek ezin bestekoak dira, AMR-ren lokalizazioa burutzeko. Simulazioak baliaigarriak izan dira inteligentzia garatzeko eta AMR-ak dituen mugak ikusarazteko. LiDAR -raren kasuan, korridore luze eta simetrikotan plataforma mugikorak era eraginkor batean ezin da lokalizatu. Bi algoritmo planteatu dira lokalizaziorako, bata iragazki partikuletan oinarrituta eta bestea sare neuronalekin oinarrituta. Lehengo sistematikak bi prozedura ezberdin frogatu egin dira. Lehengoak ICP teknika erabiltzen du, SVD matrizeak erabiliz. Bigarrenak aldiz, erregresio lineala planteatzen du, SVD teknika ordezkatzeko. Bi kasuak TwinCat barnean programaturik daude, AMR-ak eremu errealean frogatzeko. Horrekin bat, AMR baten garapena landu da, ekipamendu industrialia erabiliz. Bestalde, plataforma mugiarazteko Roomba algoritmoa eta segurtasun funtzioak garatu dira, denbora errealean abiaraziz (10 ms-ro programa abiarazi egiten da). Horretako, LiDAR ez industrialia C++ eta ADS komunikazioa bitartez IPC barnean integratu da, informazioa TwinCat barnean izateko.

Eremu errealean posizionamenduen frogapenak ez dira egin. Berez, halako eragiketa egiteko bi oztupo gainditu behar direlako. Alde batetik, lan eremuaren maparen eraikitzea ez dagoelako planteatuta, batez be maparen gordetze prozedura nola egin behar den ez delako burutu. Bestetik, behin mapa gordeta izanda, TwinCat informazio hori nola ireki eta erabili behar duen ez delako egin. Tesi honek lokalizazio eta nabigazio algoritmoetan oinarritu egin da, horregatik hurrengo pausuak maparen integrazioa burutzea izango litzateke.

Badaude ideia batzuk lehengo pausuak burutzeko. MatLab eta TwinCat ADS bitartez komunikatu daitezke. Protokolo honek informazioa tartekatze aukera ematen du, bai MatLab AMR-ren IPC-an integratuta baldin bada, baita Ethernet bitartez beste ordenagailu batean MatLab abiarazten baldin bada. Unibertsitatearen plataformaren IPC-a ezagututa, lehengo aukera baztertu egin behar da, azken finean nukleo bakarreko ordenagailua delako eta haren kapazitatetik, TwinCat % 80 erabiltzen duelako.

Tesi honetan aurkezteko eremuan, inteligentzia guztian AMR-ren IPC-an egoteko planteatu da. Horrekin bat, programa guztiak TwinCat barnean programatuta egotea edota ADS-ren bitartez eta C++ erabilita, behar besteko programazioak burutzea. Adibidez:

LiDAR - raren informazioaren irakurketa. Oinarri hau erabiliz, C++ erabiliz programa bat garatu daiteke, non mapa irudi batean gordeta egonda, programak ADS-ren bitartez TwinCat-era bektore moduan informazioa bidali dezakeen. Gainera, logika hau erabilita, mapa eraikitzerako orduan, TwinCat LiDAR-raren eta gurpilaren informazioa bektore moduan, C++-era bidali daiteke eta bertan mapa eraiki daiteke. Behin mapa burututa izanda, irudi batean bilakatu daiteke gordetzeko.

Pausu hauek AMR-ren izaera industrialia mantentzen dute, non informazioa prozesatzen duen unitateak zentralak eremu esanguratzat prestatuta dagoen. Baina halako ekipamenduak zenbait mugaketa ekartzen ditu, adibidez irudien prozesamendurako erabiltzen diren sare neuronalen exekuzioekin. Hauek Nvidia-ko GPU-etan abiarazteko prestatuta daudelako. Halako teknologia erabiltzeko eta proiektu berriak sortzeko asmoz, tesi hau aurkezten den unean Nvidia Jertson AGX baten ezarpena Pioneer 3 robot batean egiten hari da (ikusi Irudia 71). Pioneer 3-ak 1995-garren urteko robota da eta eskolako helburua berriztea da. Horretarako Jertsona erabiliko da, non 64 bit eta 8 nukleoko ARM prozesadorea daukan eta 512 nukleoko Nvidia GPU - a atxiki dauka.



(a)



(b)



(c)

Irudia 71 Unibertsitateak planteatuko dituen beste plataforma mugikorrek sare neuronalak erabiltzeko a) Nvidia Jertson AGX, b) Pioneer 3 robot mugikorra, c) Garatutako moldaketak

Pioneer proiektua hasi berria dago, non teknologia zaharra baztertu da eta ekipamendu berriarentzako euskarriak prestatu diren (ikusi Irudia 71c). Proiektu paralelo honen helburua gaur egungo zientzia landu nahi da, industriako ekipamenduak moldatzeko. Horrela urte batzuk geroago GPU-ak industria barneko teknologia izango dira.

# VII.

## Atalburua

Hain urrun heltzen diren pertsonak, non okertzen ez diren, hauek lana albo batera utzi dute -Max Planck-

### Ekarpenak, ondorioak eta hurrengo urratsak

---

---

Atal honetan Tesi hau ikasitakoa eta ikertutakoa laburbilduko da. Bertatik ateratako zenbait ondorioak Autorearen erakarpenak bilakatu dira, hauek barne nabigaziorako hobekuntzak erakarrit. Bestalde, eskolako AMR-ren hurrengo pausuak ezartzen dira, kontuan izanda ikerketa bi adarretan banandu dela. Alde batetik sare neuronal konboluzionaletan oinarritutako bidea eta bestetik AMR-ak duen LiDAR-raz baliatuz SLAM bat burutzea.

#### 1 Autorearen ekarpenak

---

Tesiarekin amaitzeko, autorearen argitalpenak aurkeztuko dira. Hauek AMR-ren garapenarentzako oinarrituta daude bati bat, bereziki unibertsitatean plataforma mugikor baten prototipoa burutzeko asmoa dagoelako. Argitalpen hauek bi multzotan bereiziko dira, non lehengo multzoan lehengo autore bezala burututako argitalpenak aipatuko diren eta bigarreanean beste autore batzuekin egindako kolaborazioak aztertuko diren.

Lehengo autore bezala hiru artikulu eta kongresu bat burutu dira. Multzo honen lehengo argitalpena 2019-ko otsailaren batean eman zen, *“A Free Navigation of an AGV to a Non-Static Target with Obstacle Avoidance”* [137] izenarekin. Bertan AMR baten gidaketa egiteko algoritmo bat proposatzen da. LiDAR-aren informazioa erabiliz eta eragiketa matematiko sinpleak burutuz (gehiketak, biderketak, etab.), AMR-ak puntu mugikor bat jarraitu dezake eremuan aurkitutako objektuak saihestuz.

Bigarrena 2019-ko abuztuaren 26-an argitaratu zen, *“Predictive Dynamic Window Approach Development with Artificial Neural Fuzzy Inference Improvement”* [138] tituluarekin. Garapen hau DWA hobetzeko helbururekin etorri zen. Berez, DWA 1997-garren urtean burututako teknika bat da eta honek eremu ezberdinetik abiarazteko parametro finko batzuk hautatu behar dira. Parametroak era dinamikoak moldatzeko aukera dago. Horretarako ANFIS motatako sare neuronalaren erabilpenekin, parametro horiek moldatzen dira. ANFIS-ek lehengo etapan maparen izaera ikasten du, balio optimoak ikasiz. Behin ikasita, DWA behar dituen parametro horiek, lan eremuaren egoeraren arabera aldatzen ditu. Ikerketa bertan behera utzi beharrean, sistema prediktibo bat atxiki zitzaion etorkizunean zer gerta lekioken aurreikusteko eta horren ondorioz erabaki ezberdinak hautatu ahal ditu algoritmoak

Hirugarrena *“Semantic Segmentation to Develop an Indoor Navigation System for an Autonomous Mobile Robot”* [139] 2020 maiatzaren 25-an argitaratu izan da. Sare neuronalak erabiliz, segmentazioa eragiten duen sare bat prestatu da korridore baten nabigazio lurzorua identifikatzeko. Behi eremu hori identifikatuta, geratzen zen espazio libretik AMR-ak abiarazten da. Nabigazio algoritmo baten lehengo pausuak dira. Horrekin bat GPU-aren abantailak ikusarazi ziren, eta sare neuronalen CPU-etan abiarazita zeukaten errendimenduak ere. Artikulu honek, nabigazio inguruko arloan beste ikuspuntu ezberdinak erakarri ditu. Bati bat, LiDAR-rak sortzen



dituen mugapenak ikusmen artifizialean desagertzen direla. Seigarren atalburuan 1.3.7 atalean adierazi den moduan, LiDAR-ak korridore simetriko eta luzeetan zailtasunak topa dezake, puntu esanguratsuak identifikatzen ez dituelako. Kanpo nabigazioan egiten den moduan, barne nabigazioan segmentazioa, objektuen identifikapena, etab. sare neuronal bitartez buru daiteke, nabigazio espazioa kalkulatzeko, objektuak ekiditeko, etab. Horregatik ikerketa adar bat, GPU erabiltzen duten dispositiboetara eraman da, inteligentzia artifizialak edozein nabigaziorako malgutasuna dakarrelako.

Multzo honetan azkeneko lana kongresu batetik dator, konkretuki SOCO 2020. Bertan *“A relative positioning development for an Autonomous Mobile Robot with a linear regression technique”* [140] lana aurkeztu da. Argitalpen horretan, ICP algoritmoak SVD matrizeak erabili beharrean, erregresio lineala planteatzen da, bi laginketa artean desplazamendua eta biraketa kalkulatzeko.

Bigarren multzoan ordea egindako kolaborazioak aurkezten dira. Horien artean, *“Performance enhancement of the artificial neural network-based reinforcement learning for wind turbine yaw control”*[141], bertan haize turbinen yaw angelua kontrolatzeko estrategia berria planteatzen da, *“reinforcement learning”* teknika planteatuz. Kontua da, haize errotaren sortzeko potentzia handitu behar da, paletan sortzen diren kargak murriztuz.

Bestetik, *“Parametric Study of a Gurney Flap Implementation in a DU91W(2)250 Airfoil”* [142] aurkitzen da, non ikerketa parametrikoki eginez, haize errotaren *“Gurney flaps”*-en (Gfs) luzera optimoa aurkitzeko baliagarria da. Horretarako, eraso angelu positibo duten profila erabili dira eta ikerketa hoztean, zein Gfs balio behar den angelu ezberdinetarako aurkezten da.

*“Modeling of Motorized Orthosis Control”*[143]-an dispositibo ortopedikoen kontrola eta mugimendua hobetzeko teknika proposatzen da. Eboluzio diferentziala aplikatzen duen sistema ahaltsu batek, zenbait parametroei balio optimoak kalkulatu ditu. Erabiltzailearen esfortzua murrizteko minimizazio funtzio batek dispositiboa kontrolatzen du.

Beste aldetik *“ANN-Based Stop Criteria for a Genetic Algorithm Applied to Air Impingement Design”*[144] aurkezten da, non sare neuronalak erabiliz algoritmo genetikoen gelditze prozesu bezala erabiltzen da. Horrek, optimizazioen algoritmoen iterazio kopuruak murrizteko aukera ematen du. Frogapenari bi test egiten zaizkie. Alde batetik zortzi optimizazio funtzioekin balioztatzen da. Ondoren prozesu konkretu baten aurrean frogatzen da.

*“Flow control based 5 MW wind turbine enhanced energy production for hydrogen generation cost reduction”*[145], non eragilearen analisi bat planteatzen da zurrunbilo sortzaileentzako eta ohetxo aleroientzako biak ardatz horizontaleko haize errotetan ezarrita. Horretarako fluxuaren kontrola eginez, potentziaren ebaluazioa egin daiteke, ikusiz errendimenduaren iraganeko datuak baliatuz haize errotaren errendimenduaren hobekuntza.

Haize errotekin jarraituz, *“Kharitonov theorem based Robust Stability Analysis of a Wind Turbine Pitch Control System”* [146] argitalpena dago. Bertan Kharitonov egonkortasun metodologia erabiliz, haize errota baten inklinazio kontrolaren egonkortasuna aztertzen da. Hau da, inklinazio kontrolagailuaren erantzuna aztertu nahi da, funtzionamenduan sorturiko ziurgabetasunen aurrean. Erabilitako modeloaren mugapen ondorioz, garatutako metodologia zehaztasun galera izan dezake aldakortasun handiagoetan.

Taula 11. Autorearen ekarpenak guztiak

Titulua	Aldizkaria	Erreferentziak
A Free Navigation of an AGV to a Non-Static Target with Obstacle Avoidance	MDPI Electronics (Q3)	[137]
Predictive Dynamic Window Approach Development with Artificial Neural Fuzzy Inference Improvement	MDPI Electronics (Q3)	[138]
Semantic Segmentation to Develop an Indoor Navigation System for an Autonomous Mobile Robot	MDPI Mathematics (Q1)	[139]
Performance enhancement of the artificial neural network–based reinforcement learning for wind turbine yaw control	Wind Energy (Q2)	[141]
Parametric Study of a Gurney Flap Implementation in a DU91W(2)250 Airfoil	MDPI Energies (Q3)	[142]
Modeling of Motorized Orthosis Control	Applied Science (Q2)	[143]
ANN-Based Stop Criteria for a Genetic Algorithm Applied to Air Impingement Design	MDPI Energies (Q3)	[144]
Flow control based 5 MW wind turbine enhanced energy production for hydrogen generation cost reduction	International Journal of Hydrogen Energy (Q1)	[145]
Kharitonov theorem based Robust Stability Analysis of a Wind Turbine Pitch Control System	MDPI Mathematics (Q1)	[146]

## 2 Ondorioak eta hurrengo urratsak

Tituluak adierazten duen moduan, tesi osoan zehar jasotako ondorioak bilduko dira, baita ze nolako hurrengo urratsak egin daitezke ikerketarekin jarraitzeko.

Nabigazio autonomoa dela eta, kanpo nabigazioa garatuago dagoela esan daiteke. Berez, automobilgintzako sektoreek diru inbertsioa egin dutelako. Barne nabigazioan aldiz, parametro asko definitu behar direla ikusten da eta industrian zalantza asko daudela antzemateen da. Azken finean, fabrikatzaile bakoitzak bere sistema proposatzen duelako. Irakurritako erakarpenez, kanpo nabigazioa eragiten diren algoritmoak, barneko sistementzako erabili daitezke. Baina barnean elementu arautuak ez daudenez, nabigazioa konplikatu daiteke. Elementu arautu horien artean, trafikoaren seinaleak, errepideko lerroak, etab. Dena den, plataforma barne instalazioetan mugiarazten baldin bada, klimak sartzen dituen aldagaiekin jokatzeko ez du.

Argi eta garbi SLAM-aren inplementazioak egin behar dira, baina ikertzaileek esandakoaren arabera, teknologia halako algoritmoak denbora errealean abiarazteko ez daude prestatuta. Berez, maparen etengabeko berritze prozesuak, hardwarearen baliabide asko behar ditu, horregatik aplikazio gehienetan lokalizazioa soilik burutzen da. Horrela maparen berritzapena erabiltzailearen arduraren geratzen da. Egindako frogapenaren ondorioz, iragazki partikulen lokalizazioak oso ondo funtzionatzen du, baina LiDAR-rak neur dezakeen distantzia maximoaren menpe gelditzen da. Eskolan buruturiko AMR-aren kasuan 40 metrotako korridorean mugiaraztea eta lokalizazioa burutzea jardura astuna izango lirakeke, bi hormak ia simetrikoak direlako eta puntu esanguratsuak ez dituztelako.

Eskolako plataforma mugikorrek lehengo pausuak eman ditu soilik. Naiz eta mugiarazteko sistema autonomo bat izan (Roomba Mode), bere burua lokalizatzeko zenbait programa falta dira. Horien artean maparen eraikitzea eta mapa horren informazioaren erabilpena. Dena den, halako programak burutu bezain laster, bi lokalizazio sistema prestatuta daude frogatzeko. Programazio hau testu egituratua programaturik daude TwinCat barnean, eta horrek programaren exekuzio denboraren kontrola ematen du. Etorkizunari begira eta MatLab-eko programak abiarazita, 10 ms-ko exekuzioak ezin izango dira egin. Azken finean lokalizazioa prozedura astuna delako, optimizazio funtzioengatik.

Horrekin bat, Unibertsitateko AMR-ak behar dituen Nanotec gurpilei ezker, abiadura ondo estimatu daiteke, mapa bat eraikitzeko arazo bat konponduz. Behi C++ bitartez datuak gordetzeko funtzioa burutu ostean, eskolako mapa egin daiteke era egoki batean. Momentu honetan burututako mapa, AMR-a erabili gabe osatu delako.

Mercedez – Benz barnean dagoen prototipoen ondorioak ezin dira aipatu, haien ikerketa babesteko. Dena den, dokumentu honetan aurkeztutako informazioaren arabera, ikusmen artifiziala baliagarria da, muntaketa lerroan dagoen abiadura jarraitzeko. Haren zehaztasuna eta kodeak ez dira aipatuko, esandako arrazoiagatik.

Eman behar diren hurrengo urratsak Gasteizko Ingeniaritza Eskolako prototipoarentzat izango dira. Azken finean, beste pertsona batek ikerketa jarraitzeko aukera ematen duelako. Hain bat alditan esan den moduan, lokalizazioarekin amaitzeko, mapa inguruko kontuak amaitu behar dira. LiDAR-rak emandako esperientziagatik segurtasun elementu bezala erabiltzea egokia izango lirateke. Horren ondorioz, ikusmen artifiziala oinarrituriko nabigazio sistema bat burutzea ezin besteko izango lirateke. Horretarako, sare neuronalen erabilpena aproposena izango litzateke. Horrek hardware aldaketak ekarriko ditu, baina AMR-ak seguruagoak, malguagoak eta aldakortasun haurrean efizienteagoak izango dira. Inteligentzia Artifizialak ematen dituen baliabideak neurgaitzak direlako.

Dena den, teknologiak AI aplikatzeko plataformak garatzen hasi da eta hemendik urte batzuetara, industria mailan GPU duten dispositiboak ikusiko dira. Horren ondorioz, Unibertsitateko AMR-ak haren izaera mantendu dezake, naiz eta sare neuronalak erabiliz.

-FIN-

Eskerrik asko irakurtzeagatik



**Mercedes-Benz**

FÁBRICA VITORIA

eman ta zabal zazu



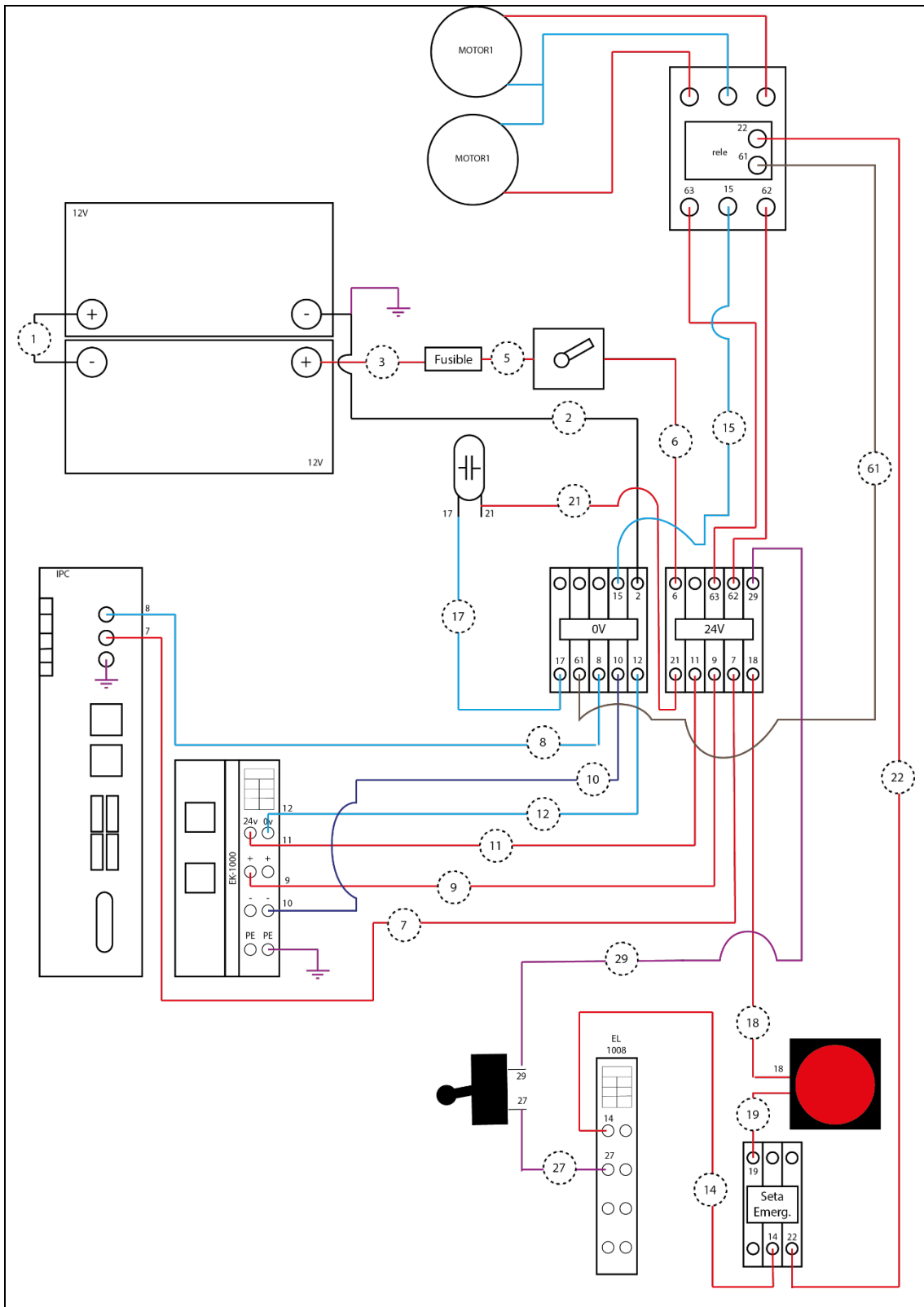
Universidad del País Vasco    Euskal Herriko Unibertsitatea

VITORIA-GASTEIZKO  
INGENIARITZA  
ESKOLA  
ESCUELA  
DE INGENIERÍA  
DE VITORIA-GASTEIZ



# Eranskinak

## 1 Eskema elektrikoa



## 2 Tesi honetan burututako Kodeak

---

### Kodea 3 MatLab eta LiDAR arteko komunikazioa sortzeko

---

```
1 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2 //////////////////////////////////////////////////////////////////// CREAM LA COMUNICACI"ON CON EL LIDAR ////////////////////////////////////////////////////////////////////
3 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
4
5
6 // Incluir las librerías necesarias
7 #include "mex.h" // Librería de MatLab
8 #include "include/rplidar.h" // Librería del LIDAR
9 #include <Windows.h>
10 #include "matrix.h"
11 #include <windows.h>
12 #include <conio.h>
13 #include <stdio.h>
14 #include <string.h>
15
16 #define delay(x) ::Sleep(x) // Defino una variable para generar una pausa
17
18 // Declaro que elemento utilizo de la estructura
19 using namespace rp::standalone::rplidar;
20 //////////////////////////////////////////////////////////////////// COMIENZO CON LA FUNCION ////////////////////////////////////////////////////////////////////
21 void mexFunction(int nlhs, mxArray *plhs[], int nrhs, const mxArray *prhs[]) {
22     /*
23     nlhs Number of expected output mxArray
24     plhs Array of pointers to the expected output mxArray
25     nrhs Number of input mxArray
26     prhs Array of pointers to the input mx Array
27     */
28
29     //////////////////////////////////////////////////////////////////// Genero los parámetros de la comunicación ////////////////////////////////////////////////////////////////////
30     // Puerto de comunicaciones del LIDAR
31     const char * opt_com_path = "com3";
32     // Baudios de comunicacion
33     _u32 opt_com_baudrate = 115200;
34     // Creo el puntero que me permite realizar diversas funciones con el LIDAR
35     RPlidarDriver * drv = RPlidarDriver::CreateDriver(0x0); // 0x0 -> Serial Communication
36     // Creo la conexión y me aseguro de que no tiene errores 1
37     if (IS_FAIL(drv->connect(opt_com_path, opt_com_baudrate)){
38         printf("ERROR en la conexión \n");
39     }
40     else{
41         printf("CONECTED \n");
42         drv->startMotor();
43         printf("start\n");
44         drv->startScan(0,1);
45         // Como todo esta correcto extraigo el puntero
46         long long *ip;
47         union {long long theinteger; void *thepointer;} ivp;
48         ivp.theinteger = 0;
49         ivp.thepointer = drv; // your pointer
50         plhs[0] = mxCreateNumericMatrix(1,1,mxINT64_CLASS,mxREAL);
51         ip = (long long *) mxGetData(plhs[0]);
52         *ip = ivp.theinteger;
53     }
54 }
```

---

### Kodea 4 MatLab eta LiDAR artean informazioa tartekatzeko

---

```
1 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2 //////////////////////////////////////////////////////////////////// LEE DATOS CON EL LIDAR ////////////////////////////////////////////////////////////////////
3 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
4
5
6 // Incluir las librerías necesarias
7 #include "mex.h" // Librería de MatLab
8 #include "include/rplidar.h" // Librería del LIDAR
9 #include <Windows.h>
10 #include "matrix.h"
11 #include <windows.h>
12 #include <conio.h>
13 #include <stdio.h>
14 #include <string.h>
15
16 using namespace rp::standalone::rplidar;
17 RPlidarDriver * drv;
18 void mexFunction(int nlhs, mxArray *plhs[], int nrhs, const mxArray *prhs[]) {
19     /*
20     nlhs Number of expected output mxArray
21     plhs Array of pointers to the expected output mxArray
22     nrhs Number of input mxArray
23     prhs Array of pointers to the input mx Array
24     */
25     void *vp;
26     long long *ip;
27     union {long long theinteger; void *thepointer;} ivp;
28     ip = (long long *) mxGetData(prhs[0]);
```

---

```

29     ivp.theinteger = *ip;
30     drv = (RPLidarDriver*)ivp.thepointer; // your pointer
31     ///////////////////////////////////////////////////////////////////
32     /// UNA VEZ Importado el puntero genero mi programa ///////////////////////////////////////////////////////////////////
33     ///////////////////////////////////////////////////////////////////
34     double *PunteroMatriz_Distancia;
35     double *PunteroMatriz_Posici n;
36     double *PunteroMatriz_Calidad;
37     rplidar_response_measurement_node_t  nodes[8192];
38     size_t  count = _countof(nodes);
39     u_result ans;
40     // Genero el puntero de extracci n de datos
41     plhs[0] = mxCreateDoubleMatrix(1,count,mxREAL);// Datos de velocidad
42     plhs[1] = mxCreateDoubleMatrix(1,count,mxREAL);// Datos de posici n
43     plhs[2] = mxCreateDoubleMatrix(1,count,mxREAL);// Datos de calidad
44     // Reservo los espacio de memoria
45     PunteroMatriz_Distancia =(double *)mxGetData(plhs[0]);
46     PunteroMatriz_Posici n = (double *)mxGetData(plhs[1]);
47     PunteroMatriz_Calidad = (double *)mxGetData(plhs[2]);
48     // Generando la salida de datos
49     ans = drv->grabScanData(nodes, count); // Genera una actualizaci n de datos en 360 
50     drv->ascendScanData(nodes, count); // Acedo a los datos almacenados en el driver
51     for (int pos = 0; pos < count ; ++pos) {
52         PunteroMatriz_Distancia[pos]=(nodes[pos].distance_q2/4.0f);
53         RPLIDAR_RESP_MEASUREMENT_ANGLE_SHIFT)/64.0f;
54         PunteroMatriz_Calidad [pos]= (nodes[pos].sync_quality);
55     }
56 }
57 }

```

### Kodea 5 MatLab eta LiDAR arteko komunikazioa eteteko

```

1 ///////////////////////////////////////////////////////////////////
2 /////////////////////////////////////////////////////////////////// Parar EL LIDAR. ///////////////////////////////////////////////////////////////////
3 ///////////////////////////////////////////////////////////////////
4
5
6 // Incluir las librer as necesarias
7 #include "mex.h" // Libreria de MatLab
8 #include "include/rplidar.h" // Libreria del LIDAR
9 #include <windows.h>
10 #include "matrix.h"
11 #include <windows.h>
12 #include <conio.h>
13 #include <stdio.h>
14 #include <string.h>
15
16 using namespace rp::standalone::rplidar;
17 RPLidarDriver * drv;
18 void mexFunction(int nlhs, mxArray *plhs[], int nrhs, const mxArray *prhs[]) {
19     /*
20     nlhs Number of expected output mxArray
21     plhs Array of pointers to the expected output mxArray
22     nrhs Number of input mxArray
23     prhs Array of pointers to the input mx Array
24     */
25     void *vp;
26     long long *ip;
27     union {long long theinteger; void *thepointer;} ivp;
28     ip = (long long *) mxGetData(prhs[0]);
29     ivp.theinteger = *ip;
30     // Mi puntero es de un tipo espec fico que he de nombrar aqu  (RPLidarDriver*)
31     drv = (RPLidarDriver*)ivp.thepointer; // your pointer
32     printf("Goodbye\n");
33     drv->stop();
34     drv->stopMotor();
35     RPLidarDriver::DisposeDriver(drv);
36     drv = NULL;
37     printf("BYE \n");
38 }

```

### Kodea 6 LiDAR-reko informazioa MatLab-en gorde

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 %%%%%%%%% Sistema de Adquisici n de datos %%%%%%%%%
3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4 clear all
5 close all
6 clc
7 figure (1)
8 % Realizo el escaneado hasta que decido parar el programa
9 ButtonHandle = uicontrol('Style', 'PushButton', ...
10     'String', 'Stop loop', ...
11     'Callback', 'delete(gcf)'); % Boton para parar
12 pause(0.01);
13 % En primer lugar creo el dialogo con el Driver del LIDAR
14 LIDAR = RPLIDAR_CREATE; % Almaceno el puntero que guarda la comunicaci n
15
16 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```



---

```

17 %%%%                                OBTENER DATOS DEL LIDAR                                %%%%
18 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
19 cont = 1;
20 scan = {};
21 while (ishandle(ButtonHandle))
22     figure (2);
23     hold on
24     [ranges,angles,calidad] = RPLIDAR_SCAN(LIDAR); % LLamo a la funci3n de lectura
25     Ranges = ranges'/1000; % Adecuo unidades a metros;
26     Angles = angles'*pi/180; % Adecuo unidades a radianes
27
28     %scan{cont} = lidarScan(ranges,angles);
29     [X,Y] = pol2cart(Angles,Ranges);% Guardo en metros
30     plot(X,Y,'xr','LineWidth',1);
31     scan{cont}.Cartesian = [X,Y];
32     scan{cont}.Polar = [Angles, Ranges];
33     cont= cont +1;
34
35     pause(0.5);% adquisici3n cada 500ms
36     close (2); %Simplemente cierro la figura que quiero
37 end
38 RPLIDAR_STOP(LIDAR); % Cierro la aplicaci3n del lidar
39 %%% Guardo los datos obtenidos
40 A = dir('*.*mat'); % B'sco 'nicamente los archivos relacionados con .mat
41 Samples = size(A,1);
42 fileName= sprintf('%s%d','Scan_',Samples);
43 save(fileName);

```

---

### Kodea 7 Mapa sortzeko MatLab-eko programa

---

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 %%%%                                Mapeado el entorno del Lab                                %%%%
3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4 clear all
5 close all
6 clc
7 figure
8 % Realizo el esaneado hasta que decido parar el programa
9 ButtonHandle = uicontrol('Style', 'PushButton', ...
10     'String', 'Stop loop', ...
11     'Callback', 'delete(gcf)'); % Boton para parar
12 pause(0.01);
13 % En primer lugar creo el dialogo con el Driver del LIDAR
14 LIDAR = RPLIDAR_CREATE; % Almaceno el puntero que guarda la comunicaci3n
15
16 % Cuando ponemos por ejemplo (Distancia,2), el 2 hace referencia a las columnas; en cambio, el
17 1 hace referencia a las filas
18 n =1;
19 cont = 1;
20 datos.posrobot = [];
21 datos.Cartesian=[];
22 datos.Polares= [];
23 datos.calidad = [];
24 datos.Pose_Est =[];
25 datos = {};
26 z = 0;
27 while (ishandle(ButtonHandle))
28     z = z+1;
29     x=input('dame x:') % Introducir posici3n en X del AGV respecto origen (coord. absolutas)
30     y=input('dame y:') % Introducir posici3n en Y del AGV respecto origen (coord. absolutas)
31     thita=input('dame thita:') %Introducir orientaci3n del AGV respecto origen (coord.
32     absolutas)
33     [Distancia,angulo,calidad] = RPLIDAR_SCAN(LIDAR);
34     % De polares a Cartesianas (pol2cart fallaba)
35     for h = 1:size(Distancia,2)
36         Y(h) = Distancia(h)*cos((angulo(h)*pi/180));
37         X(h) = Distancia(h)*sin((angulo(h)*pi/180));
38     end
39     plot(X,Y,'xr','LineWidth',1);
40     datos{z}.posrobot=[x,y,thita];
41     datos{z}.Cartesian=[X,Y];
42     datos{z}.Polares =[Distancia',angulo'];
43     datos{z}.calidad =[calidad'];
44     %pause;
45 end
46 % Funci3n de guardado
47 A = dir('datos_*.mat');
48 Guardar = size(A,1);
49 filename = sprintf('%s%d','datos_',Guardar);
50 save(filename,'datos','-v7.3')
51 RPLIDAR_STOP(LIDAR); % Dentengo el Driver

```

---

## Kodea 8 Mapa eraikitzeko MatLab-eko programa

---

```

1  Mapa = [];
2  Mapa.Cartesianas = [];
3  Mapa.PoseAbsoluta = [];
4  cont = 1 % Contador inicia desde 1
5  close all
6  A = dir('datos_*.mat') % definir A, buscar en el directorio todos los archivos que comiencen
con datos_ y con extensión .mat
7  Samples = size(A,1) % tamaño de las muestras que estan en las filas (1) de A.
8  gg = 1;
9  for h = 1:Samples
10     load (A(h).name) % Cargar los datos tomados con el lidar
11     cantidad_Datos = size(datos,2) % tamaño de la cantidad de datos guardados en las columnas
(2) de datos
12     for n = 1:cantidad_Datos ; %Guardar en AgvX, AgvY, AgvThita la posición del AGV respecto
al origen (Coord. absolutas) y en AgvX_Correct y AgvY_Correct la posición del AGV respecto al
origen teniendo en cuenta la rotación del ultimo movimiento.
13         S = size(datos{n}.Cartesian,1) % Guardamos en S el tamaño de las filas (1) de cada
medición, desde la primera hasta la ultima.
14
15         % Absolutas
16         if n~=1 %Entramos a partir de la segunda medición (n distinto de 1)
17             AgvThita = datos{n}.posrobot(1,3)+Mapa.PoseAbsoluta(gg-1,3); % Obtenemos la
orientación del AGV respecto al origen sumando la orientación absoluta del estado anterior a
la orientación relativa (ultimo movimiento) respecto a la medición anterior.
18             AgvX = (datos{n}.posrobot(1,1)*cos(AgvThita*pi/180)-
datos{n}.posrobot(1,2)*sin(AgvThita*pi/180))+Mapa.PoseAbsoluta(gg-1,1); % Obtenemos la
posición absoluta del AGV en X sumando la posición absoluta en X d despues de la medición
anterior a la posición relativa en X respecto la medición anterior.
19             AgvY = (datos{n}.posrobot(1,2)*cos(AgvThita*pi/180)+
datos{n}.posrobot(1,1)*sin(AgvThita*pi/180))+Mapa.PoseAbsoluta(gg-1,2); % Obtenemos la
posición absoluta del AGV en Y sumando la posición absoluta en Y despues de la medición
anterior a la posición relativa en X respecto la medición anterior.
20             Mapa.PoseAbsoluta(gg,:) = [AgvX,AgvY,AgvThita]; % Guardamos cada una de las
posiciones en X,Y,Thita en el mapa.
21             else % Entramos SOLO en la primera medición (n=1), fijamos el sistema de
referencia inicial respecto a un sistema origen (0,0,0)
22                 Mapa.PoseAbsoluta(gg,:) =
[datos{n}.posrobot(1,1),datos{n}.posrobot(1,2),datos{n}.posrobot(1,3)]; % Guardamos la primera
posición X,Y,Thita en el mapa.
23                 AgvX = datos{n}.posrobot(1,1); % Guardamos en AgvX la posición en X de nuestra
primera medición.
24                 AgvY = datos{n}.posrobot(1,2); % Guardamos en AgvY la posición en Y de nuestra
primera medición.
25                 AgvThita = datos{n}.posrobot(1,3); % Guardamos en AgvThita la orientación de
nuestra primera medición.
26             end
27             for j = 1:S % S es el numero de filas (1) de Datos{n}.cartesian
28
29                 if datos{n}.Cartesian(j,2)>=0 && datos{n}.Polares(j,1) ~= 0
30                     X = datos{n}.Cartesian(j,1);
31                     Y = datos{n}.Cartesian(j,2);
32                     X_Correct = X*cos(AgvThita*pi/180) - Y*sin(AgvThita*pi/180); % MATRIZ DE
ROTACION en Z: Obtenemos la posición absoluta en X teniendo en cuenta la rotación que ha
podido realizar el AGV.
33                     Y_Correct = Y*cos(AgvThita*pi/180) + X*sin(AgvThita*pi/180); % MATRIZ DE
ROTACION en Z:Obtenemos la posición absoluta en Y teniendo en cuenta la rotación que ha podido
realizar el AGV.
34                     Mapa.Cartesianas(cont,1) = X_Correct + AgvX; %
35                     Mapa.Cartesianas(cont,2) = Y_Correct + AgvY; %
36                     cont = cont + 1;
37                 end
38             end
39             gg = gg +1;
40         end
41     end
42
43     save('Mapa','Mapa','-v7.3')
44     figure(2)
45     grid on % agraga lineas de cuadrícula a los ejes actuales
46     plot(Mapa.Cartesianas(:,1),Mapa.Cartesianas(:,2),'xr')

```

---

## Kodea 9 LiDAR funtzioa

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Mediciones en el Mapa %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %%%% Autores: Ekaitz Zulueta y Daniel Teso
3  % Funci3n Pensada para simular el LIDAR y la adquisici3n de datos
4  % Emplea un mapa realizado en el sotano de la universidad.
5  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6  %%%% Unidades %%%%
7  % AGVPose [m,m,rad]
8  % Points [rad,m] Cartesianas
9  % Mapa.Cartesianas [mm,mm]
10 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
11 function [Points] = LidarMap (Mapa,AGVPose)
12     cont = 1; % Un contador para el If
13     Cartesian= [];
14     Correct = [];
15     Points = [];
16     MaxLidar = 8000; % mm
17
18 % For para analizar cada punto del Mapa
19     for h = 1:size(Mapa.Cartesianas,1)
20         % Extraer los puntos del Mapa
21         X = Mapa.Cartesianas(h,1);%/1000;
22         Y = Mapa.Cartesianas(h,2);%/1000;
23         % Me quedo solo con los datos que estan dentro del rango del LIDAR
24         if X <= AGVPose(1)+ MaxLidar && X>=AGVPose(1)-MaxLidar && Y <=AGVPose(2)+MaxLidar% &&
Y >= AGVPose(2)-MaxLidar
25             X_Correct = (X-AGVPose(1))*cos((90-AGVPose(3))*pi/180) - (Y-AGVPose(2))*sin((90-
AGVPose(3))*pi/180);% Traslado y roto
26             Y_Correct = (Y-AGVPose(2))*cos((90-AGVPose(3))*pi/180) + (X-AGVPose(1))*sin((90-
AGVPose(3))*pi/180);%
27             Correct(cont,:) = [X_Correct,Y_Correct]; % Estos son los puntos que ve el lidar,
28             % Las coordenadas de estos puntos estan situadas en el
29             % origen del lidar.
30             Cartesian(cont,:) = [X,Y];
31             cont = cont +1;
32         end
33     end
34 % Esposible que existan puntos que esten en el mismo grado, pero el radio
35 % sea diferente, Ejemplo : Si X1 = [10m,2rad] y X2 = [12m,2rad].
36 % Tecnicamente el punto X2 no lo podr3a ver el lidar, por ello hay que
37 % descartarlo.
38     if isempty(Cartesian) ~= 1
39         [Thetac, Rc] = cart2pol(Correct(:,1),Correct(:,2));
40         % Analizo por cada angulo los puntos que existen a diferentes radios. Si
41         % coinciden solo puedo ver el m's cercano, el resto quedar3a oculto porque
42         % en la realidad lo tapar3a ese objeto que ha visto al inicio.
43         contador = 1;
44         for j = 0:0.01:pi;
45             [Filas,Columnas,Valor]=find(round(Thetac,2)==j); % busco aquellos valores que
concuerden
46             if isempty(Valor)~=1 % En el caso que si exista una coincidencia me quedare con el
radio m's peque0o
47             % (Probar si la matriz est vac3a), f = isempty(A) devuelve l3gico verdadero (1) si A
es una matriz
48             %vac3a y l3gico falso ( 0) de lo contrario. Una matriz vac3a tiene al menos una
dimensi3n de tama0o
49             %cero, por ejemplo, 0 por 0 o 0 por 5.
50             R_min = min(Rc(Filas));
51             % polar(j,R_min,'bo');
52             [Xp,Yp] = pol2cart(j,R_min); % de polares a cartesianas
53             % Pongo los puntos en coordenadas encima del AGV.
54             V = (Xp)*cos((AGVPose(3)-90)*pi/180) - (Yp)*sin((AGVPose(3)-
90)*pi/180)+AGVPose(1);% Roto Y Traslado
55             Z = (Yp)*cos((AGVPose(3)-90)*pi/180) + (Xp)*sin((AGVPose(3)-
90)*pi/180)+AGVPose(2);% + POSE(2);
56             Points(contador,1) = V;
57             Points(contador,2) = Z;
58             %             Points(contador,1) = j;
59             %             Points(contador,2) = R_min;
60             contador = contador +1;
61         end
62     end
63     else % Y si no hay datos devuelvo un dato que no existe
64         Points = [1000000,0];
65     end
66     if isempty(Points) == 1 %0 si la matriz esta vacia.
67         Points = [1000000,0];
68     end
69
70 End

```

## Kodea 10 Mapa irudi batean bilakatu

```

1 clear all; close all; clc
2 % Carga de parámetros
3 BuildMap = [];
4 BuildMap.Cartesianas = [];
5 BuildMap.AgvPose = [];
6
7 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
8 %%% Esto lo hago para simular la creación del mapa
9 load('Mapa.mat'); % Cargo el mapa sobre el cual me voy a ubicar.
10 AGVPose = [0 00 90]; % [m m grados] ABS
11 % Cálculo desplazamiento en función del task time
12 AGVSpeed = 0.2; %m/s
13 TaskTime = 50; %ms
14 movement = AGVSpeed *TaskTime;
15 Ut = [0 movement 0]; % [mm mm grados] RELATIVO
16 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
17
18 v = 2000;
19 contador = 1;
20 for i = 1:v
21     BuildMap.AgvPose(i,:) = AGVPose;
22     %%% JARRI LIDARREN FUNTZIOA EDO LIDARREN DATUAK
23     [Points] = LidarMap (Mapa,AGVPose);
24
25     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
26     for g = 1:size(Points)
27         if Points(g,1) ~= 0 & Points(g,2) ~= 0
28             BuildMap.Cartesianas(contador,:) = Points(g,:);
29             contador = 1 + contador;
30         end
31     end
32     AGVPose = AGVPose + Ut;
33
34 end
35 figure(1)
36 plot(BuildMap.Cartesianas(:,1),BuildMap.Cartesianas(:,2),'xr')
37 title ('Lortutako mapa')
38
39
40
41 %% Irudiaren tamaina lortzeko.
42 t = 1 ;
43 if t == 1
44     XDistMin = min (BuildMap.Cartesianas(:,1))-1000;
45     XDistMax = max (BuildMap.Cartesianas(:,1))+1000;
46     XDist = round( (XDistMax -XDistMin)/10,0);
47     YDistMin = min (BuildMap.Cartesianas(:,2))-1000;
48     YDistMax = max (BuildMap.Cartesianas(:,2))+1000;
49     YDist = round((YDistMax -YDistMin)/10,0);
50     %% Irudiaren tamaina jartzen dut
51     Image = zeros(XDist,YDist);
52     %% Puntuak piseleten bilakatu
53     for h = 1 : size(BuildMap.Cartesianas,1)
54         Ximg = round((BuildMap.Cartesianas(h,1)-XDistMin)/10,0);
55         Yimg = round((BuildMap.Cartesianas(h,2)-YDistMin)/10,0);
56         Image (Ximg,Yimg) = 255;
57     end
58     figure(2)
59     imshow(Image);
60     imwrite(Image,'Map2.png')
61     MapData = [XDistMin,YDistMin];
62     save('MapData2','MapData')
63
64 end

```

## Kodea 11 Iruditik maparen informazioa lortu.

```

1 function [Points,ZSimple] = ImageLidar (Image,
2 AgvPose,LidarDistMax,LidarDistMin,LidarThetaMax,LidarThetaMin,MapData)
3 %%% Funtzio honek AGV-ren kokapenaren arabera datuak ateratzen ditu
4 % behar diren zenbait datu
5 Points = [];
6 cont = 1;
7 AgvX = AgvPose(1);
8 AgvY = AgvPose(2);
9 % Aztertu lidarrak zein puntu ikusten dituen.
10 for ImgX = 1: size(Image,1)
11     for ImgY = 1: size(Image,2)
12         if Image(ImgX,ImgY)~= 0; % Puntu bat daukagunean, puntu hori non eman den
13             controlatu behar da
14                 Xpoin = ImgX*10 +MapData(1,1); %%% Conversiones por constantes que he
15                 Ypoin = ImgY*10 +MapData(1,2);
16                 %%% Xpoin e Ypoin mm dago.
17                 %%% Rotazioa Agvaren egoeratik aztertzeko.
18                 Xpoint = Xpoin * cos((-AgvPose(3))) - Ypoin * sin((-AgvPose(3)));
19                 Ypoint = Xpoin * sin((-AgvPose(3))) + Ypoin * cos((-AgvPose(3)));
20                 if Xpoint <= AgvX + LidarDistMax && Xpoint >= AgvX + LidarDistMin % Erradio
21                     concretu batean

```

```

19         if Ypoint <= AgvY + LidarThetaMax && Ypoint >= AgvY + LidarThetaMin
20             %% Berritzuli Agv egoeratik mapa absoluturra
21             Xpuntua = Xpoint * cos((AgvPose(3))) - Ypoint * sin((AgvPose(3)));
22             Ypuntua = Xpoint * sin((AgvPose(3))) + Ypoint * cos((AgvPose(3)));
23             Points(cont,:) = [Xpuntua,Ypuntua] ;
24             cont = cont +1;
25         end
26     end
27 end
28 end
29 end
30 %Lidarrak ezin ditu ikuzi pareta atzean dauden puntuak. Horregatik
31 %ikusitako puntu guztietatik
32 % Horretarako, Aurkitutako puntual (Points) koordenatu polarretara
33 % pasatuko ditut. Horrera theta bakoitzeko erradio guztiak ikertuko
34 % ditut, eta erradio txikiarekin geratuko naiz. Erradio Txikiena Agv-tik
35 % gertuena delatko, eta puntu hori izango da, lidarrak ikusiko duena.
36 [Thetas,Rs] = cart2pol(Points(:,1),Points(:,2));
37 Z = [];
38 contador = 1;
39 % Aurkitutako Theta guztietatik, zegurazki decimal azko izango dituzte,
40 % eta errealtatean decimal gutxiekien lan egiten da. Horren ondorioz,
41 % decimal bakar batera borobilduko ditut theta guztiak
42 ThetaDec = round(Thetas,2);
43 % Orain thetaDec guztiak aztertuko ditut, eta zenbaki berak dituztenak
44 % erradioa aztertuko ditut
45 for conter = 1:size(ThetaDec,1)
46     Rlocalization = find(ThetaDec == ThetaDec(conter));
47     if ~isempty(Rlocalization)
48         Rmin = min(Rs(Rlocalization));
49     else
50         Rmin = Rs(conter);
51     end
52     [ZSimple(conter,1),ZSimple(conter,2)] = pol2cart (ThetaDec(conter),Rmin);
53 end
54

```

## Kodea 12 MatLab bidezko Roomba Mode

```

1  clear all
2  close all
3  clc
4
5  %addpath('Win32Mex'); %Para 32 bits
6  addpath('Win64Mex'); %Para 64 bits
7  IP = '169.254.153.119.1.1'
8
9  % En primer lugar creo el dialogo con el Driver del LIDAR
10 LIDAR = RPLIDAR_CREATE; % Almaceno el puntero que guarda la comunicaci n
11 n =1; % Siwtch para salir del while
12 cont = 1; % Siwtch para moverse entre el Modo Autom tico y Manual
13 % Variables internas
14 Speed = 0;
15 Steer = 0;
16 variable = 'STOP'
17 while n == 1
18     %% TURNO DE LEER VALORES
19     % Leer variable REAL
20     [a,b] = Read_REAL(IP,851,'MAIN.nReal');
21     % Leer datos del lidar
22     [variable,Steer] = LIDARTime(LIDAR);
23
24     if cont == 1
25         % Sistema para poder Inicializar el AGV
26         % Si el AGV se ha detenido por el comando STOP entrar aqu  tambien
27         variable=input('Qu  desea hacer? START o EXIT: ','s');
28         variable = upper(variable);
29         cont = 0;
30     end
31     %% TURNO DE ENVIAR VALORES
32     switch variable
33     case 'START' %% Iniciar la marcha del AGV
34         h = Write_BOOL(IP,851,'MAIN.bBool',true);
35         h = Write_BYTE(IP,851,'MAIN.nByte',0);%% Mando se al
36         h = Write_BYTE(IP,851,'MAIN.nByte',255); %% Reseteo el case del AGV
37         variable = 'MOVE'; % Una vez Inicializado lo puedo comandar;
38         Speed = 0.2;% Inicializo el Movimiento
39         Steer = 0; % Inicializo el el angulo de movimiento
40     case 'STOP' %% Detener la marcha del AGV
41         h = Write_BOOL(IP,851,'MAIN.bBool',false);
42         h = Write_BYTE(IP,851,'MAIN.nByte',0);
43         h = Write_BYTE(IP,851,'MAIN.nByte',255);%% Reseteo el case del AGV
44         cont = 1; % Vuelvo a tener que inicializar la m quina
45     case 'MOVE' %% Determinar la velocidad crucero
46         %Speed=input('Qu  velocidad desea en m/s?: ');
47         Speed = 0.11;
48         Speed = Speed *1000; % Velocidad en mm/s evito usar reales
49         h = Write_UINT(IP,851,'MAIN.nUint',Speed)
50         %Steer=input('Qu  direcci n desea en grados?: ');
51         Steer = Steer;% * pi /180; % Paso a radianes
52         h = Write_INT(IP,851,'MAIN.nInt',Steer);
53         h = Write_BYTE(IP,851,'MAIN.nByte',1); %% Mando la se al para actuar.
54         h = Write_BYTE(IP,851,'MAIN.nByte',255);%% Reseteo el case del AGV

```

---

```

55     case 'EXIT'
56         h = Write_BOOL(IP,851,'MAIN.bBool',false);
57         n=0;
58     end
59     pause (0.01); % Tiempo de cilco de 10ms
60 end
61 RPLIDAR_STOP(LIDAR); % Dentengo el Driver
62
63 function [Var,thita] = LIDARTime(LIDAR)
64 % Esta funcion se encarga de la lectura del LIDAR y Tratar si debe seguir
65 % hacia delante, girar 90 grados o detenerse en caso de que encuentre un
66 % objeto
67 [Distancia,angulo,calidad] = RPLIDAR_SCAN(LIDAR);
68 [Var, thita] = SECURITYLIDAR (Distancia, angulo)
69 end
70 function [Var,thita] = SECURITYLIDAR (Distancia, angulo)
71 % Eliminar todos los ceros posibles para poder analizar los posibles
72 % problemas que existan al rededor del AGV.
73 Radio = [];
74 ang=[];
75 cont = 1;
76 for h = 1: size(Distancia,2)
77     if Distancia(h) ~= 0
78         Radio(cont) = Distancia(h);
79         ang (cont) = angulo(h);
80         cont = cont +1;
81     end
82 end
83 % Miro cual es el punto que m's cerca esta del AGV
84 [valor,lugar] = min(Radio);
85 if valor <= 400 % Si esta inferior a esta distancia detener el AGV
86     Var = 'STOP';
87     thita = 0;
88 else % No he de parar el AGV pero analizo otros casos
89     if valor <= 700 % Si esta cerca y enfrente del AGV he de girar el AGV
90         if ang(lugar)<90
91             Var = 'MOVE';
92             thita = 90;% Si el punto esta en la derecha giro a la Izquierda
93         elseif ang(lugar)>270
94             Var = 'MOVE';
95             thita = 270; % Si el punto esta en la izquierda giro a la derecha
96         else
97             Var = 'MOVE';
98             thita = 90;
99         end
100     else% No existe ningun problema Continua hacia delante.
101         Var = 'MOVE';
102         thita = 0;
103     end
104 end
105 end

```

---

### Kodea 13 C++ LiDAR eta TwinCat komunikazioa

---

```

1 #include <iostream>
2 #include <windows.h>
3 #include <conio.h>
4 #include <stdio.h>
5 #include "TcAdsDef.h"
6 #include "TcAdsAPI.h"
7 #include "rplidar.h"
8
9
10 using namespace std;
11 using namespace rp::standalone::rplidar;
12
13 #define ESCAPE_KEY 27
14 #define delay(x)  ::Sleep(x)
15
16 // Teclado() {
17 //     int ValorTeclado;
18 //
19 // }
20 //}
21 void main()
22 {
23     int ValorTeclado;
24     long nErr, nPort;
25     AmsAddr Addr;
26     PamsAddr pAddr = &Addr;
27     ULONG lHdlVar;//, nData;
28     FLOAT nData[600];
29     FLOAT nPose[600];
30     ULONG nTeclado;
31     char szVar[] = { "GVL_Lidar.arrDato" };//{"MAIN.PLCVar"};
32     char szPos[] = { "GVL_Lidar.arrPose" };
33     char szTeclado[] = { "GVL_DigitalIO.nTeclado" };
34     // LIDAR
35     rplidar_response_measurement_node_t nodes[600];
36     size_t countRP = _countof(nodes);
37     u_result ans;
38     // Open communication port on the ADS router

```

---

```

39     nPort = AdsPortOpen();
40     nErr = AdsGetLocalAddress(pAddr);
41     if (nErr) std::cerr << "Error: AdsGetLocalAddress: " << nErr << '\n';
42     pAddr->port = 851;
43
44     printf("Hi");
45
46     //////////// Genero los par·metros de la comunicaci3n LIDAR ////////////
47
48     // Puerto de comunicaciones del LIDAR
49     const char * opt_com_path = "com4";
50     // Baudios de comunicacion
51     _u32         opt_com_baudrate = 115200;
52
53     RPlidarDriver *drv = RPlidarDriver::CreateDriver(0x0);
54
55     if (!drv) {
56         printf("No memory");
57     }
58     if IS_FAIL(drv->connect(opt_com_path, opt_com_baudrate)) {
59         printf("Error");
60         return;
61     }
62     else {
63         //drv->reset();
64         printf("%d", drv);
65         rplidar_response_device_info_t devinfo;
66         u_result op_result;
67         op_result = drv->getDeviceInfo(devinfo);
68         printf("%d", op_result);
69         if (IS_OK(op_result)) {
70             printf("Start");
71             delay(100);
72             drv->startMotor();
73             printf("Motor");
74             drv->startScan(0, 1);
75         }
76         else {
77             printf("No se Ha conectado");
78             drv = NULL;
79         }
80         printf("Esperando a velocidad m·xima");
81         delay(3000);
82
83
84         int ftt = 1;
85         while (ftt == 1) {
86             // Lectura de teclado
87             //ValorTeclado = Teclado();
88             if (_kbhit()) {
89                 ValorTeclado = _getch();
90                 //return ValorTeclado;
91             }
92             // Read LIDAR
93             drv->grabScanData(nodes, countRP); // Genera una actualizaci3n de datos en 360
94             drv->ascendScanData(nodes, countRP); // Acedo a los datos almacenados en el
95 driver
96             for (int pos = 0; pos < countRP; ++pos) {
97                 nData[pos] = 0;
98                 nPose[pos] = 0;
99                 nData[pos] = (nodes[pos].distance_q2 / 4.0f);
100                nPose[pos] = (nodes[pos].angle_q6_checkbit >>
RPLIDAR_RESP_MEASUREMENT_ANGLE_SHIFT) / 64.0f;
101
102            }
103            // Send Data
104            // Get the handle of the PLC-variable <szVar>
105            nErr = AdsSyncReadWriteReq(pAddr, ADSIGRP_SYM_HNDBYNAME, 0x0, sizeof(lHdlVar),
&lHdlVar, sizeof(szVar), szVar);
106            // Envio Los radio
107            nErr = AdsSyncWriteReq(pAddr, ADSIGRP_SYM_VALBYHND, lHdlVar, sizeof(nData),
&nData);
108            // Cojo otra vez el Handle
109            nErr = AdsSyncReadWriteReq(pAddr, ADSIGRP_SYM_HNDBYNAME, 0x0, sizeof(lHdlVar),
&lHdlVar, sizeof(szPos), szPos);
110            // envio los datos
111            nErr = AdsSyncWriteReq(pAddr, ADSIGRP_SYM_VALBYHND, lHdlVar, sizeof(nPose),
&nPose);
112            // Cojo otra vez el Handle para enviar los valores del teclado
113            nErr = AdsSyncReadWriteReq(pAddr, ADSIGRP_SYM_HNDBYNAME, 0x0, sizeof(lHdlVar),
&lHdlVar, sizeof(szTeclado), szTeclado);
114            // envio los datos
115            nErr = AdsSyncWriteReq(pAddr, ADSIGRP_SYM_VALBYHND, lHdlVar, sizeof(ValorTeclado),
&ValorTeclado);
116            delay(1);
117        } // Programa continuo hasta pulsar espacio
118
119        // Release the handle of the PLC-variable
120        nErr = AdsSyncWriteReq(pAddr, ADSIGRP_SYM_RELEASEHND, 0, sizeof(lHdlVar), &lHdlVar);
121        if (nErr) std::cerr << "Error: AdsSyncWriteReq: " << nErr << '\n';
122
123        // Close the communication port
124        nErr = AdsPortClose();
125        if (nErr) std::cerr << "Error: AdsPortClose: " << nErr << '\n';
126        printf("Goodbye\n");

```

```

127     drv->stop();
128     drv->stopMotor();
129     RPLidarDriver::DisposeDriver(drv);
130     drv = NULL;
131 }
132 }

```

#### Kodea 14 Roomba Mode funtzioa TwinCat bidez programatuta

```

1 IF GVL_Seguridad.ValorMin<nDistGiro THEN
2   ThetaLidar := GVL_Lidar.arrPose [GVL_Seguridad.Lugar];
3   IF ThetaLidar >= 360 THEN
4     ThetaLidar := ThetaLidar-360;
5   END_IF
6   // giro
7   IF ThetaLidar>=0 AND ThetaLidar<=120 THEN
8     //GIRO IZQUIERDA
9
10    Vc:=GVL_MARGA.RoombaSpped;
11    Direccion:= -GVL_MARGA.RoombaDireccion;
12  ELSIF ThetaLidar>=250 AND ThetaLidar<360 THEN
13    //GIRO DERECHA
14    Vc:=GVL_MARGA.RoombaSpped ;
15    Direccion:= GVL_MARGA.RoombaDireccion;
16  ELSE
17    // Recto
18    Vc:=GVL_MARGA.RoombaSpped ;
19    Direccion:= 0;
20  END_IF
21 ELSE
22   // recto
23   Vc:=GVL_MARGA.RoombaSpped;
24   Direccion:= 0;
25 END_IF

```

#### Kodea 15 Segurtasun funtzioa TwinCat bidez programatuta

```

1  ///// Lidarraren datu guztiak aztertzen dira
2  FOR posicion:=1 TO 600 BY 1 DO
3    IF GVL_Lidar.arrDato[posicion]>0 THEN
4      IF GVL_Lidar.arrDato[posicion]< GVL_Seguridad.ValorMin THEN
5        GVL_Seguridad.ValorMin:= GVL_Lidar.arrDato[posicion];
6        GVL_Seguridad.Lugar:=posicion;
7
8      END_IF
9    END_IF
10   ///// Si ese punto esta dentro de un umbral el AGV se ha de detener.
11  END_FOR
12  IF GVL_Seguridad.ValorMin<GVL_Seguridad.nDistSeguridad OR GVL_Lidar.arrDato[1] < 0 THEN
13    GVL_Seguridad.bSeguridad := TRUE;
14    ///// Primero analizo si el punto esta muy cerca como para detener el AGV, pero además de ello
15    compruebo que si he quitado la seguridad por el teclado
16    IF GVL_Seguridad.bSeguridad = TRUE THEN
17      GVL_Seguridad.bStopAGV := TRUE;
18    ELSIF GVL_DigitalIO.bTeclado = TRUE AND GVL_Seguridad.bSeguridad = TRUE THEN
19      GVL_Seguridad.bStopAGV := FALSE;
20    END_IF
21  ELSE
22    GVL_Seguridad.bStopAGV := FALSE;
23  END_IF

```

#### Kodea 16 ( 32 ) ekuazioaren programazioa

```

1 function Particulas = GeneroParticulas (Pose_ant,Npart,Ut)
2 % Variables del programa
3   Particulas = zeros(Npart,3);
4
5 % Genero cada partícula en función de la pose anterior del AMR y luego las
6 % desplazo
7   for i = 1:Npart
8     % Calculo un grado de incertidumbre para cada partícula
9     X_rand = round(random('norm',Pose_ant(1),0.04,1,1),3);
10    Y_rand = round(random('norm',Pose_ant(2),0.04,1,1),3);
11    Theta_rand = round(random('norm',Pose_ant(3),0.01,1,1),3);
12    % Actualizo el estado de cada Partícula
13    Particulas(i,1) = Ut(1)+X_rand; % X
14    Particulas(i,2) = Ut(2)+Y_rand; % Y
15    Particulas(i,3) = Ut(3)+Theta_rand; % Theta
16    % Estimo la posición del AGV conociendo el Desplazamiento
17
18   end
19 end
20

```



Kodea 17 LiDAR-ren informazioaz baliatuz, desplazamendu eta biraketaren kalkulua

---

```

1 function [R,T] = Regrecion_ICP (data2,data1,method)
2
3     % data1 = [x(t)1 x(t)2 x(t)3 ...]
4     % data2 = [x(t+1)1 x(t+1)2 x(t+1)3 ...]
5     % x=[x y z]'
6
7     %Variables
8     preError=0;%
9     dError=1000000;%
10    EPS=0.0001;%
11    maxIter=4000;%
12    count=0;%
13
14    R=eye(2);
15    T=zeros(2,1);%
16
17    while ~(dError < EPS)
18        count=count+1;
19
20        [ii, error]=FindNearestPoint(data1, data2);
21
22        if method == "ICP" % Iterative Closed Point erabili
23            [R1, t1]=SVMotionEstimation(data1, data2, ii);%
24
25        elseif method == "LR" % Erregresio lineala erabili
26            [R1, t1]=LRegresionMotionEstimation(data1, data2, ii);
27            end
28
29        data2=R1*data2;
30        data2=[data2(1,:)+t1(1) ; data2(2,:)+t1(2)];
31        R = R1*R; % Biraketak eragin, minimizazioa lortzeko
32        T = R1*T + t1; % Traslazioak eragin minimizazioa lortzeko
33
34        dError=abs(preError-error);%
35        preError=error;%
36
37        if count > maxIter %
38            disp('Max Iteration');return;
39        end
40    end
41
42 end

```

---

Kodea 18 Bi laginketen elkartasun kodea.

---

```

1 function [index, error]=FindNearestPoint(data1, data2)
2
3     m1=size(data1,2);
4     m2=size(data2,2);
5     index=[];
6     error=0;
7
8     for i=1:m1
9         dx=(data2-repmat(data1(:,i),1,m2));
10        dist=sqrt(dx(1,:).^2+dx(2,:).^2);
11        [dist, ii]=min(dist);
12        index=[index; ii];
13        error=error+dist;
14    end
15 end

```

---

Kodea 19 Singular Value Desconposition kodea

---

```

1 function [R, t]=SVMotionEstimation(data1, data2, index)
2     M = data1;
3     mm = mean(M,2);
4     S = data2(:,index);
5     ms = mean(S,2);
6     Sshifted = [S(1,:)-ms(1); S(2,:)-ms(2)];
7     Mshifted = [M(1,:)-mm(1); M(2,:)-mm(2)];
8
9     W = Sshifted*Mshifted';
10
11    [U,A,V] = svd(W); % MatLab-ek emandako funtioa
12
13    R = (U*V')';
14    t = mm - R*ms;
15 end

```

---

## Kodea 20 Erregresio linealaren kodea

```

1 function [R, T]=LRegressionMotionEstimation(Sact, Sant, index)
2 % Datuen Prestakuntza
3   Sant = Sant(:,index);
4   cont = size(Sant,2);
5   Phi = zeros(cont*2,4);
6   Y = zeros(cont*2,1);
7
8   for i = 1:cont
9     Phi(i,:) = [-Sant(2,i),Sant(1,i),1,0];
10    Y(i,1) = [Sact(1,i)];
11  End
12
13  gg = 1;
14
15  for i = cont+1:cont*2
16    Phi(i,:) = [Sant(1,gg),Sant(2,gg),0,1];
17    Y(i,1) = [Sact(2,gg)];
18    gg = gg + 1;
19  End
20
21  Regression = inv(Phi'*Phi)*Phi'*Y; % Erregresio linealaren funtzioa
22
23  % Emaitzak
24  R = [Regression(2),-Regression(1);Regression(1),Regression(2)];
25  T = [Regression(3);Regression(4)];
26 end

```

## Kodea 21 Probabilitate funtzioa

```

1 function [Estimada,Probabilidad] = Prob(R,T, ParticlePose)
2 % Partikula bakoitzaren probabilitatearen funtzioa
3   thita = 13; % Balio finko bat
4   P1 = 1/exp((T(1)*thita)^2);
5   P2 = 1/exp((T(2)*thita)^2);
6   P3 = 1/exp(acos(R(1))^2);
7
8   Estimada = [ParticlePose(1)*P1, ParticlePose(2)*P2, ParticlePose(3)*P3];
9
10  Probabilidad = [P1,P2,P3];
11 end

```

## Kodea 22 Partikula iragazkiaren funtzio nagusia

```

1 % Mapa kargatu
2 load('Mapa.mat');
3
4 % Beste datu batzuk
5 Npart = 30; % Partikula Kopurua
6 AGVIniPose = [0 0 90]; % [m m gradu] ABS
7 Ut = [0 0.01 0]; % [m m gradu] RELATIVO
8 Objetos = []; % cartesianas [m m] ABS
9 Pose_ant = AGVIniPose;
10 AGVPose = AGVIniPose + Ut;
11
12 %% For edo while bat ezarri
13
14 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Partikula Iragazkiaren funtzioa %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
15
16 Estimada = [];
17 Probabilidad = [];
18
19 % Partikulen sorrera
20 Particulas = GeneroParticulas (Pose_ant,Npart,Ut);
21
22 %%%% LIDAR-raren simulazioa -> Sistema errealearen simulazioa -> Funtzio hau kendu
23 errealtatean abiarazteko eta zuzenean LiDAR-rak emante dituen datuak erabili
24 [ZLidar] = LidarSimulator (Mapa,AGVPose);
25 Gra2 = []; R2 = [];
26 [Gra2,R2] = pol2cart(ZLidar(:,1),ZLidar(:,2));
27 ZLidark = [Gra2,R2];
28
29 %%%% Partikula bakoitzaren probabilitatea kalkulatu
30 for part = 1: Npart
31   % Mapatik datuak lurty
32   [Z] = LidarSimulator (Mapa,Particulas(part,:));
33
34   % Lortutako datuak Polarretatik Kartesianetara pasa
35   Gra1 = [];
36   R1 = [];
37   [Gra1,R1] = pol2cart(Z(:,1),Z(:,2));
38   Zk = [Gra1,R1];
39
40   %Partikulak ikusten dituen oztupoak LiDAR-ak ikusten dituenarekin
41   %konparatu egiten da. ICP edo Erregresio lineala bitartez

```

---

```

42     [R,T] = Regresion_ICP (Zk'*1000,ZLidark'*1000,"LR"); %*1000 milimetrotara pasatzeko
43     T = T/1000;
44     AGVParticle = Particulas(part,:);
45     [Estimada(part,:),Probabilidad(part,:)] = Prob(R,T, Particulas(part,:));
46
47     end
48
49     Pose_est = sum(Estimada)./sum(Probabilidad); %Estimatutako posearen kalkulua
50
51     %Hurrengo abiaraketarako prestatu datuak
52     Pose_ant = Pose_est;
53     AGVPose = AGVPose + Ut; %HurreMugitu AMR posizioa -> Soilik simulazioa delako

```

---

### Kodea 23 Laginketak sortzeko kodea

---

```

1     Nparticulas = 3000;
2     cont = 1;
3     LaginketaKargatu = 1; % 0 -> laginketa berriz egin 1-> Kargatu gordetako laginketa
4
5     if LaginketaKargatu == 1
6         Fitxategi = dir("TrainData/TrainData*");
7         Kopurua = size(Fitxategi,1)
8         Mesua = strcat(int2str(Kopurua),"Fitxategi dituzu aukeratu 1-etik ",int2str(Kopurua),"ra:");
9         Fitxategi = input(Mesua)
10        Kargatu = strcat("TrainData/TrainData",int2str(Fitxategi),".mat")
11        load(Kargatu)
12    else
13
14        for part = 1:Nparticulas % bucle para obtener los datos de cada partícula
15            PartX = round(random('norm',0,1000,1,1));
16            PartY = round(random('norm',0,40000,1,1));
17            PartZ = round(random('norm',0,360,1,1),1);
18            Train(cont).POSE = [PartX PartY PartZ]; % Genero la pose de la partícula
19
20            % Analizo los puntos del lidar para cada pose
21            [Train(cont).Cartesian,Train(cont).Correct,...
22            Train(cont).Points]=MedidasLIDAR(PartX,PartY,PartZ,Mapa,MaxLidar); %%
23            % baliteke partikularen bat, mapatik at egon daitekela, horren ondorioz
24            % hauek descartatzen ditut.
25            if Train(cont).Points(1,1) <= MaxLidar && isempty(Train(cont).Points) ~=1
26                cont = cont+1;
27            else
28                cont = cont;
29                if Nparticulas == part
30                    Train(cont).Points = Train(cont-1).Points;
31                end
32            end
33
34        end
35        % Gorde fitxategiak
36        carpeta = dir("TrainData")
37        if size(carpeta,1) == 0
38            mkdir("TrainData")
39        end
40        a = dir("TrainData/TrainData*"); % Karpeta honetan entrenamenduaren fitxategiak gordetzen ditut
41        Fitxategiak = size(a,1)+1;
42        Izena = strcat ("TrainData/TrainData",int2str(Fitxategiak))
43        save(Izena,'Train','-v7.3')
44    end

```

---

### Kodea 24 Sare egokiena aukeratu

---

```

1     function [net,errorea] = ...
2     TrainNetworks(Input,Output,Layers,Neurons,trainFcn,ErroreKopurua,Mapa,MaxLidar,Test,Minerror)
3     % Create Layers
4     errorea = [];
5     cont = 1;
6     erroreMinimoa = 10000;
7     % Conbinazioak
8     Konbinazioak = combinator(Neurons,Layers,'p','r');
9     a = zeros(1,Layers);
10    Konbinaketak = [];
11
12    for gg = 1:Neurons
13        a(1)= gg;
14        Konbinazioak = [a;Konbinazioak];
15    end
16    if Layers>2
17        for gg = 2:Layers-1
18            Konbinaketak=combinator(Neurons,gg,'p','r');
19            a = zeros(size(Konbinaketak,1),Layers-gg);
20            Konbinaketak = [Konbinaketak,a];
21            Konbinazioak = [Konbinazioak;Konbinaketak];
22        end
23    end
24    Konbinazioak;

```

---

---

```

25 for Neu=1:size(Konbinazioak,1) % Combinazio Kopurua
26     Layers=[];
27     for Lay= 1: size(Konbinazioak,2)
28         if Konbinazioak(Neu,Lay) ~= 0
29             % Create a Fitting Network
30             Layer(Lay) = Konbinazioak(Neu,Lay);
31         end
32     end
33     net = fitnet(Layer,trainFcn);
34
35 Sarearen sarrera eta irteeraren prozesamendua
36 % Choose Input and Output Pre/Post-Processing Functions
37 % For a list of all processing functions type: help nnprocess
38 net.input.processFcns = {'removeconstantrows','mapminmax'};
39 net.output.processFcns = {'removeconstantrows','mapminmax'};
40 Datuen Antolaketa
41 % Setup Division of Data for Training, Validation, Testing
42 % For a list of all data division functions type: help nndivision
43 net.divideFcn = 'dividerand'; % Divide data randomly
44 net.divideMode = 'sample'; % Divide up every sample
45 net.divideParam.trainRatio = 70/100;
46 net.divideParam.valRatio = 15/100;
47 net.divideParam.testRatio = 15/100;
48
49 % Choose a Performance Function
50 % For a list of all performance functions type: help nnperformance
51 net.performFcn = 'mse'; % Mean Squared Error
52
53 % Choose Plot Functions
54 % For a list of all plot functions type: help nnplot
55 net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
56               'plotregression','plotfit'};
57 Entrenamenduaren parametroak
58 net.trainparam.epochs=10000;
59 net.trainParam.showWindow = true;
60 Entrenatu
61 % Train the Network
62 [net,tr] = train(net,Input,Output);
63
64 Frogapena
65 error = TestNet (ErroreKopurua,Mapa,MaxLidar,net,Test);
66 errorea(cont) = error;
67
68 if erroreMinimoa > min(errorea) && cont > 1 % Errore minimoa conprobatu
69     erroreMinimoa = min(errorea);
70
71     if Test == "X" % Test arabera gorde eta ezabatu.
72         a = dir("Testeando/SareaX*");
73         if size(a,1)>1
74             izena = strcat("Testeando/",a(end).name);
75             delete(izena);
76         end
77     elseif Test == "Y"
78         a = dir("Testeando/SareaY*");
79         if size(a,1)>1
80             izena = strcat("Testeando/",a(end).name);
81             delete(izena);
82         end
83     elseif Test == "Z"
84         a = dir("Testeando/SareaZ*");
85         if size(a,1)>1
86             izena = strcat("Testeando/",a(end).name);
87             delete(izena);
88         end
89     end
90     Izena =
91     strcat("Testeando/Sarea",Test,"_Layers",int2str(Lay),"_Neurons",int2str(Neu))
92     save(Izena,'net','-v7.3')
93     elseif erroreMinimoa > min(errorea) && cont == 1
94         Izena =
95         strcat("Testeando/Sarea",Test,"_Layers",int2str(Lay),"_Neurons",int2str(Neu))
96         save(Izena,'net','-v7.3')
97     end
98     cont = cont +1;
99 end

```

---

### Kodea 25 Errorearen kalkuluaren funtzioa

---

```

1 function error = TestNet (ErroreKopurua,Mapa,MaxLidar,net,Test)
2 error = 0;
3 poses = 0;
4 TotalInput = [];
5     for testing = 1:ErroreKopurua
6         while poses == 0
7             PoseX = round(random('norm',0,1000,1,1));
8             PoseY = round(random('norm',0,40000,1,1));
9             PoseZ = round(random('norm',0,360,1,1),1);
10            [Cartesian,Correct,Points] = MedidasLIDAR(PoseX,PoseY,PoseZ,Mapa,MaxLidar);
11            if Points(1) >= MaxLidar
12                poses = 0;
13            else

```

---

```

14         poses = 1;
15     end
16 end
17 [InputAgr,Input] = cart2pol(Points(:,1),Points(:,2));
18 TotalInput = [Input;InputAgr];
19 % Estimatu
20 OUTPUT = net(TotalInput);
21 % Errorrea dagokion sareari kalkulatzeko da. X net aztertzen baldin
22 % bada, orduan Test 1 ezarriko da.
23 if Test == "X"
24     error = error + sqrt((PoseX-OUTPUT)^2);
25 elseif Test == "Y"
26     error = error + sqrt((PoseY-OUTPUT)^2);
27 elseif Test == "Z"
28     error = error + sqrt((PoseZ-OUTPUT)^2);
29 end
30 end
31 error = error/ErroreKopurua;
32 end

```

---

### Kodea 26 Segmentazio sareak emandako erantzunetik erdiko puntua atera

---

```

1 %% Navigation test
2 % Kode honekin AGV-navegazioa testeatzeko balio du.
3
4 clear all
5 close all
6 clc
7 %% Datuak prestatu.
8 % Sarea kargatu
9
10 load('Unet.mat');
11 %%
12 % Behin sarea kargatuta, video prestatzen da
13
14 %GordeLekua = 'C:\Users\dteso001\ownCloud\Tracking_DeepLearning\Image
Segmentation\Datos\IMG_7793.MOV';
15 %mkdir("Test8")
16 %karpeta = dir("Test8");
17 Tejecution = [];
18 %GordeLekua = '/Users/dani/ownCloud/Tracking_DeepLearning/Image
Segmentation/Datos/IMG_7282.MOV';
19 %obj = VideoReader(GordeLekua);
20 %Frames = obj.NumFrames;
21 %%
22 %
23 %% Frogapena
24
25 sprintf("runing")
26 %a = dir("Argazki3\*.jpg");
27 a = dir('/Users/dani/ownCloud/Tank/Argazki3/*.jpg')
28 Frames = size(a,1)
29 for i = 1:Frames
30     izena = strcat(a(i).folder,"/",a(i).name);
31     I = imread(izena);
32     %I = imresize(I,[240,240]);
33     tic
34     %C = semanticseg(I,net,"ExecutionEnvironment","gpu");
35     C = semanticseg(I,net);
36     B = labeloverlay(I,C);
37     Tejecution = [Tejecution;toc];
38     %toc;
39     Mask = C == 'Floor';
40     figure
41     imshowpair(I, Mask,'montage')
42
43 %%
44 % Irudi binario filtrazio bat aplikatu, gunea lehuntzeko. Horretarako pilota
45 % bat erabiltzen da
46
47
48 Irudia = (Mask*255);
49 [MeanPath] = PathBuilding (Irudia,1);
50 if isempty(MeanPath) == 1
51     H = Irudia;
52 else
53     H = insertMarker(Irudia,MeanPath);
54     H1 = insertMarker(I,MeanPath);
55     Calculos = [];
56     f = size(MeanPath,1);
57     for w = f-70:f
58         %Controlatzen gunea
59         Calculos(w) = (240/2-MeanPath(w,1));
60     end
61     [valor,lugar] = max(abs(Calculos));
62     H1 =insertMarker(H,[MeanPath(lugar,1),MeanPath((lugar),2)],"circle","Color","magenta");
63 %     imshow(H1)
64 end
65 figure
66 % imshowpair(H, H1,'montage')
67 arg = sprintf("Fotodd %d.jpg",i)

```

---

```

68 %izena = sprintf("C:/Users/dteso001/ownCloud/Tracking_DeepLearning/Image
Segmentation/CNN/Test3/IrudiaK_%d.jpg",i);
69 %izena = strcat(karpeta(end).folder,"\",arg)
70 %%imwrite(H1,izena,'jpg')
71 end
72 %%
73 %
74 %% Beharrezko diren funtzioak
75 % Lehengo funtzio honek lerro horizontal bakoitzaren luzeera aztertzen du, horretarako
76 % zenbat pixel txuri dauden contuan hartzen ditu.
77
78 function [MeanPaths] = PathBuilding (I,step)
79     MeanPaths = []; % Bide libre dauden erdiko puntuak identifikatzen ditu
80     for u=1:step:size(I,1)
81         a = find(I(u,:)==255);
82         if isempty(a) == 0
83             [balioa,lugar,Zenbat]= Tartea2 (a);
84             Erdia = balioa/2+Zenbat(lugar,2);
85             MeanPaths = [MeanPaths;Erdia,u];
86
87         end
88     end
89 end
90
91 %%
92 % Bidea moztuta baldin badago, akziorenbat egin behar da. Demagun bide erdian
93 % kutxa bat dagoela, horren ondorioz jakin behar da zenbat tarte dauden, eta tarte
94 % horietatik zein den hautatu behar dena
95
96
97 function [balioa,lugar,Zenbat]= Tartea2 (a)
98     Neurketa = 0;
99     Zenbat = [];
100     cont = 1;
101     if size(a,2)== 1
102         Zenbat = [1 1 1];
103     else
104         for g = 2:size(a,2)
105             if a(g-1)+1 == a(g) && g ~= size(a,2)
106                 Neurketa = Neurketa +1;
107             elseif a(g-1)+1~=a(g) || g == size(a,2)
108                 if g == size(a,2)
109                     Zenbat(cont,1) = Neurketa+1;
110                 else
111                     Zenbat(cont,1) = Neurketa;
112                 end
113                 if cont > 1
114                     Zenbat(cont,2) = Zenbat(cont-1,3);
115                     Zenbat(cont,3) = Zenbat(cont,3) + a(g);
116                 else
117                     Zenbat(cont,2) = a(1);
118                     Zenbat(cont,3) = a(g);
119                 end
120                 cont = cont+1;
121                 Neurketa = 0;
122             end
123         end
124     end
125     [balioa,lugar] = max(Zenbat(:,1));
126     balioa = balioa(1);
127     lugar = lugar(1);
128 end
129

```

---

### Kodea 27 Sare Konboluzional batek objektuak identifikatzeko

```

1 %% Object Detection Using YOLO v2 Deep Learning
2 % This example shows how to train a you only look once (YOLO) v2 object detector.
3 %
4 % Deep learning is a powerful machine learning technique that you can use to
5 % train robust object detectors. Several techniques for object detection exist,
6 % including Faster R-CNN and you only look once (YOLO) v2. This example trains
7 % a YOLO v2 vehicle detector using the |trainYOLOv2ObjectDetector| function. For
8 % more information, see <docid:vision_doccenter#bvd8yot-1 object detection using
9 % deep learning>.
10
11 doTraining = true;
12
13 %% Load Dataset
14
15 data = load('DataStorage.mat');
16 vehicleDataset = data.Location;
17 % Display first few rows of the data set.
18 vehicleDataset(1:4,:)
19 %%
20 % Split the data set into a training set for training the detector, and a test
21 % set for evaluating the detector. Select 60% of the data for training. Use the
22 % rest for evaluation.
23 %
24 % Split the data set into a training set for training the detector, and a test
25 % set for evaluating the detector. Select 60% of the data for training. Use the
26 % rest for evaluation.
27

```

---

```

28 rng(0);
29 shuffledIndices = randperm(height(vehicleDataset));
30 idx = floor(0.6 * length(shuffledIndices) );
31 trainingDataTbl = vehicleDataset(shuffledIndices(1:idx),:);
32 testDataTbl = vehicleDataset(shuffledIndices(idx+1:end),:);
33 %%
34 % Use |imageDatastore| and |boxLabelDatastore| to create datastores for loading
35 % the image and label data during training and evaluation.
36
37 imdsTrain = imageDatastore(trainingDataTbl{:,'ImgLocation'});
38 bldsTrain = boxLabelDatastore(trainingDataTbl{:,'BBOX'});
39
40 imdsTest = imageDatastore(testDataTbl{:,'ImgLocation'});
41 bldsTest = boxLabelDatastore(testDataTbl{:,'BBOX'});
42 %%
43 % Combine image and box label datastores.
44
45 trainingData = combine(imdsTrain,bldsTrain);
46 testData = combine(imdsTest,bldsTest);
47 %%
48 % Display one of the training images and box labels.
49
50 data = read(trainingData);
51 I = data{1};
52 bbox = data{2};
53 annotatedImage = insertShape(I,'Rectangle',bbox);
54 annotatedImage = imresize(annotatedImage,2);
55 figure
56 imshow(annotatedImage)
57 %% Create a YOLO v2 Object Detection Network
58
59 inputSize = [224 224 3];
60 %%
61 % Define the number of object classes to detect.
62
63 numClasses = size(vehicleDataset,2)-1;
64
65 trainingDataForEstimation = transform(trainingData,@(data)preprocessData(data,inputSize));
66 numAnchors = 7;
67 [anchorBoxes, meanIoU] = estimateAnchorBoxes(trainingDataForEstimation, numAnchors)
68 %%
69 featureExtractionNetwork = resnet50;
70 %%
71 % Select '|activation_40_relu|' as the feature extraction layer to replace the
72 % layers after '|activation_40_relu|' with the detection subnetwork.
73
74 featureLayer = 'activation_40_relu';
75 %%
76 % Create the YOLO v2 object detection network.
77
78 lgraph = yolov2Layers(inputSize,numClasses,anchorBoxes,featureExtractionNetwork,featureLayer);
79
80 %% Data Augmentation
81 augmentedTrainingData = transform(trainingData,@augmentData);
82 %%
83 % Read the same image multiple times and display the augmented training data.
84
85 % Visualize the augmented images.
86 augmentedData = cell(4,1);
87 for k = 1:4
88     data = read(augmentedTrainingData);
89     augmentedData{k} = insertShape(data{1},'Rectangle',data{2});
90     reset(augmentedTrainingData);
91 end
92 figure
93 montage(augmentedData,'BorderSize',10)
94 %% Preprocess Training Data
95 % Preprocess the augmented training data to prepare for training.
96
97 preprocessedTrainingData =
98 transform(augmentedTrainingData,@(data)preprocessData(data,inputSize));
99 %%
100 % Read the preprocessed training data.
101
102 data = read(preprocessedTrainingData);
103 %%
104 % Display the image and bounding boxes.
105
106 I = data{1};
107 bbox = data{2};
108 annotatedImage = insertShape(I,'Rectangle',bbox);
109 annotatedImage = imresize(annotatedImage,2);
110 figure
111 imshow(annotatedImage)
112 %% Train YOLO v2 Object Detector
113
114 options = trainingOptions('sgdm', ...
115     'MiniBatchSize', 16, ...
116     'InitialLearnRate', 1e-3, ...
117     'MaxEpochs', 20, ...
118     'CheckpointPath', tempdir, ...
119     'Shuffle', 'never');
120 %%
121 if doTraining
122     % Train the YOLO v2 detector.

```

```

123     [detector,info] = trainYOLOv2ObjectDetector(preprocessedTrainingData,lgraph,options);
124 else
125     % Load pretrained detector for the example.
126     pretrained = load('yolov2ResNet50VehicleExample_19b.mat');
127     detector = pretrained.detector;
128 end
129 I = imread(testDataTbl.ImgLocation{1});
130 I = imresize(I,inputSize(1:2));
131 tic
132 [bboxes,scores] = detect(detector,I);
133 toc
134 % Display the results.
135
136 I = insertObjectAnnotation(I,'rectangle',bboxes,scores);
137 figure
138 imshow(I)
139
140 preprocessedTestData = transform(testData,@(data)preprocessData(data,inputSize));
141 %%
142 % Run the detector on all the test images.
143
144 detectionResults = detect(detector, preprocessedTestData);
145 %%
146 % Evaluate the object detector using average precision metric.
147
148 [ap,recall,precision] = evaluateDetectionPrecision(detectionResults, preprocessedTestData);
149 %%
150 % The precision/recall (PR) curve highlights how precise a detector is at varying
151 % levels of recall. The ideal precision is 1 at all recall levels. The use of
152 % more data can help improve the average precision but might require more training
153 % time. Plot the PR curve.
154
155 figure
156 plot(recall,precision)
157 xlabel('Recall')
158 ylabel('Precision')
159 grid on
160 title(sprintf('Average Precision = %.2f',ap))
161 %% Functions
162 function B = augmentData(A)
163 % Apply random horizontal flipping, and random X/Y scaling. Boxes that get
164 % scaled outside the bounds are clipped if the overlap is above 0.25. Also,
165 % jitter image color.
166 B = cell(size(A));
167
168 I = A{1};
169 sz = size(I);
170 if numel(sz)==3 && sz(3) == 3
171     I = jitterColorHSV(I,...
172         'Contrast',0.2,...
173         'Hue',0,...
174         'Saturation',0.1,...
175         'Brightness',0.2);
176 end
177
178 % Randomly flip and scale image.
179 tform = randomAffine2d('XReflection',true,'Scale',[1 1.1]);
180 rout = affineOutputView(sz,tform,'BoundsStyle','CenterOutput');
181 B{1} = imwarp(I,tform,'OutputView',rout);
182
183 % Apply same transform to boxes.
184 [B{2},indices] = bboxwarp(A{2},tform,rout,'OverlapThreshold',0.25);
185 B{3} = A{3}(indices);
186
187 % Return original data only when all boxes are removed by warping.
188 if isempty(indices)
189     B = A;
190 end
191 end
192
193 function data = preprocessData(data,targetSize)
194 % Resize image and bounding boxes to the targetSize.
195 scale = targetSize(1:2)./size(data{1},[1 2]);
196 data{1} = imresize(data{1},targetSize(1:2));
197 data{2} = bboxresize(data{2},scale);
198 end
199 %% References
200 % [1] Redmon, Joseph, and Ali Farhadi. "YOLO9000: Better, Faster, Stronger."
201 % 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE,
202 % 2017.
203 %
204 % _Copyright 2019 The MathWorks, Inc._
205

```

---

### Kodea 28 Nanotec gurpilentzako Main programa

---

```

1  fDigitalIO();
2  /// Lectura de los sensores
3  Fb_Lectura();
4  F_Teclado();
5  F_Seguridad();
6  FB_CanOpenLecture(ADSNetID:= '169.254.153.119.2.1' , PortNr:= 16#1001 , CO_Index:= 16#1000 ,
7  CO_SubIndex:= 16#0, DataLength:= 10, StartReading:= TRUE , ReadData=> ,

```

---



```

8      ReadDataAvailable=> , Error=> );
9
10     CASE nCASOS OF
11
12     /// Maquina de estados
13     0: // Estado de Paro
14         IF GVL_DigitalIO.START = TRUE THEN
15             bHabil := TRUE; // Activar motores
16             /// Modos del Motor
17             GVL_Motor.OperationModeRD := 8;
18             GVL_Motor.OperationModeRI := 8;
19             GVL_Motor.SubmodeMotorRD := 1;
20             GVL_Motor.SubmodeMotorRI := 1;
21             IF bStatusRD = TRUE AND bStatusRI = TRUE THEN
22                 nCASOS := 10;
23                 bHabilito := FALSE;
24             ELSE
25                 bHabilito := TRUE;
26             END_IF
27         END_IF
28     IF GVL_DigitalIO.STOP = TRUE THEN
29         nCASOS := 0;
30         bHabil := FALSE; // Desactivar Motores
31         bHabilito := FALSE;
32         /// Modos del Motor
33         GVL_Motor.OperationModeRD := 0;
34         GVL_Motor.OperationModeRI := 0;
35         GVL_Motor.SubmodeMotorRD := 0;
36         GVL_Motor.SubmodeMotorRI := 0;
37         GVL_Seguridad.bSeguridad := FALSE;
38         GVL_Seguridad.bStopAGV := FALSE;
39     END_IF
40     // ESPERO antes de pasar a otro estado. Si no esta activo, sigo
41     mandando que se activen
42
43     Fb_SpeedControl(bHabilitar:= FALSE );
44     fMotores(bHabilitar:= bHabil , Vc:=0, Direccion:= 0, bStatusRD=>bStatusRD ,
45             bStatusRI=> bStatusRI);
46     RDITp(IN:= bHabilito, PT:=T#5S , Q=> , ET=> );
47     // Reseteo Motores
48     Fb_McResetRd(Axis:=GVL_MARGA.Eje1 , Execute:= RDITp.Q, Done=> ,
49                Busy=> , Error=> , ErrorID=> );
50     Fb_McResetRi(Axis:= GVL_MARGA.Eje2 , Execute:=RDITp.Q ,
51                Done=> , Busy=> , Error=> , ErrorID=> );
52
53     10: // Estado de START
54     IF GVL_DigitalIO.Roomba = TRUE THEN
55         nCASOS := 20;
56         Fb_SpeedControl(bHabilitar:= TRUE );
57     END_IF
58     IF GVL_DigitalIO.STOP = TRUE THEN
59         nCASos :=0;
60     END_IF
61     IF GVL_DigitalIO.bTeclado = TRUE THEN
62         nCasos := 30;
63         Fb_SpeedControl(bHabilitar:= TRUE );
64     END_IF
65     20: // Estado de Roomba
66     IF GVL_DigitalIO.Roomba = FALSE THEN
67         nCASOS :=0;
68     ELSE
69         fRoomba(Vc=> Vc, Direccion=> Direccion );
70         fMotores(bHabilitar:= TRUE, Vc:= Vc , Direccion:= -Direccion ,
71                bStatusRD=> , bStatusRI=> );
72     END_IF
73     30: // Estado Matlab
74
75     IF GVL_DigitalIO.bMatlab = FALSE THEN
76         nCASOS :=0;
77         Fb_FMovement(bHabilitar:=FALSE , speed:= 0 );
78     ELSE
79         F_Teclado(Vc=>Vc , Direccion=> Direccion );
80         fMotores( bHabilitar:= TRUE, Vc:= Vc , Direccion:= -Direccion ,
81                bStatusRD=> , bStatusRI=> );
82     END_IF
83     GVL_Matlab.bStart := BMatalb;
84     GVL_Matlab.Distacia := Dist;
85     GVL_MARGA.PoseRD := PoseRD;
86     GVL_MARGA.PoseRI := PoseRI;
87     GVL_MARGA.VelRD := VelRD;
88     GVL_MARGA.VelRI := VelRI;
89     // Calculo del desplazamiento del AGV
90
91     END_CASE

```

#### Kodea 29 Iterative Closed Point funtzioa testu egituratuan

```

1 // Reseteo de las variables
2 Ricp11 :=1.0;
3 Ricp12 :=0.0;
4 Ricp21 :=0.0;
5 Ricp22 :=1.0;

```

```

6      Ticp1      :=0.0;
7      Ticp2      :=0.0;
8      n_dError   := 100000.0;
9      n_preError  := 0.0;
10     FOR n_count := 1 TO n_maxIter BY 1 DO
11     // Buscamos cual es la mejor posicion de los puntos que ve el LIDAR, con los del EL MAPA DEL
    AGV
12     fbFindNearesPoint(
13     data1_X:= data1_X, data1_Y:= data1_Y, data2_X:= data2_X, data2_Y:= data2_Y,
14     n_index=> n_index, error=> error);
15
16     // Una vez buscaddo una posicion y estimado un error es hora de realizar el cálculo de cuanto
    esta rotado y traslado el AGV respecto de la vision del LIDAR
17     fbSVDMotionEstimation(
18     data1_X:= data1_X, data1_Y:= data1_Y, data2_X:= data2_X, data2_Y:= data2_Y,
19     n_index:= n_index, R11=> R11, R12=> R12, R21=> R21, R22=> R22, t1=> T1, t2=> T2);
20
21     // Se realizan las nuevas adaptaciones y se obtiene si la posición nueva estimada tiene un
    error menor que las otras iteraciones
22     Ricp11 := R11*Ricp11 + R21*Ricp12;
23     Ricp12 := R12*Ricp11 + R22*Ricp12;
24     Ricp21 := R11*Ricp21 + R21*Ricp22;
25     Ricp22 := R12*Ricp21 + R22*Ricp22;
26     Ticp1 := R11*Ticp1 + R12*Ticp2 + T1;
27     Ticp2 := R21*Ticp1 + R22*Ticp2 + T2;
28     n_dError := ABS(n_preError -error);
29     n_preError := error;
30     IF n_dError < n_EPS THEN // Si se ha minimizado la funcion lo suficiente entoces se
    abandona el bucle
31         EXIT;
32     END_IF
33 END_FOR

```

### Kodea 30 Bi laginketa arteko korrelazioa bilatu ST hizkuntzan

```

1     //// ESTA FUNCION ANALIZA CUAL ES EL PUNTO MÁS CERCANO EN CADA MOMENTO //////////////////////////////////
2     error := 0;
3     FOR n_count := 1 TO 500 BY 1 DO
4         // dx=(data2-repmat(data1(:,i),1,m2));
5         n_min := 10000;
6         FOR n_count1 := 1 TO 500 BY 1 DO // Funcion repmat de Matlab
7             // Comparo todos los puntos del array 2 con cada punto del array uno, para
    encontrar el mejor resultado
8             dx_X [n_count1] := data2_X[n_count1] - data1_X[n_count];
9             dx_Y [n_count1] := data2_Y[n_count1] - data1_Y[n_count];
10            dist [n_count1] := SQRT(dx_X[n_count1]*dx_X[n_count1] +
    dx_Y[n_count1]*dx_Y[n_count1]);
11            // Analizo si la distancia dada es menor que las obtenidas hasta la fecha
12            IF n_min > dist[n_count1] THEN
13                n_min := dist [n_count1];
14                n_index[n_count] := n_count1;
15            END_IF
16        END_FOR
17        error := error +n_min;
18    END_FOR

```

### Kodea 31 SVD programazioa ST hizkuntzan

```

1     //// SVD (Singular Value Decomposition ) //////////////////////////////////
2     ///Preparando los datos en función de la gestion de los punto
3     FOR n_cont := 1 TO 500 BY 1 DO
4         n_SX[n_cont] := data2_X[n_index[n_cont]];
5         n_SY[n_cont] := data2_Y[n_index[n_cont]];
6     END_FOR
7     /// REALIZAR 4 Medias una por cada X e Y de datos
8     sum1 := 0;
9     sum2 := 0;
10    sum3 := 0;
11    sum4 := 0;
12    FOR n_cont := 1 TO 500 BY 1 DO
13        sum1 := data1_X[n_cont] + sum1;
14        sum2 := data1_Y[n_cont] + sum2;
15        sum3 := n_SX[n_cont] + sum3;
16        sum4 := n_SY[n_cont] + sum4;
17    END_FOR
18    mDatal_X := sum1/500;
19    mDatal_Y := sum2/500;
20    mSX      := sum3/500;
21    mSY      := sum4/500;
22
23
24    // Reseteo la matriz W del estado anterior
25    W11 := 0;
26    W12 := 0;
27    W21 := 0;
28    W22 := 0;
29    FOR n_cont := 1 TO 500 BY 1 DO
30        // Sshifted
31        arr_SshiftedX[n_cont] := (n_SX[n_cont] - mSX);
32        arr_SshiftedY[n_cont] := (n_SY[n_cont] - mSY);

```

```

33 // Sshifted
34 arr_MshiftedX[n_cont] := (data1_X[n_cont] - mData1_X) ;
35 arr_MshiftedY[n_cont] := (data1_Y[n_cont] - mData1_Y) ;
36 // Calcular la matriz W del SVD [n_W11 , n_W12 ; n_W21 , n_W22]
37 W11 := arr_SshiftedX[n_cont] * arr_MshiftedX[n_cont]+W11;
38 W12 := arr_SshiftedX[n_cont] * arr_MshiftedY[n_cont]+W12;
39 W21 := arr_SshiftedY[n_cont] * arr_MshiftedX[n_cont]+W21;
40 W22 := arr_SshiftedY[n_cont] * arr_MshiftedY[n_cont]+W22;
41
42 END_FOR
43 /// C = W'*W
44 C11 := W11*W11 + W21*W21;
45 C12 := W11*W12 + W21*W22;
46 C21 := W11*W12 + W21*W22;
47 C22 := W12*W12 + W22*W22;
48
49 // Calculo de la Matriz V, autobalores y autovectores de una matriz SVD
50 // Ecuacion de segundo grado
51 x1 := C11/2 + C22/2 + SQRT((C11 + C22)*(C11 + C22) - 4*C11*C22 + 4*C12*C21)/2;
52 x2 := C11/2 + C22/2 - SQRT((C11 + C22)*(C11 + C22) - 4*C11*C22 + 4*C12*C21)/2;
53 // Si alguno de los valores de x1 o x2 es cero significa se puede simplificar el calculo
54 IF x1 = 0 OR x2 = 0 THEN
55 //http://ocw.uc3m.es/matematicas/mathematical-methods/homework-sheets-2/homework-sheet-
6.-full-solution-to-svd-exercises
56 V11 := 0;
57 V12 := 1;
58 V21 := 1;
59 V22 := 0;
60 S11 := SQRT(x1);
61 S12 := 0;
62 S21 := 0;
63 S22 := SQRT(x2);
64 U11 := V11;
65 U12 := V12;
66 U21 := V21;
67 U22 := V22;
68 // Rotacion
69 R11 := U11*V11 + U12*V12;
70 R12 := U21*V11 + U22*V12;
71 R21 := U11*V21 + U12*V22;
72 R22 := U21*V21 + U22*V22;
73 // Matriz de Traslacion
74 mm1 := mData1_X;
75 mm2 := mData1_Y;
76 ms1 := mSX;
77 ms2 := mSY;
78 t1 := mm1 - R11*ms1 - R12*ms2;
79 t2 := mm2 - R21*ms1 - R22*ms2;
80
81 ELSE
82 V11 := 1/SQRT((C22/2 - C11/2 + SQRT((C11 + C22)*(C11 + C22) - 4*C11*C22 +
4*C12*C21)/2)*(C22/2 - C11/2 + SQRT((C11 + C22)*(C11 + C22) - 4*C11*C22 +
4*C12*C21)/2)/C12*C12 + 1);
83 V12 := 1/SQRT((C11/2 - C22/2 + SQRT((C11 + C22)*(C11 + C22) - 4*C11*C22 +
4*C12*C21)/2)*(C11/2 - C22/2 + SQRT((C11 + C22)*(C11 + C22) - 4*C11*C22 +
4*C12*C21)/2)/C12*C12 + 1);
84 V21 := (C22/2 - C11/2 + SQRT((C11 + C22)*(C11 + C22) - 4*C11*C22 +
4*C12*C21)/2)/(C12*SQRT((C22/2 - C11/2 + SQRT((C11 + C22)*(C11 + C22) - 4*C11*C22 +
4*C12*C21)/2)*(C22/2 - C11/2 + SQRT((C11 + C22)*(C11 + C22) - 4*C11*C22 +
4*C12*C21)/2)/C12*C12 + 1));
85 V22 := -(C11 - C22 + SQRT((C11 + C22)*(C11 + C22) - 4*C11*C22 +
4*C12*C21))/(2*C12*SQRT((C11/2 - C22/2 + SQRT((C11 + C22)*(C11 + C22) - 4*C11*C22 +
4*C12*C21)/2)*(C11/2 - C22/2 + SQRT((C11 + C22)*(C11 + C22) - 4*C11*C22 +
4*C12*C21)/2)/C12*C12 + 1));
86 S11 := SQRT(2)/SQRT(SQRT(W11*W11*W11*W11 + 2*W11*W11*W12*W12 + 2*W11*W11*W21*W21 -
2*W11*W11*W22*W22 + 8*W11*W12*W21*W22 + W12*W12*W12*W12 - 2*W12*W12*W21*W21 +
2*W12*W12*W22*W22 + W21*W21*W21*W21 + 2*W21*W21*W22*W22 + W22*W22*W22*W22) + W11*W11 + W12*W12
+ W21*W21 + W22*W22);
87 S12 := 0;
88 S21 := 0;
89 S22 := SQRT(2)/SQRT(W11*W11 - SQRT(W11*W11*W11*W11 + 2*W11*W11*W12*W12 +
2*W11*W11*W21*W21 - 2*W11*W11*W22*W22 + 8*W11*W12*W21*W22 + W12*W12*W12*W12 -
2*W12*W12*W21*W21 + 2*W12*W12*W22*W22 + W21*W21*W21*W21 + 2*W21*W21*W22*W22 + W22*W22*W22*W22)
+ W12*W12 + W21*W21 + W22*W22);
90 // Matriz de Rotacion
91 R11 := V11*(S11*(V11*W11 + V21*W12) + S21*(V12*W11 + V22*W12)) + V12*(S12*(V11*W11 +
V21*W12) + S22*(V12*W11 + V22*W12));
92 R12 := V11*(S11*(V11*W21 + V21*W22) + S21*(V12*W21 + V22*W22)) + V12*(S12*(V11*W21 +
V21*W22) + S22*(V12*W21 + V22*W22));
93 R21 := V21*(S11*(V11*W11 + V21*W12) + S21*(V12*W11 + V22*W12)) + V22*(S12*(V11*W11 +
V21*W12) + S22*(V12*W11 + V22*W12));
94 R22 := V21*(S11*(V11*W21 + V21*W22) + S21*(V12*W21 + V22*W22)) + V22*(S12*(V11*W21 +
V21*W22) + S22*(V12*W21 + V22*W22));
95 // Matriz de Traslacion
96 mm1 := mData1_X;
97 mm2 := mData1_Y;
98 ms1 := mSX;
99 ms2 := mSY;
100 t1 := mm1 - ms1*(V11*(S11*(V11*W11 + V21*W12) + S21*(V12*W11 + V22*W12)) +
V12*(S12*(V11*W11 + V21*W12) + S22*(V12*W11 + V22*W12)) - ms2*(V11*(S11*(V11*W21 + V21*W22) +
S21*(V12*W21 + V22*W22)) + V12*(S12*(V11*W21 + V21*W22) + S22*(V12*W21 + V22*W22)));
101
102 t2 := mm2 - ms1*(V21*(S11*(V11*W11 + V21*W12) + S21*(V12*W11 + V22*W12)) +
V22*(S12*(V11*W11 + V21*W12) + S22*(V12*W11 + V22*W12)) - ms2*(V21*(S11*(V11*W21 + V21*W22) +
S21*(V12*W21 + V22*W22)) + V22*(S12*(V11*W21 + V21*W22) + S22*(V12*W21 + V22*W22)));
103

```

```
104
105 END_IF
```

### Kodea 32 Erregresio linealaren funtzio nagusia ST hizkuntzan

```
1  /// Orden de Ejecución
2
3  // Eq1
4  F_PreparaDatos(
5      Yant:= Yant, Yact:= Yact, Xant:= Xant, Xact:= Xact, PhiC1=> PhiC1,
6      PhiC2=> PhiC2, PhiC3=> PhiC3, PhiC4=> PhiC4, PhiF1=> PhiF1,
7      PhiF2=> PhiF2, PhiF3=> PhiF3, PhiF4=> PhiF4, Y=> Y);
8  // Eq2
9  F_Phi_tras_X_Phi(
10     PhiC1:= PhiC1 , PhiC2:= PhiC2, PhiC3:= PhiC3, PhiC4:= PhiC4,
11     PhiF1:= PhiF1, PhiF2:= PhiF2, PhiF3:= PhiF3, PhiF4:= PhiF4,
12     F1C1=> F1C1, F1C2=> F1C2, F1C3=> F1C3, F1C4=> F1C4, F2C1=> F2C1,
13     F2C2=> F2C2, F2C3=> F2C3, F2C4=> F2C4, F3C1=> F3C1, F3C2=> F3C2,
14     F3C3=> F3C3, F3C4=> F3C4, F4C1=> F4C1, F4C2=> F4C2, F4C3=> F4C3, F4C4=> F4C4);
15  // Eq3
16  F_Inversa(
17     F1C1:= F1C1, F1C2:= F1C2, F1C3:= F1C3, F1C4:= F1C4, F2C1:= F2C1,
18     F2C2:= F2C2, F2C3:= F2C3, F2C4:= F2C4, F3C1:= F3C1, F3C2:= F3C2,
19     F3C3:= F3C3, F3C4:= F3C4, F4C1:= F4C1, F4C2:= F4C2, F4C3:= F4C3,
20     F4C4:= F4C4, InvF1=> InvF1, InvF2=> InvF2, InvF3=> InvF3, InvF4=> InvF4);
21  // Eq4
22  F_Inv_X_Phi_tras(
23     InvF1:= InvF1, InvF2:= InvF2, InvF3:= InvF3, InvF4:= InvF4,
24     PhiF1:= PhiF1, PhiF2:= PhiF2, PhiF3:= PhiF3, PhiF4:= PhiF4,
25     D1Filas=> D1Filas, D2Filas=> D2Filas, D3Filas=> D3Filas, D4Filas=> D4Filas);
26  // Eq5
27  F_Resultado(
28     D1Filas:= D1Filas, D2Filas:= D2Filas, D3Filas:= D3Filas, D4Filas:= D4Filas,
29     Y:= Y, seno=> seno, coseno=> coseno, tx=> tx, ty=> ty);
```

### Kodea 33 Datuen prestaketa

```
1  (* Esta función prepara 3 matrices esenciales en la regresión lineal. La primera
2  se trata de la matriz phi(representado en columnas), la matriz phi'(Representada en filas) y
3  la segunda el vector y (aquí se alojan los valores actuales del lidar*)
4
5  gg:=1;
6  //Se usan dos variables, una para guardar el valor de la variable de salida y la otra para
7  contar todos los números del 1 al 600//
8  FOR i:=1 TO 600 DO
9      Y[gg]:=Yact[i];
10     PhiC1[gg]:=-Yant[i];
11     PhiC2[gg]:=Xant[i];
12     PhiC3[gg]:=1;
13     PhiC4[gg]:=0;
14     PhiF1[gg]:= PhiC1[gg];
15     PhiF2[gg]:= PhiC2[gg];
16     PhiF3[gg]:= PhiC3[gg];
17     PhiF4[gg]:= PhiC4[gg];
18     gg:=gg+1;
19 END_FOR
20 //Al poner gg=gg+1 el siguiente "FOR" contará desde 601 hasta 1200//
21
22 FOR i:=1 TO 600 DO
23     Y[gg]:=Xact[i];
24     PhiC1[gg]:=Xant[i];
25     PhiC2[gg]:=-Yant[i];
26     PhiC3[gg]:=0;
27     PhiC4[gg]:=1;
28     gg:=gg+1;
29 END_FOR
```

### Kodea 34 Bi matrize arteko biderketa

```
1  (*Esta función realiza la primera acción de multiplicación. Es el caso de...[phi' · phi]....*)
2  sum:=0;
3
4  // Obtener el valor de multiplicar la phi' · phi estructurado en 16 valores
5  //////////////////////////////////////F1C1
6  FOR i:=1 TO 1200 DO
7      sum:=PhiF1[i]*PhiC1[i]+sum;
8  END_FOR
9  F1C1:=sum;
10 sum:=0;
11 //////////////////////////////////////F1C2
12 FOR i:=1 TO 1200 DO
13     sum:=PhiF1[i]*PhiC2[i]+sum;
14 END_FOR
15 F1C2:=sum;
```

---

```

16 sum:=0;
17 ////////////////////////////////////////F1C3
18 FOR i:=1 TO 1200 DO
19     sum:=PhiF1[i]*PhiC3[i]+sum;
20 END_FOR
21 F1C3:=sum;
22 sum:=0;
23 ////////////////////////////////////////F1C4
24 FOR i:=1 TO 1200 DO
25     sum:=PhiF1[i]*PhiC4[i]+sum;
26 END_FOR
27 F1C4:=sum;
28 sum:=0;
29
30 ////////////////////////////////////////F2C1
31 FOR i:=1 TO 1200 DO
32     sum:=PhiF2[i]*PhiC1[i]+sum;
33 END_FOR
34 F2C1:=sum;
35 sum:=0;
36 ////////////////////////////////////////F2C2
37 FOR i:=1 TO 1200 DO
38     sum:=PhiF2[i]*PhiC2[i]+sum;
39 END_FOR
40 F2C2:=sum;
41 sum:=0;
42 ////////////////////////////////////////F2C3
43 FOR i:=1 TO 1200 DO
44     sum:=PhiF2[i]*PhiC3[i]+sum;
45 END_FOR
46 F2C3:=sum;
47 sum:=0;
48 ////////////////////////////////////////F2C4
49 FOR i:=1 TO 1200 DO
50     sum:=PhiF2[i]*PhiC4[i]+sum;
51 END_FOR
52 F2C4:=sum;
53 sum:=0;
54 ////////////////////////////////////////F3C1
55 FOR i:=1 TO 1200 DO
56     sum:=PhiF3[i]*PhiC1[i]+sum;
57 END_FOR
58 F3C1:=sum;
59 sum:=0;
60 ////////////////////////////////////////F3C2
61 FOR i:=1 TO 1200 DO
62     sum:=PhiF3[i]*PhiC2[i]+sum;
63 END_FOR
64 F3C2:=sum;
65 sum:=0;
66 ////////////////////////////////////////F3C3
67 FOR i:=1 TO 1200 DO
68     sum:=PhiF3[i]*PhiC3[i]+sum;
69 END_FOR
70 F3C3:=sum;
71 sum:=0;
72 ////////////////////////////////////////F3C4
73 FOR i:=1 TO 1200 DO
74     sum:=PhiF3[i]*PhiC4[i]+sum;
75 END_FOR
76 F3C4:=sum;
77 sum:=0;
78 ////////////////////////////////////////F4C1
79 FOR i:=1 TO 1200 DO
80     sum:=PhiF4[i]*PhiC1[i]+sum;
81 END_FOR
82 F4C1:=sum;
83 sum:=0;
84 ////////////////////////////////////////F4C2
85 FOR i:=1 TO 1200 DO
86     sum:=PhiF4[i]*PhiC2[i]+sum;
87 END_FOR
88 F4C2:=sum;
89 sum:=0;
90 ////////////////////////////////////////F4C3
91 FOR i:=1 TO 1200 DO
92     sum:=PhiF4[i]*PhiC3[i]+sum;
93 END_FOR
94 F4C3:=sum;
95 sum:=0;
96 ////////////////////////////////////////F4C4
97 FOR i:=1 TO 1200 DO
98     sum:=PhiF4[i]*PhiC4[i]+sum;
99 END_FOR
100 F4C4:=sum;
101 sum:=0;

```

---



```

F1C2*F2C4*F3C1*F4C3 + F1C2*F2C4*F3C3*F4C1 + F1C3*F2C1*F3C2*F4C4 - F1C3*F2C1*F3C4*F4C2 -
F1C3*F2C2*F3C1*F4C4 + F1C3*F2C2*F3C4*F4C1 + F1C3*F2C4*F3C1*F4C2 - F1C3*F2C4*F3C2*F4C1 -
F1C4*F2C1*F3C2*F4C3 + F1C4*F2C1*F3C3*F4C2 + F1C4*F2C2*F3C1*F4C3 - F1C4*F2C2*F3C3*F4C1 -
F1C4*F2C3*F3C1*F4C2 + F1C4*F2C3*F3C2*F4C1);
13 InvF4[1]:= -(F2C1*F3C2*F4C3 - F2C1*F3C3*F4C2 - F2C2*F3C1*F4C3 + F2C2*F3C3*F4C1 +
F2C3*F3C1*F4C2 - F2C3*F3C2*F4C1)/(F1C1*F2C2*F3C3*F4C4 - F1C1*F2C2*F3C4*F4C3 -
F1C1*F2C3*F3C2*F4C4 + F1C1*F2C3*F3C4*F4C2 + F1C1*F2C4*F3C2*F4C3 - F1C1*F2C4*F3C3*F4C2 -
F1C2*F2C1*F3C2*F4C4 + F1C2*F2C1*F3C4*F4C3 + F1C2*F2C3*F3C1*F4C4 - F1C2*F2C3*F3C4*F4C1 -
F1C2*F2C4*F3C1*F4C3 + F1C2*F2C4*F3C3*F4C1 + F1C3*F2C1*F3C2*F4C4 - F1C3*F2C1*F3C4*F4C2 -
F1C3*F2C2*F3C1*F4C4 + F1C3*F2C2*F3C4*F4C1 + F1C3*F2C4*F3C1*F4C2 - F1C3*F2C4*F3C2*F4C1 -
F1C4*F2C1*F3C2*F4C3 + F1C4*F2C1*F3C3*F4C2 + F1C4*F2C2*F3C1*F4C3 - F1C4*F2C2*F3C3*F4C1 -
F1C4*F2C3*F3C1*F4C2 + F1C4*F2C3*F3C2*F4C1);
14 InvF4[2]:= (F1C1*F3C2*F4C3 - F1C1*F3C3*F4C2 - F1C2*F3C1*F4C3 + F1C2*F3C3*F4C1 + F1C3*F3C1*F4C2
- F1C3*F3C2*F4C1)/(F1C1*F2C2*F3C3*F4C4 - F1C1*F2C2*F3C4*F4C3 - F1C1*F2C3*F3C2*F4C4 +
F1C1*F2C3*F3C4*F4C2 + F1C1*F2C4*F3C2*F4C3 - F1C1*F2C4*F3C3*F4C2 - F1C2*F2C1*F3C3*F4C4 +
F1C2*F2C1*F3C4*F4C3 + F1C2*F2C3*F3C1*F4C4 - F1C2*F2C3*F3C4*F4C1 - F1C2*F2C4*F3C1*F4C3 +
F1C2*F2C4*F3C3*F4C1 + F1C3*F2C1*F3C2*F4C4 - F1C3*F2C1*F3C4*F4C2 - F1C3*F2C2*F3C1*F4C4 +
F1C3*F2C2*F3C4*F4C1 + F1C3*F2C4*F3C1*F4C2 - F1C3*F2C4*F3C2*F4C1 - F1C4*F2C1*F3C2*F4C3 +
F1C4*F2C1*F3C3*F4C2 + F1C4*F2C2*F3C1*F4C3 - F1C4*F2C2*F3C3*F4C1 - F1C4*F2C3*F3C1*F4C2 +
F1C4*F2C3*F3C2*F4C1);
15 InvF4[3]:= -(F1C1*F2C2*F4C3 - F1C1*F2C3*F4C2 - F1C2*F2C1*F4C3 + F1C2*F2C3*F4C1 +
F1C3*F2C1*F4C2 - F1C3*F2C2*F4C1)/(F1C1*F2C2*F3C3*F4C4 - F1C1*F2C2*F3C4*F4C3 -
F1C1*F2C3*F3C2*F4C4 + F1C1*F2C3*F3C4*F4C2 + F1C1*F2C4*F3C2*F4C3 - F1C1*F2C4*F3C3*F4C2 -
F1C2*F2C1*F3C3*F4C4 + F1C2*F2C1*F3C4*F4C3 + F1C2*F2C3*F3C1*F4C4 - F1C2*F2C3*F3C4*F4C1 -
F1C2*F2C4*F3C1*F4C3 + F1C2*F2C4*F3C3*F4C1 + F1C3*F2C1*F3C2*F4C4 - F1C3*F2C1*F3C4*F4C2 -
F1C3*F2C2*F3C1*F4C4 + F1C3*F2C2*F3C4*F4C1 + F1C3*F2C4*F3C1*F4C2 - F1C3*F2C4*F3C2*F4C1 -
F1C4*F2C1*F3C2*F4C3 + F1C4*F2C1*F3C3*F4C2 + F1C4*F2C2*F3C1*F4C3 - F1C4*F2C2*F3C3*F4C1 -
F1C4*F2C3*F3C1*F4C2 + F1C4*F2C3*F3C2*F4C1);
16 InvF4[4]:= (F1C1*F2C2*F3C3 - F1C1*F2C3*F3C2 - F1C2*F2C1*F3C3 + F1C2*F2C3*F3C1 + F1C3*F2C1*F3C2
- F1C3*F2C2*F3C1)/(F1C1*F2C2*F3C3*F4C4 - F1C1*F2C2*F3C4*F4C3 - F1C1*F2C3*F3C2*F4C4 +
F1C1*F2C3*F3C4*F4C2 + F1C1*F2C4*F3C2*F4C3 - F1C1*F2C4*F3C3*F4C2 - F1C2*F2C1*F3C3*F4C4 +
F1C2*F2C1*F3C4*F4C3 + F1C2*F2C3*F3C1*F4C4 - F1C2*F2C3*F3C4*F4C1 - F1C2*F2C4*F3C1*F4C3 +
F1C2*F2C4*F3C3*F4C1 + F1C3*F2C1*F3C2*F4C4 - F1C3*F2C1*F3C4*F4C2 - F1C3*F2C2*F3C1*F4C4 +
F1C3*F2C2*F3C4*F4C1 + F1C3*F2C4*F3C1*F4C2 - F1C3*F2C4*F3C2*F4C1 - F1C4*F2C1*F3C2*F4C3 +
F1C4*F2C1*F3C3*F4C2 + F1C4*F2C2*F3C1*F4C3 - F1C4*F2C2*F3C3*F4C1 - F1C4*F2C3*F3C1*F4C2 +
F1C4*F2C3*F3C2*F4C1);

```

---

### Kodea 36 CharNum programazio Kodea.

```

1 function NUM = CharNum(Acarac,Numero)
2 % Esta funcion lee cada character de tipo ASCII y lo transforma en
3 % n'meros. Luego cada n'mero hay que multiplicar por su valor
4 % correspondiente
5 Val = str2num(char(Acarac(6*Numero:(6*Numero)+3)));
6 a = Val(1)*1000;
7 b = Val(2)*100;
8 c = Val(3)*10;
9 d = Val(4)*1;
10 NUM = (a+b+c+d)/10000;
11 end
12

```

---

### Kodea 37 Labview-tik MatLab-era informazioa trukatzeko

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% BBox Analyze %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 %%% Leer los txt generados para Ngene de Labview y transformarlos para que
3 %%% se puedan utilizar en matlab.
4 %%% Se genera la groundTruth para poder utilizarla en la red neuronal
5 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6 clear all
7 close all
8 clc
9 %%% Data reading and .mat source
10 Txt = dir('Etiquetas/*.txt'); % -> Analisis todos los archivos tipo txt
11 IMG = dir('*.png'); % -> Analisis todas las fotos tipo png
12 Aumentation = 300;
13
14 SamplesTxt = size(Txt,1);
15 SamplesIMG = size(IMG,1);
16
17 %%% From TXT to .mat
18 %Label = 'RPS'
19 Labels = [];
20 BBOX = {};
21 addpath (Txt(1).folder);
22 A = [];
23 for i = 1:SamplesTxt
24 fileID = fopen( Txt(i).name,'r'); % Leo Una imagen
25 tab = char(09); % Caracter del tabulador
26 [Acarac,count] = fscanf(fileID,['%s' tab '%f\t']); %Busco los valores en ASCII
27 fclose(fileID);
28 %DeCaracteres a Numeros
29 for w = 1:4
30 A(w) = CharNum(Acarac,w);
31 end
32 for h = 1:4
33 I = imread(IMG(i).name);
34 if rem(h,2)==0
35 Foto = size(I,1); % Ancho de la foto

```

```

36     else
37         Foto = size(I,2); % Alto de la foto
38     end
39     A(h) = A(h)*Foto;
40 end
41 % Bounding box
42 BBOX{i,1} = [A(1),A(2),A(3)-A(1),A(4)-A(2)]; % Primer punto(X,Y), Ancho, Alto
43 end
44

```

---

### Kodea 38 Irudiak aldatzen dituen programa nagusia

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% DATA AUMENTATION %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  g = 1; % Quiero data Aumentation
3  if g == 1
4      % Variables
5      NewBBOX = {};
6      cont = 1;
7      % Guarda datos
8      % Programa
9      for j= 1:Aumentation
10         b1 = random('uniform',0.6,1);
11         b2 = random('uniform',0.6,1);
12         b3 = random('uniform',0.6,1);
13         a1 = random('uniform',0.2,0.4);
14         a2 = random('uniform',0.2,0.4);
15         a3 = random('uniform',0.2,0.4);
16         %value = [.2 .3 0; .6 .7 1]
17         value = [a1 a2 a3;b1 b2 b3];
18         for h=1:SamplesIMG % 10 nuevas imagenes por cada imangen adquirida
19             I = imread(IMG(h).name);
20             text=IMG(h).name;
21             I = Intensity (I,value);
22             saveImg (I,cont,text);
23             % NewBBOX(sampleBBOX+cont,1) = BBOX{h,1}; % Guardo el BBox Correspondiente
24             cont = cont +1;
25         end
26     end
27     IMGs = dir('*.png');
28     samples = size(IMGs,1);
29     cont = 1;
30     c = 1;
31     for o=1:samples
32         NewBBOX{o,1}=BBOX{c,1};
33         cont = cont +1;
34         if cont > Aumentation +1;
35             c = c+1;
36             cont = 1;
37         end
38     end
39     BBOX = NewBBOX;
40 end
41
42

```

---

### Kodea 39 Moldatutako irudiak gorde BBox bakoitzarekin

```

1  IMG = dir('*.png');
2  ImgLocation = {};
3  ImgLocation{1,1} = strcat(IMG(1).folder,'\',IMG(1).name);
4  for t = 2: size(BBOX,1)
5      ImgLocation{t,1} = strcat(IMG(t).folder,'\',IMG(t).name);
6  end
7  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% GENERAR LA MATRIZ %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
8  gTruth = GenTruth (BBOX,ImgLocation)
9  save('gtruth','gTruth');
10 function gTruth = GenTruth (BBOX,ImgLocation)
11 % Esta funcion genera la gTruth dñde se encuentra la localizaciñ de las
12 % fotos y los bbox de cada una de las fotos.
13 % Verificar si hay BBOX con valores 0
14 BBOX_new = {};
15 ImgLocation_new = {};
16 cont = 1;
17 for p = 1:size(BBOX,1)
18     if BBOX{p,1} ~= [0 0 0 0]
19         BBOX_new{cont,1} = BBOX{p,1};
20         ImgLocation_new{cont,1} = ImgLocation{p,1};
21         cont = cont +1;
22     end
23 end
24 Table = table(ImgLocation_new,BBOX_new);
25 imageFileNames = Table.ImgLocation_new(1:end);
26 %%%% GroundTruth matrix
27 dataSource = groundTruthDataSource(imageFileNames);
28 ldc = labelDefinitionCreator();
29 addLabel(ldc,'RPS',labelType.Rectangle);
30 labelDefs = create(ldc);

```



---

```

31     labelNames = {'RPS'};
32     labelData = table(BBOX_new, 'VariableNames', labelNames);
33     gTruth = groundTruth(dataSource, labelDefs, labelData);
34 end

```

---

### Kodea 40 Combinator funtzioaren garapena

---

```

1 function [A] = combinator(N,K,s1,s2)
2 %COMBINATOR Perform basic permutation and combination samplings.
3 % COMBINATOR will return one of 4 different samplings on the set 1:N,
4 % taken K at a time. These samplings are given as follows:
5 %
6 % PERMUTATIONS WITH REPETITION/REPLACEMENT
7 % COMBINATOR(N,K,'p','r') -- N >= 1, K >= 0
8 % PERMUTATIONS WITHOUT REPETITION/REPLACEMENT
9 % COMBINATOR(N,K,'p') -- N >= 1, N >= K >= 0
10 % COMBINATIONS WITH REPETITION/REPLACEMENT
11 % COMBINATOR(N,K,'c','r') -- N >= 1, K >= 0
12 % COMBINATIONS WITHOUT REPETITION/REPLACEMENT
13 % COMBINATOR(N,K,'c') -- N >= 1, N >= K >= 0
14 %
15 % Example:
16 %
17 % To see the subset relationships, do this:
18 % combinator(4,2,'p','r') % Permutations with repetition
19 % combinator(4,2,'p') % Permutations without repetition
20 % combinator(4,2,'c','r') % Combinations with repetition
21 % combinator(4,2,'c') % Combinations without repetition
22 %
23 %
24 % If it is desired to use a set other than 1:N, simply use the output from
25 % COMBINATOR as an index into the set of interest. For example:
26 %
27 % MySet = ['a' 'b' 'c' 'd'];
28 % MySetperms = combinator(length(MySet),3,'p','r'); % Take 3 at a time.
29 % MySetperms = MySet(MySetperms)
30 %
31 %
32 % Class support for input N:
33 % float: double, single
34 % integers: int8,int16,int32
35 %
36 %
37 % Notes:
38 % All of these algorithms have the potential to create VERY large outputs.
39 % In each subfunction there is an anonymous function which can be used to
40 % calculate the number of row which will appear in the output. If a rather
41 % large output is expected, consider using an integer class to conserve
42 % memory. For example:
43 %
44 % M = combinator(int8(30),3,'p','r'); % NOT uint8(30)
45 %
46 % will take up 1/8 the memory as passing the 30 as a double. See the note
47 % below on using the MEX-File.
48 %
49 % To make your own code easier to read, the fourth argument can be any
50 % string. If the string begins with an 'r' (or 'R'), the function
51 % will be called with the replacement/repetition algorithm. If not, the
52 % string will be ignored.
53 % For instance, you could use: 'No replacement', or 'Repetition allowed'
54 % If only two inputs are used, the function will assume 'p','r'.
55 % The third argument must begin with either a 'p' or a 'c' but can be any
56 % string beyond that.
57 %
58 % The permutations with repetitions algorithm uses cumsum. So does the
59 % combinations without repetition algorithm for the special case of K=2.
60 % Unfortunately, MATLAB does not allow cumsum to work with integer classes.
61 % Thus a subfunction has been placed at the end for the case when these
62 % classes are passed. The subfunction will automatically pass the
63 % necessary matrix to the built-in cumsum when a single or double is used.
64 % When an integer class is used, the subfunction first looks to see if the
65 % accompanying MEX-File (cumsumall.cpp) has been compiled. If not,
66 % then a MATLAB For loop is used to perform the cumsumming. This is
67 % VERY slow! Therefore it is recommended to compile the MEX-File when
68 % using integer classes.
69 % The MEX-File was tested by the author using the Borland 5.5 C++ compiler.
70 %
71 % See also, perms, nchoosek, npermutek (on the FEX)
72 %
73 % Author: Matt Fig
74 % Contact: popkenai@yahoo.com
75 % Date: 5/30/2009
76 %
77 % Reference: http://mathworld.wolfram.com/BallPicking.html
78
79 ng = nargin;
80
81 if ng == 2
82     s1 = 'p';
83     s2 = 'r';
84 elseif ng == 3
85     s2 = 'n';
86 elseif ng ~= 4

```

---

---

```

87     error('Only 2, 3 or 4 inputs are allowed.  See help.')
88 end
89
90 if isempty(N) || K == 0
91     A = [];
92     return
93 elseif numel(N)~=1 || N<=0 || ~isreal(N) || floor(N) ~= N
94     error('N should be one real, positive integer. See help.')
95 elseif numel(K)~=1 || K<0 || ~isreal(K) || floor(K) ~= K
96     error('K should be one real non-negative integer. See help.')
97 end
98
99 STR = lower(s1(1)); % We are only interested in the first letter.
100
101 if ~strcmpi(s2(1),'r')
102     STR = [STR,'n'];
103 else
104     STR = [STR,'r'];
105 end
106
107 try
108     switch STR
109         case 'pr'
110             A = perms_rep(N,K);    % strings
111         case 'pn'
112             A = perms_no_rep(N,K); % permutations
113         case 'cr'
114             A = combs_rep(N,K);    % multichoose
115         case 'cn'
116             A = combs_no_rep(N,K); % choose
117         otherwise
118             error('Unknown option passed.  See help')
119     end
120 catch
121     rethrow(lasterror) % Throw error from here, not subfunction.
122     % The last error thrown should be K>N for non-replacement calls.
123 end
124
125
126
127
128 function PR = perms_rep(N,K)
129 % This is (basically) the same as npermutek found on the FEX.  It is the
130 % fastest way to calculate these (in MATLAB) that I know.
131 % pr = @(N,K) N^K; Number of rows.
132 % A speed comparison could be made with COMBN.m, found on the FEX.  This
133 % is an excellent code which uses ndgrid.  COMBN is written by Jos.
134 %
135 %     % All timings represent the best of 4 consecutive runs.
136 %     % All timings shown in subfunction notes used this configuration:
137 %     % 2007a 64-bit, Intel Xeon, win xp 64, 16 GB RAM
138 %     tic,Tc = combinator(single(9),7,'p','r');toc
139 %     %Elapsed time is 0.199397 seconds.  Allow Ctrl+T+C+R on block
140 %     tic,Tj = combn(single(1:9),7);toc
141 %     %Elapsed time is 0.934780 seconds.
142 %     isequal(Tc,Tj) % Yes
143
144 if N==1
145     PR = ones(1,K,class(N));
146     return
147 elseif K==1
148     PR = (1:N).';
149     return
150 end
151
152 CN = class(N);
153 M = double(N); % Single will give us trouble on indexing.
154 L = M^K; % This is the number of rows the outputs will have.
155 PR = zeros(L,K,CN); % Preallocation.
156 D = ones(1,N-1,CN); % Use this for cumsumming later.
157 LD = M-1; % See comment on N.
158 VL = [-(N-1) D].'; % These values will be put into PR.
159 % Now start building the matrix.
160 TMP = VL(:,ones(L/M,1,CN)); % Instead of repmatting.
161 PR(:,K) = TMP(:); % We don't need to do two these in loop.
162 PR(1:M^(K-1):L,1) = VL; % The first column is the simplest.
163 % Here we have to build the cols of PR the rest of the way.
164 for ii = K-1:-1:2
165     ROWS = 1:M^(ii-1):L; % Indices into the rows for this col.
166     TMP = VL(:,ones(length(ROWS)/(LD+1),1,CN)); % Match dimension.
167     PR(ROWS,K-ii+1) = TMP(:); % Build it up, insert values.
168 end
169
170 PR(1,:) = 1; % For proper cumsumming.
171 PR = cumsum2(PR); % This is the time hog.
172
173
174
175
176 function PN = perms_no_rep(N,K)
177 % Subfunction: permutations without replacement.
178 % Uses the algorithm in combs_no_rep as a basis, then permutes each row.
179 % pn = @(N,K) prod(1:N)/(prod(1:(N-K))); Number of rows.
180
181 if N==K
182     PN = perms_loop(N); % Call helper function.

```

---

---

```

183 % [id,id] = sort(PN(:,1)); %ok Not nec., uncomment for nice order.
184 % PN = PN(id,:); % Return values.
185     return
186 elseif K==1
187     PN = (1:N).'; % Easy case.
188     return
189 end
190
191 if K>N % Since there is no replacement, this cannot happen.
192     error(['When no repetitions are allowed, '...
193         'K must be less than or equal to N'])
194 end
195
196 M = double(N); % Single will give us trouble on indexing.
197 WV = 1:K; % Working vector.
198 lim = K; % Sets the limit for working index.
199 inc = 1; % Controls which element of WV is being worked on.
200 BC = prod(M-K+1:M); % Pre-allocation of return arg.
201 BC1 = BC / ( prod(1:K)); % Number of comb blocks.
202 PN = zeros(round(BC),K,class(N));
203 L = prod(1:K) ; % To get the size of the blocks.
204 cnt = 1+L;
205 P = perms_loop(K); % Only need to use this once.
206 PN(1:(1+L-1),:) = WV(P); % The first row.
207
208 for ii = 2:(BC1 - 1);
209     if logical((inc+lim)-N) % The logical is nec. for class single(?)
210         stp = inc; % This is where the for loop below stops.
211         flg = 0; % Used for resetting inc.
212     else
213         stp = 1;
214         flg = 1;
215     end
216
217     for jj = 1:stp
218         WV(K + jj - inc) = lim + jj; % Faster than a vector assignment!
219     end
220
221     PN(cnt:(cnt+L-1),:) = WV(P); % Assign block.
222     cnt = cnt + L; % Increment base index.
223     inc = inc*flg + 1; % Increment the counter.
224     lim = WV(K - inc + 1) ; % lim for next run.
225 end
226
227 V = (N-K+1):N; % Final vector.
228 PN(cnt:(cnt+L-1),:) = V(P); % Fill final block.
229 % The sorting below is NOT necessary. If you prefer this nice
230 % order, the next two lines can be un-commented.
231 % [id,id] = sort(PN(:,1)); %ok This is not necessary!
232 % PN = PN(id,:); % Return values.
233
234 function P = perms_loop(N)
235 % Helper function to perms_no_rep. This is basically the same as the
236 % MATLAB function perms. It has been un-recursive for a runtime of around
237 % half the recursive version found in perms.m For example:
238 %
239 %     tic, Tp = perms(1:9); toc
240 %     %Elapsed time is 0.222111 seconds. Allow Ctrl+T+C+R on block
241 %     tic, Tc = combinator(9,9, 'p'); toc
242 %     %Elapsed time is 0.143219 seconds.
243 %     isequal(Tc, Tp) % Yes
244
245 M = double(N); % Single will give us trouble on indexing.
246 P = 1; % Initializer.
247 G = cumprod(1:(M-1)); % Holds the sizes of P.
248 CN = class(N);
249
250 for n = 2:M
251     q = P;
252     m = G(n-1);
253     P = zeros(n*m,n,CN);
254     P(1:m, 1) = n;
255     P(1:m, 2:n) = q;
256     a = m + 1;
257
258     for ii = n-1:-1:1,
259         t = q;
260         t(t == ii) = n;
261         b = a + m - 1;
262         P(a:b, 1) = ii;
263         P(a:b, 2:n) = t;
264         a = b + 1;
265     end
266 end
267
268 function CR = combs_rep(N,K)
269 % Subfunction multichoose: combinations with replacement.
270 % cr = @(N,K) prod((N):(N+K-1))/(prod(1:K)); Number of rows.
271
272 M = double(N); % Single will give us trouble on indexing.
273 WV = ones(1,K,class(N)); % This is the working vector.
274 mch = prod((M:(M+K-1)) ./ (1:K)); % Pre-allocation.
275 CR = ones(round(mch),K,class(N));
276
277 for ii = 2:mch
278     if WV(K) == N

```

---

---

```

279     cnt = K-1; % Work backwards in WV.
280
281     while WV(cnt) == N
282         cnt = cnt-1; % Work backwards in WV.
283     end
284
285     WV(cnt:K) = WV(cnt) + 1; % Fill forward.
286     else
287         WV(K) = WV(K)+1; % Keep working in this group.
288     end
289
290     CR(ii,:) = WV;
291 end
292
293 function CN = combs_no_rep(N,K)
294 % Subfunction choose: combinations w/o replacement.
295 % cn = @(N,K) prod(N-K+1:N)/(prod(1:K)); Number of rows.
296 % Same output as the MATLAB function nchoosek(1:N,K), but often faster for
297 % larger N.
298 % For example:
299 %
300 %     tic,Tn = nchoosek(1:17,8);toc
301 %     %Elapsed time is 0.430216 seconds. Allow Ctrl+T+C+R on block
302 %     tic,Tc = combinator(17,8,'c');toc
303 %     %Elapsed time is 0.024438 seconds.
304 %     isequal(Tc,Tn) % Yes
305
306 if K>N
307     error(['When no repetitions are allowed, '...
308         'K must be less than or equal to N'])
309 end
310
311 M = double(N); % Single will give us trouble on indexing.
312
313 if K == 1
314     CN = (1:N)'; % These are simple cases.
315     return
316 elseif K == N
317     CN = (1:N);
318     return
319 elseif K==2 && N>2 % This is an easy case to do quickly.
320     BC = (M-1)*M / 2;
321     id1 = cumsum2((M-1):-1:2)+1;
322     CN = zeros(BC,2,class(N));
323     CN(:,2) = 1;
324     CN(1,:) = [1 2];
325     CN(id1,1) = 1;
326     CN(id1,2) = -((N-3):-1:0);
327     CN = cumsum2(CN);
328     return
329 end
330
331 WV = 1:K; % Working vector.
332 lim = K; % Sets the limit for working index.
333 inc = 1; % Controls which element of WV is being worked on.
334 BC = prod(M-K+1:M) / (prod(1:K)); % Pre-allocation.
335 CN = zeros(round(BC),K,class(N));
336 CN(1,:) = WV; % The first row.
337
338 for ii = 2:(BC - 1);
339     if logical((inc+lim)-N) % The logical is nec. for class single(?)
340         stp = inc; % This is where the for loop below stops.
341         flg = 0; % Used for resetting inc.
342     else
343         stp = 1;
344         flg = 1;
345     end
346
347     for jj = 1:stp
348         WV(K + jj - inc) = lim + jj; % Faster than a vector assignment.
349     end
350
351     CN(ii,:) = WV; % Make assignment.
352     inc = inc*flg + 1; % Increment the counter.
353     lim = WV(K - inc + 1); % lim for next run.
354 end
355
356 CN(ii+1,:) = (N-K+1):N;
357
358 function A = cumsum2(A)
359 %CUMSUM2, works with integer classes.
360 % Duplicates the action of cumsum, but for integer classes.
361 % If Matlab ever allows cumsum to work for integer classes, we can remove
362 % this.
363
364 if isfloat(A)
365     A = cumsum(A); % For single and double, use built-in.
366     return
367 else
368     try
369         A = cumsumall(A); % User has the MEX-File ready?
370     catch
371         warning('Cumsumming by loop. MEX cumsumall.cpp for speed.') %#ok
372         for ii = 2:size(A,1)
373             A(ii,:) = A(ii,:) + A(ii-1,:); % User likes it slow.
374         end

```

---

---

```
375     end
376 end
```

---

#### Kodea 41 Transfer Learning erabiliz, objektuak detektatzen dituen sarea

---

```
1 net = resnet50;
2 % Es necesario convertir la red a lgraph para poder manejar las capas.
3 lgraph = layerGraph(net);
4
5 % Eliminar las 3 últimas capas de la red.
6 layersToRemove = {
7     'fc1000'
8     'fc1000_softmax'
9     'ClassificationLayer_fc1000'
10 };
11 lgraph = removeLayers(lgraph, layersToRemove);
12
13 % Elegir el número de clases que la red va a identificar.
14 % Sólo se va a identificar un objeto -> Percha.
15 numClasses = 1;
16 numClassesPlusBackground = numClasses + 1;
17
18 % Definir las nuevas capas de clasificación.
19 newLayers = [
20     fullyConnectedLayer(numClassesPlusBackground, 'Name', 'rcnnFC')
21     softmaxLayer('Name', 'rcnnSoftmax')
22     classificationLayer('Name', 'rcnnClassification')
23 ];
24
25 % Añadir las capas de clasificación.
26 lgraph = addLayers(lgraph, newLayers);
27
28 % Conectar las capas de la red.
29 lgraph = connectLayers(lgraph, 'avg_pool', 'rcnnFC');
30
31 % Definir el número de salidas de la fully connected layer (capa totalmente conectada).
32 numOutputs = 4 * numClasses;
33
34 % Crear las capas de box regression.
35 boxRegressionLayers = [
36     fullyConnectedLayer(numOutputs, 'Name', 'rcnnBoxFC')
37     rcnnBoxRegressionLayer('Name', 'rcnnBoxDeltas')
38 ];
39
40 % Añadir las capas a la red.
41 lgraph = addLayers(lgraph, boxRegressionLayers);
42
43 % Conectar las capas de regresión a la capa 'avg_pool'.
44 lgraph = connectLayers(lgraph, 'avg_pool', 'rcnnBoxFC');
45
46 % Seleccionar la capa de extracción de características.
47 featureExtractionLayer = 'activation_40_relu';
48
49 % Desconectar las capas adjuntas a la capa de extracción de características seleccionadas.
50 lgraph = disconnectLayers(lgraph, featureExtractionLayer, 'res5a_branch2a');
51 lgraph = disconnectLayers(lgraph, featureExtractionLayer, 'res5a_branch1');
52
53 % Añadir una capa para ROI max pooling.
54 outputSize = [14 14];
55 roiPool = roiMaxPooling2dLayer(outputSize, 'Name', 'roiPool');
56 lgraph = addLayers(lgraph, roiPool);
57
58 % Conectar las capas de extracción de características a la nueva capa ROI max pooling.
59 lgraph = connectLayers(lgraph, featureExtractionLayer, 'roiPool/in');
60
61 % Conectar las salidas de ROI max pool a las capas desconectadas desde arriba.
62 lgraph = connectLayers(lgraph, 'roiPool', 'res5a_branch2a');
63 lgraph = connectLayers(lgraph, 'roiPool', 'res5a_branch1');
64
65 % Definir el ancho de las cajas.
66 anchorBoxes = [
67     44 21
68 ];
69
70 % Crear la capa region proposal layer.
71 proposalLayer = regionProposalLayer(anchorBoxes, 'Name', 'regionProposal');
72
73 lgraph = addLayers(lgraph, proposalLayer);
74
75 % Número de cjas de anclaje.
76 numAnchors = size(anchorBoxes,1);
77
78 % Número de mapas de características resultantes de la capa de extracción de características.
79 numFilters = 1024;
80
81 rpnLayers = [
82     convolution2dLayer(3, numFilters, 'padding', [1 1], 'Name', 'rpnConv3x3')
83     reluLayer('Name', 'rpnRelu')
84 ];
85
86 lgraph = addLayers(lgraph, rpnLayers);
87
88 % Conectar la RPN a la capa de extracción de características.
```

---

---

```

89 lgraph = connectLayers(lgraph, featureExtractionLayer, 'rpnConv3x3');
90
91 % Añadir las capas de clasificación RPN.
92 rpnClsLayers = [
93     convolution2dLayer(1, numAnchors*2, 'Name', 'rpnConv1x1ClsScores')
94     rpnSoftmaxLayer('Name', 'rpnSoftmax')
95     rpnClassificationLayer('Name', 'rpnClassification')
96 ];
97 lgraph = addLayers(lgraph, rpnClsLayers);
98
99 % Conectar las capas de clasificación a la red RPN.
100 lgraph = connectLayers(lgraph, 'rpnRelu', 'rpnConv1x1ClsScores');
101
102 % Añadir capas de regresión RPN.
103 rpnRegLayers = [
104     convolution2dLayer(1, numAnchors*4, 'Name', 'rpnConv1x1BoxDeltas')
105     rcnnBoxRegressionLayer('Name', 'rpnBoxDeltas');
106 ];
107
108 lgraph = addLayers(lgraph, rpnRegLayers);
109
110 % Conectar las capas de regresión a la red RPN.
111 lgraph = connectLayers(lgraph, 'rpnRelu', 'rpnConv1x1BoxDeltas');
112
113 % Conectar la RPN (Region Proposal Network).
114 lgraph = connectLayers(lgraph, 'rpnConv1x1ClsScores', 'regionProposal/scores');
115 lgraph = connectLayers(lgraph, 'rpnConv1x1BoxDeltas', 'regionProposal/boxDeltas');
116
117 % Conectar la capa region proposal al roi pooling.
118 lgraph = connectLayers(lgraph, 'regionProposal', 'roiPool/roi');
119
120 % Mostrar la red después de añadir las capas RPN.
121 figure
122 plot(lgraph)
123

```

---



# Erreferentziak

---

---

1. Vicsek, T.; Czirák, A.; Ben-Jacob, E.; Cohen, I., I; Shochet, O. Novel Type of Phase Transition in a System of Self-Driven Particles. *Physical review letters* 1995, 75, 1226-1229, DOI 10.1103/PhysRevLett.75.1226
2. Helbing, D. Traffic and related self-driven many-particle systems. *Reviews of Modern Physics* 2001, 73, 1067-1141, DOI 10.1103/RevModPhys.73.1067.
3. Fagnant, D.J.; Kockelman, K.M. The travel and environmental implications of shared autonomous vehicles, using agent-based model scenarios. *Transportation Research Part C: Emerging Technologies* 2014, 40, 1-13, DOI <https://doi.org/10.1016/j.trc.2013.12.001>.
4. Kossmann, J.; Schlosser, R. Self-driving database systems: a conceptual approach. *Distributed and Parallel Databases* 2020, DOI 10.1007/s10619-020-07288-w.
5. C. Liu; Z. Zhang; Y. Liu; Y. Si; Q. Lei Smart Self-Driving Multilevel Gate Driver for Fast Switching and Crosstalk Suppression of SiC MOSFETs. *IEEE Journal of Emerging and Selected Topics in Power Electronics* 2020, 8, 442-453, DOI 10.1109/JESTPE.2019.2947366.
6. Anonymous 2010 Book ComputerVisionECCV2010.
7. Xiaozhi Chen; Kundu, K.; Ziyu Zhang; Huimin Ma; Fidler, S.; Urtasun, R. Monocular 3D Object Detection for Autonomous Driving, *IEEE: Jun 2016;* , pp. 2147-2156.
8. Brostow, G.J.; Fauqueur, J.; Cipolla, R. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters* 2009, 30, 88-97, DOI 10.1016/j.patrec.2008.04.005.
9. A. W. M. Smeulders; M. Worring; S. Santini; A. Gupta; R. Jain Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2000, 22, 1349-1380, DOI 10.1109/34.895972.
10. A. W. M. Smeulders; M. Worring; S. Santini; A. Gupta; R. Jain Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2000, 22, 1349-1380, DOI 10.1109/34.895972.
11. Yassin, A.; Nasser, Y.; Awad, M.; Al-Dubai, A.; Ran Liu; Chau Yuen; Raulefs, R.; Aboutanios, E. Recent Advances in Indoor Localization: A Survey on Theoretical Approaches and Applications. *COMST* 2017, 19, 1327-1346, DOI 10.1109/COMST.2016.2632427.
12. Harle, R. A Survey of Indoor Inertial Positioning Systems for Pedestrians. *COMST* 2013, 15, 1281-1293, DOI 10.1109/SURV.2012.121912.00075.
13. Desouza, G.N.; Kak, A.C. Vision for mobile robot navigation: a survey. *TPAMI* 2002, 24, 237-267, DOI 10.1109/34.982903.
14. Lu, Y. Industry 4.0: A survey on technologies, applications and open research issues. *Journal of Industrial Information Integration* 2017, 6, 1-10, DOI 10.1016/j.jii.2017.04.005.
15. Martínez-Barberá, H.; Herrero-Pérez, D. Autonomous navigation of an automated guided vehicle in industrial environments. *Robotics and Computer-Integrated Manufacturing* 2010, 26, 296-311, DOI <https://doi.org/10.1016/j.rcim.2009.10.003>.
16. Vis, I.F.A. Survey of research in the design and control of automated guided vehicle systems. *European Journal of Operational Research* 2006, 170, 677-709, DOI <https://doi.org/10.1016/j.ejor.2004.09.020>
17. Tovarnov, M.S.; Bykov, N.V. A Mathematical Model of the Locomotion Mechanism of a Mobile Track Robot with the Magnetic-Tape Principle of Wall Climbing. *J Mach Manuf Reliab* 2019, 48, 250-258, DOI 10.3103/S1052618819030130.
18. Kramer, J.; Kramer, J.; Scheutz, M.; Scheutz, M. Development environments for autonomous mobile robots: A survey. *Auton Robot* 2007, 22, 101-132, DOI 10.1007/s10514-006-9013-8.
19. Bonin-Font, F.; Ortiz, A.; Oliver, G. Visual Navigation for Mobile Robots: A Survey. *J Intell Robot Syst* 2008, 53, 263-296, DOI 10.1007/s10846-008-9235-4.
20. Cao, Y.; Fukunaga, A.; Kahng, A. Cooperative Mobile Robotics: Antecedents and Directions. *Autonomous Robots* 1997, 4, 7-27, DOI 10.1023/A:1008855018923.
21. Bonin-Font, F.; Ortiz, A.; Oliver, G. Visual Navigation for Mobile Robots: A Survey. *J Intell Robot Syst* 2008, 53, 263-296, DOI 10.1007/s10846-008-9235-4.
22. Mohanan, M.G.; Salgoankar, A. A survey of robotic motion planning in dynamic environments. *Robotics and Autonomous Systems* 2018, 100, 171-185, DOI [//doi.org/10.1016/j.robot.2017.10.011](https://doi.org/10.1016/j.robot.2017.10.011)
23. Rubio, F.; Valero, F.; Llopis-Albert, C. A review of mobile robots: Concepts, methods, theoretical framework, and applications. *International Journal of Advanced Robotic Systems* 2019, 16, 172988141983959, DOI 10.1177/1729881419839596.
24. Wooten, J.; Bevely, D.; Hung, J. Piezoelectric Polymer-Based Collision Detection Sensor for Robotic Applications. *Electronics* 2015, 4, 204-220, DOI 10.3390/electronics4010204.
25. Min, J.; Ahn, K.; Park, H.; Song, J. A novel reactive-type joint torque sensor with high torsional stiffness for robot applications. *Mechatronics* 2019, 63, 102265, DOI 10.1016/j.mechatronics.2019.102265
26. Catapang, A.N.; Ramos, M. Obstacle detection using a 2D LIDAR system for an Autonomous Vehicle, *IEEE: 2016;* , pp. 441-445.
27. Wang, H.; Wang, B.; Liu, B.; Meng, X.; Yang, G. Pedestrian recognition and tracking using 3D LiDAR for autonomous vehicle. *Robotics and Autonomous Systems* 2017, 88, 71-78, DOI [//doi.org/10.1016/j.robot.2016.11.014](https://doi.org/10.1016/j.robot.2016.11.014).
28. Yin, H.; Ding, X.; Tang, L.; Wang, Y.; Xiong, R. Efficient 3D LIDAR based loop closing using deep neural network, *IEEE: Dec 2017;* , pp. 481-486.
29. Yoshitaka Fukuda; Kazuyuki Kobayashi; Kajiro Watanabe; Tetsuo Kinoshita Department of Systems Engineering; Hosei University; Tokyo LIDAR based target object classification by using reAction intensity.
30. Ray, A.K.; Gupta, M.; Behera, L.; Jamshidi, M. Sonar Based Autonomous Automatic Guided Vehicle (AGV) Navigation, *IEEE: NEW YORK; 345 E 47TH ST, NEW YORK, NY 10017 USA, 2008;* pp. +.
31. Chong, T.J.; Tang, X.J.; Leng, C.H.; Yogeswaran, M.; Ng, O.E.; Chong, Y.Z. Sensor Technologies and Simultaneous Localization and Mapping (SLAM). *Procedia Computer Science* 2015, 76, 174-179, DOI [//doi.org/10.1016/j.procs.2015.12.336](https://doi.org/10.1016/j.procs.2015.12.336).
32. Garcia-Garcia, A.; Orts-Escolano, S.; Oprea, S.; Villena-Martinez, V.; Martinez-Gonzalez, P.; Garcia-Rodriguez, J. A survey on deep learning techniques for image and video semantic segmentation. *Applied Soft Computing* 2018, 70, 41-65, DOI [//doi.org/10.1016/j.asoc.2018.05.018](https://doi.org/10.1016/j.asoc.2018.05.018).
33. Nowicki, M.R.; Wietrzykowski, J.; Skrzypczyński, P. Real-Time Visual Place Recognition for Personal Localization on a Mobile Device. *Wirel Pers Commun* 2017, 97, 213-244, DOI 10.1007/s11277-017-4502-y.



34. Elloumi, W.; Treuillet, S.; Leconge, R. Real-time camera orientation estimation based on vanishing point tracking under Manhattan World assumption. *J Real-Time Image Process* 2017, 13, 669-684, DOI 10.1007/s11554-014-0419-9.
35. Moreno, F.A.; Blanco, J.L.; Gonzalez, J. Stereo vision specific models for particle filter-based SLAM. *Robotics and Autonomous Systems* 2009, 57, 955-970, DOI //doi.org/10.1016/j.robot.2009.03.002.
36. F. Capparella; L. Freda; M. Malagnino; G. Oriolo Visual servoing of a wheeled mobile robot for intercepting a moving object, 2005 IEEE/RJS International Conference on Intelligent Robots and Systems, 2005; , pp. 2737-2743.
37. Y. Bengio; A. Courville; P. Vincent Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2013, 35, 1798-1828, DOI 10.1109/TPAMI.2013.50.
38. Y. Li; J. Wang; T. Xing; T. Liu; C. Li; K. Su TAD16K: An enhanced benchmark for autonomous driving, 2017 IEEE International Conference on Image Processing (ICIP), 2017; , pp. 2344-2348.
39. Xin, R.; Zhang, J.; Shao, Y. Complex Network Classification with Convolutional Neural Network. 2018
40. Canziani, A.; Paszke, A.; Culurciello, E. An Analysis of Deep Neural Network Models for Practical Applications. 2016
41. Chen, J.; Wang, C.; Tong, Y. AtICNet: semantic segmentation with atrous spatial pyramid pooling in image cascade network. *J Wireless Com Network* 2019, 2019, 1-7, DOI 10.1186/s13638-019-1445-x.
42. A. Loukkal; V. Fremont; Y. Grandvalet; Y. Li Improving semantic segmentation in urban scenes with a cartographic information. 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV) 2018, 400-406, DOI 10.1109/ICARCV.2018.8581165.
43. Cheng, Z.; Wang, Z.; Huang, H.; Liu, Y. Dense-ACSSD for End-to-end Traffic Scenes Recognition, *IEEE: Jun 2019*; , pp. 460-465.
44. Li, F.; Lehtomäki, M.; Oude Elberink, S.; Vosselman, G.; Kukko, A.; Puttonen, E.; Chen, Y.; Hyyppä, J. Semantic segmentation of road furniture in mobile laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing* 2019, 154, 98-113, DOI https://doi.org/10.1016/j.isprsjprs.2019.06.001.
45. Yeboah, Y.; Yanguang, C.; Wu, W.; Farisi, Z. Semantic Scene Segmentation for Indoor Robot Navigation via Deep Learning, *ACM: Aug 11, 2018*; , pp. 112-118.
46. Zhang, X.; Chen, Z.; Wu, Q.M.J.; Cai, L.; Lu, D.; Li, X. Fast Semantic Segmentation for Scene Perception. *TII* 2019, 15, 1183-1192, DOI 10.1109/TII.2018.2849348.
47. Beinschob, P.; Meyer, M.; Reinke, C.; Digani, V.; Secchi, C.; Sabattini, L. Semi-automated map creation for fast deployment of AGV fleets in modern logistics. *Robotics and Autonomous Systems* 2017, 87, 281-295, DOI //doi.org/10.1016/j.robot.2016.10.018.
48. Doan, V.; Nguyen, D.; Tran, Q.; Nguyen, D.; Le, T. Real-Time Image Semantic Segmentation Networks with Residual Depth-Wise Separable Blocks, *IEEE: Dec 2018*; , pp. 174-179.
49. Mei, J.; Chen, J.; Yao, W.; Zhao, X.; Zhao, H. Supervised Learning for Semantic Segmentation of 3D LiDAR Data, *IEEE: Jun 2019*; , pp. 1491-1498.
50. R. L. Galvez; A. A. Bandala; E. P. Dadios; R. R. P. Vicerra; J. M. Z. Maningo Object Detection Using Convolutional Neural Networks. *TENCON 2018 - 2018 IEEE Region 10 Conference 2018, 2023-2027*, DOI 10.1109/TENCON.2018.8650517.
51. A. Llopart; O. Ravn; N. A. Andersen Door and cabinet recognition using Convolutional Neural Nets and real-time method fusion for handle detection and grasping. 2017 3rd International Conference on Control, Automation and Robotics (ICCAR) 2017, 144-149, DOI 10.1109/ICCAR.2017.7942676.
52. Jain, R.; Jain, N.; Aggarwal, A.; Hemanth, D.J. Convolutional neural network based Alzheimer's disease classification from magnetic resonance brain images. *Cognitive Systems Research* 2019, 57, 147-159,
53. C. Cao; Y. Huang; Y. Yang; L. Wang; Z. Wang; T. Tan Feedback Convolutional Neural Network for Visual Localization and Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2019, 41, 1627-1640, DOI 10.1109/TPAMI.2018.2843329.
54. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection, *IEEE: Jun 2016*; , pp. 779-788.
55. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, *IEEE: Jun 2014*; , pp. 580-587.
56. Girshick, R. Fast R-CNN, *IEEE: Dec 2015*; , pp. 1440-1448.
57. Chen, L.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. 2018
58. Minaee, S.; Boykov, Y.; Porikli, F.; Plaza, A.; Kehtarnavaz, N.; Terzopoulos, D. Image Segmentation Using Deep Learning: A Survey. 2020
59. Pohlen, T.; Hermans, A.; Mathias, M.; Leibe, B. Full-Resolution Residual Networks for Semantic Segmentation in Street Scenes, *IEEE: Jul 2017*; , pp. 3309-3318.
60. Wang, H.; Sun, Y.; Liu, M. Self-Supervised Drivable Area and Road Anomaly Segmentation Using RGB-D Data For Robotic Wheelchairs. *LRA* 2019, 4, 4386-4393, DOI 10.1109/LRA.2019.2932874.
61. Sun, Y.; Zuo, W.; Liu, M. See the Future: A Semantic Segmentation Network Predicting Ego-Vehicle Trajectory With a Single Monocular Camera. *LRA* 2020, 5, 3066-3073, DOI 10.1109/LRA.2020.2975414.
62. Corke, P. *Robotics, Vision and Control: Fundamental Algorithms In MATLAB® Second, Completely Revised, Extended And Updated Edition, 2nd ed.*; Springer International Publishing: 2017;.
63. D. Fox; W. Burgard; S. Thrun The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine* 1997, 4, 23-33, DOI 10.1109/100.580977.
64. Saranrittichai, P.; Niparnan, N.; Sudsang, A. Robust local obstacle avoidance for mobile robot based on Dynamic Window approach, *IEEE: May 2013*; , pp. 1-4.
65. Z. Hong; S. Chun-Long; Z. Zi-Jun; A. Wei; Z. De-Qiang; W. Jing-Jing A Modified Dynamic Window Approach to Obstacle Avoidance Combined with Fuzzy Logic, 2015 14th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), 2015; , pp. 523-526.
66. Baehoon Choi; Beomseong Kim; Euntai Kim; Kwang Woong Yang A modified dynamic window approach in crowded indoor environment for intelligent transport robot, *IEEE: Oct 2012*; , pp. 1007-1009.
67. Zhang Hong; Sun Chun-long; Zheng Zi-jun; An Wei; Zhou De-qiang; Wu Jing-jing A Modified Dynamic Window Approach to Obstacle Avoidance Combined with Fuzzy Logic, *IEEE: Aug 2015*; , pp. 523-526.
68. J. Borenstein; Y. Koren The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation* 1991, 7, 278-288, DOI 10.1109/70.88137.

69. N. Buniyamin; W. A. J. Wan Ngah; Z. Mohamad PointsBug versus TangentBug algorithm, a performance comparison in unknown static environment, 2014 IEEE Sensors Applications Symposium (SAS), 2014; , pp. 278-282.
70. Ever, Y.K. Using simplified swarm optimization on path planning for intelligent mobile robot. *Procedia Computer Science* 2017, 120, 83-90, DOI //doi.org/10.1016/j.procs.2017.11.213.
71. Z. Mei; Y. Chen; M. Jiang; H. Wu; L. Cheng Mobile robots path planning based on dynamic movement primitives library, 2017 36th Chinese Control Conference (CCC), 2017; , pp. 6906-6911.
72. Qingjia, C.; Rongjun, D.; Bing, Z.; Xiaojian, W. Path-tracking of an autonomous vehicle via model predictive control and nonlinear filtering. *Journal of Automobile Engineering* 2018, 232, 1237-1252
73. Yu, H.; Duan, J.; Taheri, S.; Cheng, H.; Qi, Z. A model predictive control approach combined unscented Kalman filter vehicle state estimation in intelligent vehicle trajectory tracking. *Advances in Mechanical Engineering* 2015, 7, 168781401557836, DOI 10.1177/1687814015578361.
74. M. Pfeiffer; M. Schaeuble; J. Nieto; R. Siegwart; C. Cadena From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots. 2017 IEEE International Conference on Robotics and Automation (ICRA) 2017, 1527-1533, DOI 10.1109/ICRA.2017.7989182.
75. F. G. Lopez; J. Abbenseth; C. Henkel; S. Dörr A predictive online path planning and optimization approach for cooperative mobile service robot navigation in industrial applications. 2017 European Conference on Mobile Robots (ECMR) 2017, 1-6, DOI 10.1109/ECMR.2017.8098677.
76. Sathya, V.; Chinnadurai, M. Evolutionary Algorithms-Based Multi-Objective Optimal Mobile Robot Trajectory Planning. *Robotica* 2019, 37, 1363-1382, DOI 10.1017/S026357471800156X.
77. Xiaoping Fan; Xiong Luo; Sheng Yi; Shengyue Yang; Heng Zhang Optimal path planning for mobile robots based on intensified ant colony optimization algorithm. *IEEE International Conference on Robotics, Intelligent Systems and Signal Processing*, 2003. *Proceedings*. 2003 2003, 1, 131-136 vol.1, DOI 10.1109/RISSP.2003.1285562.
78. Dai, X.; Long, S.; Zhang, Z.; Gong, D. Mobile Robot Path Planning Based on Ant Colony Algorithm With A Heuristic Method. *Frontiers in neurorobotics* 2019, 13, 15, DOI 10.3389/fnbot.2019.00015.
79. H. Yang; J. Qi; Y. Miao; H. Sun; J. Li A New Robot Navigation Algorithm Based on a Double-Layer Ant Algorithm and Trajectory Optimization. *IEEE Transactions on Industrial Electronics* 2019, 66, 8557-8566, DOI 10.1109/TIE.2018.2886798.
80. Castillo, O.; Castillo, O.; Trujillo, L.; Trujillo, L.; Melin, P.; Melin, P. Multiple Objective Genetic Algorithms for Path-planning Optimization in Autonomous Mobile Robots. *Soft Comput* 2007, 11, 269-279, DOI 10.1007/s00500-006-0068-4.
81. Bayindir, L. A review of swarm robotics tasks. *Neurocomputing* 2016, 172, 292-321, DOI https://doi.org/10.1016/j.neucom.2015.05.116.
82. Achtelik, M.W.; Lynen, S.; Weiss, S.; Chli, M.; Siegwart, R. Motion- and Uncertainty-aware Path Planning for Micro Aerial Vehicles. *Journal of Field Robotics* 2014, 31, 676-698, DOI 10.1002/rob.21522.
83. Mordatch, I.; Lowrey, K.; Todorov, E. Ensemble-CIO: Full-body dynamic motion planning that transfers to physical humanoids, *IEEE: Sep 2015*; , pp. 5307-5314.
84. Aguilar, W.; Morales, S. 3D Environment Mapping Using the Kinect V2 and Path Planning Based on RRT Algorithms. *Electronics* 2016, 5, 70, DOI 10.3390/electronics5040070.
85. Botzheim, J.; Toda, Y.; Kubota, N. Bacterial memetic algorithm for offline path planning of mobile robots. *Memetic Comp* 2012, 4, 73-86, DOI 10.1007/s12293-012-0076-0.
86. Bakdi, A.; Hentout, A.; Boutami, H.; Maoudj, A.; Hachour, O.; Bouzouia, B. Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control. *Robotics and Autonomous Systems* 2017, 89, 95-109, DOI https://doi.org/10.1016/j.robot.2016.12.008.
87. I. Baturone; F. J. Moreno-Velo; S. Sanchez-Solano; A. Ollero Automatic design of fuzzy controllers for car-like autonomous robots. *IEEE Transactions on Fuzzy Systems* 2004, 12, 447-465, DOI 10.1109/TFUZZ.2004.832532.
88. Ajeil, F.H.; Ibraheem, I.K.; Sahib, M.A.; Humaidi, A.J. Multi-objective path planning of an autonomous mobile robot using hybrid PSO-MFB optimization algorithm. *Applied Soft Computing* 2020, 89, 106076, DOI https://doi.org/10.1016/j.asoc.2020.106076.
89. Zhang, Q.; Wang, P.; Chen, Z. An improved particle filter for mobile robot localization based on particle swarm optimization. *Expert Systems with Applications* 2019, 135, 181-193, DOI https://doi.org/10.1016/j.eswa.2019.06.006.
90. Chambers, A.; Scherer, S.; Yoder, L.; Jain, S.; Nuske, S.; Singh, S. Robust multi-sensor fusion for micro aerial vehicle navigation in GPS-degraded/denied environments, *American Automatic Control Council: Jun 2014*; , pp. 1892-1899.
91. Lei, T.; Liu, X.; Cai, G.; Liu, Y.; Liu, P. Pose Estimation of a Noncooperative Target Based on Monocular Visual SLAM. *International Journal of Aerospace Engineering* 2019, 2019, 1-14, DOI 10.1155/2019/9086891.
92. F. Pei; X. Wu; H. Yan Distributed SLAM system using particle swarm optimized particle filter for mobile robot navigation, 2016 IEEE International Conference on Mechatronics and Automation, 2016; , pp. 994-999.
93. Gil, A.; Reinoso, Ó; Ballesta, M.; Juliá, M. Multi-robot visual SLAM using a Rao-Blackwellised particle filter. *Robotics and Autonomous Systems* 2010, 58, 68-80, DOI //doi.org/10.1016/j.robot.2009.07.026.
94. Zhu, D.; Sun, X.; Liu, S.; Guo, P. A SLAM method to improve the safety performance of mine robot. *Safety Science* 2019, 120, 422-427, DOI https://doi.org/10.1016/j.ssci.2019.07.015.
95. Zhang, F.; Li, S.; Yuan, S.; Sun, E.; Zhao, L. Algorithms analysis of mobile robot SLAM based on Kalman and particle filter, *IEEE: Jul 2017*; , pp. 1050-1055.
96. Zhao, Y.; Wang, T.; Qin, W.; Zhang, X. Improved Rao-Blackwellised particle filter based on randomly weighted particle swarm optimization. *Computers and Electrical Engineering* 2018, 71, 477-484, DOI 10.1016/j.compeleceng.2018.07.055.
97. Welle, J.; Schulz, D.; Bachran, T.; Cremers, A.B. Optimization techniques for laser-based 3D particle filter SLAM, *IEEE: May 2010*; , pp. 3525-3530.
98. G. H. Lee; F. Fraundorfer; M. Pollefeys RS-SLAM: RANSAC sampling for visual FastSLAM, 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2011; , pp. 1655-1660.
99. Karlsson, R.; Gustafsson, F. The Future of Automotive Localization Algorithms Available, reliable, and scalable localization: Anywhere and anytime. *IEEE Signal Process Mag* 2017, 34, 60-69, DOI 10.1109/MSP.2016.2637418.
100. Campestrini, C.; Heil, T.; Kosch, S.; Jossen, A. A comparative study and review of different Kalman filters by applying an enhanced validation method. *Journal of Energy Storage* 2016, 8, 142-159, DOI //doi.org/10.1016/j.est.2016.10.004.
101. Sasaoka, T.; Kimoto, I.; Kishimoto, Y.; Takaba, K.; Nakashima, H. Multi-robot SLAM via Information Fusion Extended Kalman Filters. *IFAC-PapersOnLine* 2016, 49, 303-308, DOI //doi.org/10.1016/j.ifacol.2016.10.414.

102. A. B. S. H. M. Saman; A. H. Lotfy An implementation of SLAM with extended Kalman filter, 2016 6th International Conference on Intelligent and Advanced Systems (ICIAS), 2016; , pp. 1-4.
103. Tseng, K.; Li, J.; Chang, Y.; Yung, K.L.; Chan, C.Y.; Hsu, C. A new architecture for simultaneous localization and mapping: an application of a planetary rover. *Enterprise Information Systems* 2019, 1-17, DOI 10.1080/17517575.2019.1698772.
104. Jung, K.; Kim, J.; Kim, J.; Jung, E.; Kim, S. Positioning accuracy improvement of laser navigation using UKF and FIS. *Robotics and Autonomous Systems* 2014, 62, 1241-1247, DOI //doi.org/10.1016/j.robot.2014.03.016.
105. F. A. A. Cheein; J. M. Toibero; F. di Sciascio; R. Carelli; F. L. Pereira Monte Carlo uncertainty maps-based for mobile robot autonomous SLAM navigation, 2010 IEEE International Conference on Industrial Technology, 2010; , pp. 1433-1438.
106. Rigatos, G.; Siano, P.; Wira, P.; Busawon, K.; Binns, R. A Nonlinear H-infinity Control Approach for Autonomous Truck and Trailer Systems. *Un Sys* 2020, 08, 49-69, DOI 10.1142/S2301385020500041.
107. Rigatos, G.; Siano, P.; Wira, P.; Busawon, K.; Binns, R. A Nonlinear H-infinity Control Approach for Autonomous Truck and Trailer Systems. *Un Sys* 2020, 08, 49-69, DOI 10.1142/S2301385020500041.
108. D. Kumiawan; A. N. Jati; U. Sunarya A study of 2D indoor localization and mapping using FastSLAM 2.0, 2016 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC), 2016; , pp. 152-156.
109. Rojas, D.; Passold, F.; Osorio, R.; Cubillos, C.; Lefranc, G. Integration of Algorithms for Maps Construction and Simultaneous Localization in a Mobile Robot. *IFAC Proceedings Volumes* 2013, 46, 129-134, DOI //doi.org/10.3182/20130911-3-BR-3021.00098.
110. Montemerlo, M.; Thrun, S. Simultaneous localization and mapping with unknown data association using FastSLAM, *IEEE: 2003; , pp. 1985-1991 vol.2.*
111. Havangi, R.; Taghirad, H.D.; Nekoui, M.A.; Teshnehlab, M. A Square Root Unscented FastSLAM With Improved Proposal Distribution and Resampling. *TIE* 2014, 61, 2334-2345, DOI 10.1109/TIE.2013.2270211.
112. Wang, J.; Takahashi, Y. Particle Smoother-Based Landmark Mapping for the SLAM Method of an Indoor Mobile Robot with a Non-Gaussian Detection Model. *Journal of Sensors* 2019, 2019, 1-19, DOI 10.1155/2019/3717298.
113. Wang, Y.; Zhang, W.; An, P. A survey of simultaneous localization and mapping on unstructured lunar complex environment. *AIP Conference Proceedings* 2017, 1890, DOI 10.1063/1.5005198.
114. Carrera Villacres, J.L.; Zhao, Z.; Braun, T.; Li, Z. A Particle Filter-Based Reinforcement Learning Approach for Reliable Wireless Indoor Positioning. *J-SAC* 2019, 37, 2457-2473, DOI 10.1109/JSAC.2019.2933886.
115. G. Grisetti; C. Stachniss; W. Burgard Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters. *IEEE Transactions on Robotics* 2007, 23, 34-46, DOI 10.1109/TRO.2006.889486.
116. Raghu, S.; Sriraam, N.; Temel, Y.; Rao, S.V.; Kubben, P.L. EEG based multi-class seizure type classification using convolutional neural network and transfer learning. *Neural Networks* 2020, 124, 202-212, DOI https://doi.org/10.1016/j.neunet.2020.01.017.
117. Jo, K.; Kim, C.; Sunwoo, M. Simultaneous Localization and Map Change Update for the High Definition Map-Based Autonomous Driving Car. *Sensors (Basel, Switzerland)* 2018, 18, 3145, DOI 10.3390/s18093145.
118. Klemm, S.; Essinger, M.; Oberlander, J.; Zofka, M.R.; Kuhnt, F.; Weber, M.; Kohlhaas, R.; Kohs, A.; Roennau, A.; Schamm, T.; Zollner, J.M. Autonomous multi-story navigation for valet parking, *IEEE: November 2016; , pp. 1126-1133.*
119. Mutz, F.; Veronese, L.P.; Oliveira-Santos, T.; de Aguiar, E.; Auat Cheein, F.A.; Ferreira De Souza, A. Large-scale mapping in complex field scenarios using an autonomous car. *Expert Systems With Applications* 2016, 46, 439-462, DOI 10.1016/j.eswa.2015.10.045.
120. Thrun, S.; Montemerlo, M. The Graph SLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures. *The International Journal of Robotics Research* 2006, 25, 403-429, DOI 10.1177/0278364906065387.
121. L. Chen; S. Wang; H. Hu Pose-based GraphSLAM algorithm for robotic fish with a mechanical scanning sonar. 2013 IEEE International Conference on Robotics and Biomimetics (ROBIO) 2013, 38-43, DOI 10.1109/ROBIO.2013.6739432.
122. Wang, Y.; Xiong, R.; Huang, S. A pose pruning driven solution to pose feature GraphSLAM. *Adv Rob* 2015, 29, 683-698, DOI 10.1080/01691864.2014.998707.
123. F. Schuster; C. G. Keller; M. Rapp; M. Haueis; C. Curio Landmark based radar SLAM using graph optimization. 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC) 2016, 2559-2564, DOI 10.1109/ITSC.2016.7795967.
124. J. Huang; D. Millman; M. Quigley; D. Stavens; S. Thrun; A. Aggarwal Efficient, generalized indoor WiFi GraphSLAM. 2011 IEEE International Conference on Robotics and Automation 2011, 1038-1043, DOI 10.1109/ICRA.2011.5979643.
125. Y. Gao; D. Lin; J. Tian; C. Zou Based on Nonlinear Optimization and Keyframes Dense Mapping Method for RGB-D SLAM System, 2018 37th Chinese Control Conference (CCC), 2018; , pp. 5196-5200.
126. Kim, P.; Chen, J.; Cho, Y.K. SLAM-driven robotic mapping and registration of 3D point clouds. *Automation in Construction* 2018, 89, 38-48, DOI //doi.org/10.1016/j.autcon.2018.01.009.
127. Geun-Mo Lee; Ju-Hwan Lee; Soon-Yong Park Calibration of VLP-16 Lidar and multi-view cameras using a ball for 360 degree 3D color map acquisition, *IEEE: Nov 2017; , pp. 64-69.*
128. Qiao Wu; Kai Sun; Wenjun Zhang; Chaobing Huang; Xiaochun Wu Visual and LiDAR-based for the mobile 3D mapping, *IEEE: Dec 2016; , pp. 1522-1527.*
129. Tian, G.; Liu, Y.; Liu, L.; Ri, J.; Sun, Y. ObjectFusion: An object detection and segmentation framework with RGB-D SLAM and convolutional neural networks. *Neurocomputing* 2019, 345, 3-14, DOI 10.1016/j.neucom.2019.01.088.
130. Luo, Q.; Ma, H.; Tang, L.; Wang, Y.; Xiong, R. 3D-SSD: Learning hierarchical features from RGB-D images for amodal 3D object detection. *Neurocomputing* 2020, 378, 364-374, DOI https://doi.org/10.1016/j.neucom.2019.10.025. Available online: <http://www.sciencedirect.com/science/article/pii/S0925231219314018>.
131. J. Mullane; B. Vo; M. D. Adams; B. Vo A Random-Finite-Set Approach to Bayesian SLAM. *IEEE Transactions on Robotics* 2011, 27, 268-282, DOI 10.1109/TRO.2010.2101370.
132. Lin, L.; Song, Y.; Yang, Y.; Feng, H.; Cheng, Y.; Pan, H. Computer vision system R&D for EAST Articulated Maintenance Arm robot. *Fusion Engineering and Design* 2015, 100, 254-259, DOI https://doi.org/10.1016/j.fusengdes.2015.06.017.
133. Schreiber, D.; Schreiber, D.; Cambrini, L.; Cambrini, L.; Biber, J.; Biber, J.; Sardy, B.; Sardy, B. Online visual quality inspection for weld seams. *Int J Adv Manuf Technol* 2009, 42, 497-504, DOI 10.1007/s00170-008-1605-3.
134. Casadio, M.; Giannoni, P.; Morasso, P.; Sanguineti, V. A proof of concept study for the integration of robot therapy with physiotherapy in the treatment of stroke patients. *Clinical Rehabilitation* 2009, 23, 217-228, DOI 10.1177/0269215508096759.
135. U. Frese; B. Bauml; S. Haidacher; G. Schreiber; I. Schaefer; M. Hahnle; G. Hirzinger Off-the-shelf vision for a robotic ball catcher. *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)* 2001, 3, 1623-1629 vol.3, DOI 10.1109/IROS.2001.977211.

136. Villani, V.; Pini, F.; Leali, F.; Secchi, C. Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics* 2018, 55, 248–266, DOI <https://doi.org/10.1016/j.mechatronics.2018.02.009>.
137. Teso-Fz-Betoño, D.; Zulueta, E.; Fernandez-Gamiz, U.; Aramendia, I.; Uriarte, I. A Free Navigation of an AGV to a Non-Static Target with Obstacle Avoidance. *Electronics* 2019, 8, 159, DOI [10.3390/electronics8020159](https://doi.org/10.3390/electronics8020159).
138. Teso-Fz-Betono, D.; Zulueta, E.; Fernandez-Gamiz, U.; Saenz-Aguirre, A.; Martinez, R. Predictive Dynamic Window Approach Development with Artificial Neural Fuzzy Inference Improvement. *Electronics* 2019, 8, 935.
139. Teso-Fz-Betoño, D.; Zulueta, E.; Sánchez-Chica, A.; Fernandez-Gamiz, U.; Saenz-Aguirre, A. Semantic Segmentation to Develop an Indoor Navigation System for an Autonomous Mobile Robot. *Mathematics* 2020, 8, 855, DOI [10.3390/math8050855](https://doi.org/10.3390/math8050855).
140. Teso-Fz-Betoño, D.; Zulueta, E.; Sánchez-Chica, A.; Fernandez-Gamiz, U.; Uriarte, I.; Lopez-Guede, J. A Relative Positioning Development for an Autonomous Mobile Robot with a Linear Regression Technique, 15th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 202, , 0); Herrero, Á, Cambra, C., Urda, D., Sedano, J., Quintián, H. and Corchado, E., Eds.; Springer International Publishing: Cham, 2021, , pp. 627–635.
141. Saenz-Aguirre, A.; Zulueta, E.; Fernandez-Gamiz, U.; Ulazia, A.; Teso-Fz-Betono, D. Performance enhancement of the artificial neural network–based reinforcement learning for wind turbine yaw control. *Wind Energy* 2020, 23, 676–690, DOI [10.1002/we.2451](https://doi.org/10.1002/we.2451).
142. Iñigo Aramendia; Unai Fernandez-Gamiz; Ekaitz Zulueta; Aitor Saenz-Aguirre; Daniel Teso-Fz-Betoño Parametric Study of a Gurney Flap Implementation in a DU91W(2)250 Airfoil. *Energies* 2019, 12, 294, DOI [10.3390/en12020294](https://doi.org/10.3390/en12020294).
143. Aramendia, I.; Zulueta, E.; Teso-Fz-Betoño, D.; Saenz-Aguirre, A.; Fernandez-Gamiz, U. Modeling of Motorized Orthosis Control. *Applied Sciences* 2019, 9, 2453, DOI [10.3390/app9122453](https://doi.org/10.3390/app9122453).
144. Sánchez-Chica, A.; Zulueta, E.; Teso-Fz-Betoño, D.; Martínez-Filgueira, P.; Fernandez-Gamiz, U. ANN-Based Stop Criteria for a Genetic Algorithm Applied to Air Impingement Design. *Energies* 2019, 13, 16, DOI [10.3390/en13010016](https://doi.org/10.3390/en13010016).
145. Saenz-Aguirre, A.; Fernandez-Gamiz, U.; Zulueta, E.; Aramendia, I.; Teso-Fz-Betono, D. Flow control based 5 MW wind turbine enhanced energy production for hydrogen generation cost reduction. *International Journal of Hydrogen Energy* 2020, DOI <https://doi.org/10.1016/j.ijhydene.2020.01.022>
146. Saenz-Aguirre, A.; Zulueta, E.; Fernandez-Gamiz, U.; Teso-Fz-Betoño, D.; Olarte, J. Kharitonov Theorem Based Robust Stability Analysis of a Wind Turbine Pitch Control System. *Mathematics (Basel)* 2020, 8, 964, DOI [10.3390/math8060964](https://doi.org/10.3390/math8060964). Available online: <https://doi.org/10.3390/math8060964>.

## Webguneko Erreferentziak

- {1} Mercedes – Benz Gasiteizko Lantegiaren historia, patrimonio industrial vasco, 2020: <http://www.patrimonioindustrialvasco.com/actividades/imosa-industrias-del-motor-s-a-de-vitoria-gasteiz-y-la-furgoneta-f-1000-dkw/>
- {2} Labview Pattern Matching description, National Instrument web, 2020: [http://zone.ni.com/reference/en-XX/help/370281AD-01/nivisionconcepts/pattern\\_matching\\_techniques/](http://zone.ni.com/reference/en-XX/help/370281AD-01/nivisionconcepts/pattern_matching_techniques/)
- {3} Labview Geometric Matching description, National Instrument web, 2020: [http://zone.ni.com/reference/en-XX/help/370281AD-01/nivisionconcepts/geometric\\_matching\\_technique/](http://zone.ni.com/reference/en-XX/help/370281AD-01/nivisionconcepts/geometric_matching_technique/)
- {4} Pressure Profile Systems empresa, fabrikatearen webgunea, 2020: <https://pressureprofile.com>
- {5} Euro Herramientas, tala bidezko bihurkinen iragarpena, 2020: <https://euroherramientas.com/universal-robots-estic-colaborativos/>
- {6} The Art Of The Ergonomics, Atlas Copco, 2020: [https://www.atlascopco.com/content/dam/atlas-copco/industrial-technique/general/documents/pocketguides/Pocket%20guide\\_the%20art%20of%20ergonomics.pdf](https://www.atlascopco.com/content/dam/atlas-copco/industrial-technique/general/documents/pocketguides/Pocket%20guide_the%20art%20of%20ergonomics.pdf)
- {7} Estic Pulse Tools, ESTIC, 2020: [http://www.estic.co.jp/products/product\\_en.php#Handheld%20Nutrunner](http://www.estic.co.jp/products/product_en.php#Handheld%20Nutrunner)
- {8} Beckhoff, enpresaren webgunea, 2020: <https://www.beckhoff.com>
- {9} Visita Insitucional Mercedes Vitoria, El correo, 2019: <https://www.elcorreo.com/alava/araba/posicion-estrategica-liderazgo-20190214133341-nt.html>
- {10} Visita Institucional a la factoría de Mercedes-Benz en Vitoria-Gasteiz, Ayuntamiento de Vitoria-Gasteiz, 2019: <https://www.youtube.com/watch?v=GiFbG-k3Xho>
- {11} Fotografias de la portada de la tesis, Mercedes-Benz Media, 2020: <https://media.daimler.com/marsMediaSite/en/instance/picture/Mercedes-Benz-EQV.xhtml?oid=44173065>

Proiektu honek bi prototipo ezberdin jasotzen ditu. Alde batetik, nabigazio autonomoa erabiltzen duen AMR prototipo baten garapena erakutsiko da. Bestetik, Mercedes – Benz barnean landuriko tresneria baten ikerketa aurkeztuko da. AMR-ak plataforma mugikor ahaltsuak dira eta hauek barneko nabigazio autonomoa erabiltzen dute, edozein gune ezagunetik mugiarazteko. Horregatik, Gasteizko Ingeniaritza Eskolak halako plataforma baten diseinua burutzen hasi da, lokalizazio algoritmoak lantzeko. Robot mugikor honek elementu industrialak erabiliko ditu eta hauek inteligentzia garapenean zenbait oztopo ezarriko ditu. AMR - ri robot bat atxiki ahal zaio, horregatik Mercedes – Benz barnean elementu komertzialekin AMR bateri robot kolaboratibo bat ezarri zaio. Garapen honek lan postuen efizientziak lantzeko baliagarria izango da eta horretarako robotak etengabeko lanetan mugiarazi, kokatu eta kalitatezko lana burutu behar du.

Autonomoa

Kalitatea

Eraginkorra

Aldakortasuna

Ergonomia

Etengabeko-lana

Industria 4.0

Segurtasuna

Produktibitatea

Kolaboratiboa

Eskuragarria

