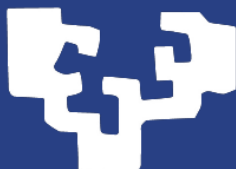


eman ta zabal zazu



Universidad
del País Vasco

Euskal Herriko
Unibertsitatea

Ibilgailu Autonomoen Kontrolerako Ikaskuntza Sakona

Mireya Cabezas Olivenza

2023



Ibilgailu autonomoen kontrolerako ikaskuntza sakona

Tesiaren autorea:

Mireya Cabezas Olivenza

Tesiaren zuzendaria:

Ekaitz Zulueta Guerrero

Doktorego programa:

Kontrol Ingeniaritza, Automatizazioa eta Robotika Doktoregoa

Aurkezpen urtea:

2023



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

LABURPENA

Ikaskuntza sakonak aurrerapauso handia suposatu du ibilgailu autonomoen munduan. Sare neuronalen garatzeak robot hauen sistemen hobekuntza bultzatu du. Beste alde batetik, ikusmen artifiziala, duen praktikotasun eta sinpletasunagatik nabarmentzen da, merkea izateaz gain, jasotako informazioa oso baliagarri bihurtzen duelako. Ondorioz, teknika asko garatu dira ikaskuntza sakona eta ikusmena konbinatzen dituztenak ibilgailu autonomoetan aplikatzeko. Alabaina, ikaskuntza sakoneko alor askotan optimizazioaren falta existitzen jarraitzen du. Barnealdeko ibilgailuen kontrolean fokua jarriz, ikerketa honek autonomoa den nabigazio sistema bat planteatzen du. Helburu horretarako ikaskuntza sakoneko tekniken optimizazioa egiten da eta ikusmen artifizialeko teknikak erabiltzen dira. Nabigazio algoritmoa hiru moduluz osatuta dago, hauek ibilbide plangintza eta kontsigna lorketa, aldibereko lokalizazio eta mapatzea, eta segurtasunerako oztopoen errekonozimendua izanik. Beraz, algoritmoa ingurune batean ibilbide bat eraikitzeke gai izan behar da, aldiune bakoitzean eskuratu beharreko posizioa ezagutzuz. Ingurune hori ezezaguna izango denez bertako mapa bat eraiki beharko da, lokalizatzeko ahalmena mantenduz. Bertan oztopoak egotea ohikoa izango denez, hauek detektatu beharko dira segurtasuna mantenduko duten ekintzak erabakitzeke.

Hitz-gakoak: *Ikaskuntza Sakona, Ikusmen Artifiziala, Sare Neuronal, Errefortzu Bidezko Ikaskuntza, Algoritmo Genetikoa, Optimizazioa, Barneko Nabigazio Autonomoa, Ibilgailu Autonomoa*

ABSTRACT

Deep learning has been a major breakthrough in the world of autonomous vehicles. The development of neural networks has boosted the improvement of these robots' systems. Machine vision, on the other hand, stands out for its practicality and simplicity, which, in addition to being inexpensive, makes the information gathered highly useful. Consequently, numerous techniques combining deep learning and machine vision have been developed for application in autonomous vehicles. However, there is still a lack of optimisation in many areas of deep learning. Focusing on indoor vehicle control, this research proposes an autonomous navigation system. For this purpose, an optimisation of deep learning techniques is performed, and computer vision techniques are used. The navigation algorithm is composed of three modules, namely trajectory planning and target achievement, simultaneous localisation and mapping, and obstacle recognition for safety. Therefore, the algorithm must be able to construct a trajectory in an environment, recognising the position to be reached at each moment. Since this environment will be unknown, a map of the environment must be constructed, while maintaining localisation capability. Since the presence of obstacles in the environment will be common, it will be necessary to detect them in order to decide on actions to maintain safety.

Keywords: *Deep Learning, Computer Vision, Neural Network, Reinforcement Learning, Genetic Algorithm, Optimisation, Autonomous Indoor Navigation, Autonomous Vehicle, Autonomous Vehicle*

RESUMEN

El aprendizaje profundo ha supuesto un gran avance en el mundo de los vehículos autónomos. El desarrollo de redes neuronales ha impulsado la mejora de los sistemas de estos robots. Por otra parte, la visión artificial, destaca por su practicidad y simplicidad, que además de ser barata, da elevada utilidad a la información recogida. En consecuencia, se han desarrollado numerosas técnicas que combinan el aprendizaje profundo y la visión artificial para ser aplicadas en vehículos autónomos. Sin embargo, sigue existiendo una falta de optimización en muchas áreas del aprendizaje profundo. Centrándose en el control de vehículos de interiores, esta investigación plantea un sistema de navegación autónoma. Para tal objetivo, se realiza una optimización de técnicas de aprendizaje profundo y se utilizan técnicas de visión artificial. El algoritmo de navegación se compone de tres módulos siendo estos la planificación de trayectorias y logro de consignas, la localización y mapeo simultaneo, y el reconocimiento de obstáculos para la seguridad. Por lo tanto, el algoritmo debe ser capaz de construir una trayectoria en un entorno, reconociendo la posición a alcanzar en cada momento. Dado que este entorno será desconocido, deberá de construirse un mapa del mismo, manteniendo la capacidad de localización. Dado que la presencia de obstáculos en el lugar será habitual, será necesario detectarlos para decidir acciones que mantengan la seguridad.

Palabras clave: *Aprendizaje Profundo, Visión Artificial, Red Neuronal, Aprendizaje Por Refuerzo, Algoritmo Genético, Optimización, Navegación Autónoma Interior, Vehículo Autónomo*

ESKER ONAK

Tesi honek iraun duen denboran zehar gorabehera ezberdinetatik pasa izan nahiz. Zorionez, pertsona asko egon dira nire ondoan, adorea emanaz eta indarberritzen. Nahiz eta ni maitekorra ez izan eta askotan ez dudan adierazi, esker anitz eman nahi diet pertsona horietako bakoitzari. Guztiengatik da lan hau egina dagoela; zuen hitzek bihotz eman didate.

Lehenik eta behin, Vitoria-Gasteizeko ingeniarietza eskolari eskerrak, doktoretza ikasketak egiteko aukera emateagatik. Partikularki Ekaitz Zulueta-ri, nire zuzendaria izateagatik. Berak esaten duen bezala, nire buruko "martzianitoei" forma ematen lagundu didalako. Bidaia akademiko neketsu baina atsegingarri honetan eman didan aparteko orientazioagatik. Bere dedikazioa, ezagutza eta pazientzia amaigabea funtsezkoak izan dira. Mila esker ere esploratzeko askatasuna emateagatik eta, aldi berean, fokotuta egoteko behar den gida izateagatik. Giza sostengua emateagatik eta ni baloratzeagatik nik egiten ez nuenean, nigan sinesteagatik. Zin dagizut positiboagoa izaten ikasiko dudala. Esaten dit eta baieztatu egin dezaket, orain lagunak gara eta Urkulura pasio asko ditugu emateke.

Nire gurasoei ere, eskerrik asko. Egunero ematen didaten maitasunagatik, nire bizitza akademikoko etapa guztietan argia izan dena. Etengabeko laguntzagatik, jakinduriagatik, aholkuengatik eta beti alboan egoteagatik. Nire emozio desberdinak jasateagatik eta beti hauek ulertzeagatik. Amari, txikitatik daukadan balioa behin eta berriz gogora ekartzeagatik, bere esaldi positiboengatik. Entzuteagatik eta beti irtenbide bat emateagatik, proposatzen naizena lortzeko gai naizela gogoratzeagatik. Aitari, jakin-nahia izaten erakusteagatik. Proposatzen baduzu, ezer ez dela ezinezkoa irakasteagatik. Nahiz eta ez ulertu, tesi honi buruz kontatzen nizekin kontu guztiei atentzia jartzeagatik. "Orrialde bat gehiago, lauza bat gehiago". Inoiz trabak ez jartzeagatik eta babesteagatik. Asko maite zaituztet, ezinezkoa da guraso hobeagoak izatea.

Nire ahizpari, Leire, bizi osoan zehar erakutsi didazulako oztupoak ez direla arazo bat eta beti dagoela beste forma bat horiek saihesteko. Egiten jarraitzen duzulako. Poztasun iturri izateagatik eta beti irribarre bat niretzako izateagatik. Nahiz eta gaiari buruz hitz egiten dudan bakoiztean oso astuna naizela esaten didazun, mila esker ere beti edonora nirekin etortzeko prest egoteagatik eta uneoro bertan egoteagatik. Nire "txitxo", beti izango nauzu alboan, osatu egiten nauzu.

Gainontzeko familiari, arnas-hitzengatik eta indarrarengatik.

Lagunei, nire tesiko prozesuan jarri duten interesagatik, entzuteagatik eta ni ulertzen saiatzeagatik. Baita ere aldi berean beraien doktoretzak egiten jardun duten kideei, haien kolaborazio eta elkarriketak nire perspektiba aldatzen eta ideia berrien ekarpenean lagundu dutelako.

Azkenik, ni neuri, proiektuan jarri dudan dedikazio, pertseberantzia eta esfortzu etengabeagatik. Proiektu honi esker gauza asko konfirmatu diot neure buruari. Erresistentzia daukat. Nire barneborrokkak gainditu ahal izan ditudalako eta inoiz amore eman ez dudalako. Burugogorra izateagatik eta gauzak bukatzea nahi izanagatik. Sentibera izateagatik eta azkenean dena iristen dela ulertzeagatik. Lan hau nahi dudana lortzeko ahalmena dudanaren oroigarri izan dadila.

Beste behin, mila esker, hau gure lana da.

AURKIBIDEA

ATALBURUA 1.....	1
1. Hitzaurrea	3
1.1 Gaur egungo testuingurua	4
1.2 Helburuak eta irismena.....	5
1.3 Memoriaren egituraketa.....	7
ATALBURUA 2.....	9
1. Gidatze Autonomia	11
1.4 Ibilgailu autonomoak industriaren barne	15
2. Adimen Artifiziala Gidatze Autonomoan	19
2.1. Entrenamendu datuen garrantzia.....	20
2.1.1. IKASKUNTZA TRANSFERITUA.....	20
2.2. Ikaskuntza automatikoaren teknikaren erabilera.....	22
2.2.1. GAINBEGIRATUTAKO IKASKUNTZA	23
2.2.2. GAINBEGIRATU GABEKO IKASKUNTZA.....	23
2.2.3. ERREFORTZU BIDEZKO IKASKUNTZA.....	23
2.2.4. IKASKUNTZA SAKONA	29
Deep Neural Network arkitektura	29
Feedforward arkitektura.....	30
Autoencoder arkitektura.....	33
Convolutional Neural Network arkitektura	36
Recurrent Neural Network arkitektura.....	42
2.3. Konputazio ebolutiboaren teknikaren erabilera.....	48
2.3.1. ALGORITMO GENETIKOAK	49
2.4. Fuzzy logikaren teknikaren erabilera	56
3. Ikusmen Artifiziala Gidatze Autonomoan	62
3.1. Datu-baseak eta erreferentzia-puntuak	62
3.2. Patroi fiduziaroak	67
3.2.1. MARKATZAILE FIDUZIARIOAK	67
4. Beste Teknika Motak Gidatze Autonomoan	71
ATALBURUA 3.....	75
1. Ibilgailu Autonomoetako Erronkak	77
1.1. Lan honetan jorratutako gabeziak	80
ATALBURUA 4.....	83
1. Erabilitako Ibilgailu Autonomia	85

1.1.	AMR-aren konfigurazioa	85
1.2.	AMR-aren programazioa	86
2.	Errefortzu Bidezko Ikaskuntzaren Optimizazioa	88
2.1.	Optimizazio analitikoa	89
2.2.	Eboluzio bidezko optimizazioa	98
2.2.1.	ALGORITMO GENETIKO BIDEZKO OPTIMIZAZIOA	102
3.	SLAM Bisuala	105
3.1.	Kameraren kalibrazioa	106
3.2.	Markatzaileen detekzioa	108
3.2.1.	MARKATZAILE ABSOLUTUAK	108
3.2.2.	MARKATZAILE ERLATIBOAK	111
3.2.3.	MAPAREN ERAIKITZEA	114
4.	Oztopoen Distantzien Kalkulua	118
4.1.	Datu-basearen sortzea	118
4.1.1.	DATUEN LEHUNTZEA ETA IRUDIEN EGOKITZEA	119
4.1.2.	DATU BASEAREN OSAKETA	121
4.2.	Sare neuronalaren diseinu eta entrenamendua	122
4.2.1.	OBJEKTUEN DETEKZIOA ETA DISTANTZIA	127
ATALBURUA 5		131
1.	Kontribuzioa	133
1.1.	Lehen autore gisa egindako lanak	133
1.2.	Kolaborazioak	134
1.3.	Kongresuak	134
ATALBURUA 6		135
1.	Deliberamendua	137
ATALBURUA 7		141
1.	Lanaren Jarraipena	143
1.1.	Tesi honen jarraipena	144
BIBLIOGRAFIA		147
ERANSKINAK		183
1.	I. Eranskina: Baldintza-agiria	185
2.	II. Eranskina: Gidaliburua	186
2.1.	Errefortzu bidezko Ikaskuntzaren Optimizazioa	186
2.1.1.	OPTIMIZAZIO ANALITIKOA	186
2.1.2.	EBOLUZIO BIDEZKO OPTIMIZAZIOA	186
	Algoritmo genetiko bidezko optimizazioa	187

2.2.	SLAM bisuala	187
2.3.	Oztopoen distantzien kalkulua.....	188
3.	III. Eranskina: Kodigoa	190
3.1.	Errefortzu bidezko Ikaskuntzaren Optimizazioa	190
3.1.1.	OPTIMIZAZIO ANALITIKOA.....	190
3.1.2.	EBOLUZIO BIDEZKO OPTIMIZAZIOA	196
	Algoritmo genetiko bidezko optimizazioa	200
3.2.	SLAM bisuala	201
3.3.	Oztopoen distantzien kalkulua.....	209

IRUDIEN AURKIBIDEA

Irudia 1: Unibertsitatean eskuragarri dagoen ibilgailu autonomoa 1	4
Irudia 2: Ibilgailu autonomoaren sistemen arkitektura ([9]-ko irudian oinarrituta)	14
Irudia 3: Ibilgailu autonomoan ezinbestekoak diren egikizunak ([17]-ko irudian oinarrituta)	15
Irudia 4: Kontrol adimendunerako erabiltzen diren adimen artifizialeko teknikak ([32]-ko iruditik moldatua).....	19
Irudia 5: Ikaskuntza automatikoa eta ikaskuntza sakonaren erlazioen eskema	22
Irudia 6: Robot mugikorren ibilbideak planifikatzeko eskema orokorra ([19]-ko iruditik moldatua) ..	25
Irudia 7: Kamera monokular era Q-Sare sakon bikoitzarekin garatutako nabigazio sistema([66]-ko iruditik moldatua)	27
Irudia 8: Deep Neural Network sare neuronalaren arkitektura	29
Irudia 9: Feedforward sare neuronalaren arkitektura.....	31
Irudia 10: Autoencoder sare neuronalaren arkitektura	33
Irudia 11: Convolutional Neural Network sare neuronalaren arkitektura	36
Irudia 12: Arrain-begi kamerarekin lortutako mapatzea ([138]-ko iruditik moldatua).....	38
Irudia 13: Hiru dimentsiotako SLAM berreraikitzea klasifikazioarekin ([149]-ko iruditik moldatua)...	40
Irudia 14: Ibilbide plangintza oztopoak dituen ingurune batean abiapuntu ezberdinetarako ([157]-ko irudia).....	41
Irudia 15: Kontroladore aurreratua CNN-an oinarrituta ([158]-ko iruditik moldatua).....	41
Irudia 16: Recurrent Neural Network sare neuronalaren arkitektura.....	42
Irudia 17: RNN eta LSTM neuronen konparaketa.....	43
Irudia 18: Algoritmo genetikoaren fluxu diagrama.....	49
Irudia 19: SLAM genetikoaren eboluzioa generazioak aurrera egin ahala ([233]-ko iruditik moldatua)	52
Irudia 20: Fuzzy kontroladorearen osagai garrantzitsuenak	56
Irudia 21: PFL metodoak sortzen duen ibilbide plangintza ([291]-ko irudia)	59
Irudia 22: Ibilgailuen jarraipena fotogramen artean ([334]-ko irudia).....	65
Irudia 23: Ibilgailu autonomoen inguruan gaur egun dauden erroak	79
Irudia 24: Unibertsitatean eskuragarri dagoen ibilgailu autonomoa 2	85
Irudia 25: Unibertsitateko AMR-aren barne konfigurazioa	86
Irudia 26: Ibilgailu autonomoan eragina duten abiadura ezberdinak.....	87
Irudia 27: Nabigazio eremuaren 2D mapa.....	88
Irudia 28: Errefortzu bidezko ikaskuntzaren eskema	90
Irudia 29: AV-aren hurrengo posizioen errepresentazioa denboran zehar	90
Irudia 30: Antzezlearen sare neuronala	91
Irudia 31: Errefortzuko ikasketaren hasierako garaien simulazioa	96
Irudia 32: Errefortzuko ikaskuntzaren garai aurreratuago batzuen simulazioa	97
Irudia 33: Antzezle optimizatuaren emaitza 191 garaitan	97
Irudia 34: Errefortzu bidezko ikaskuntzaren sarearen eguneratzea	99
Irudia 35: Kritikoaren sare neuronala.....	99
Irudia 36: Reinforcement Learning agentearen entrenamendu adibidea	100
Irudia 37: Sarearen eboluzioarekin lortutako agentearen emaitza	102
Irudia 38: Algoritmo genetikoaren bidezko K parametroaren doitzea	104
Irudia 39: Algoritmo genetikoaren bidez lortutako nabigazioa	104
Irudia 40: Tag36h11 familiako AprilTag ereduak.....	105
Irudia 41: Kamera kalibraziorako AprilTag txantiloien irudiak.....	107
Irudia 42: Eskolako pasilloetan mugatutako gunea.....	108

Irudia 43: Kameraren eta AprilTag-en erreferentzia sistemen alderaketa	108
Irudia 44: AprilTag-en posizio erreferentzia absolutuaren kalkulua	109
Irudia 45: Planoen sinplifikazioa	110
Irudia 46: AprilTag-en irakurketaren bariazioak	110
Irudia 47: Ibilgailua mugitzerakoan eman daitezkeen bi kasuak	111
Irudia 48: Ibilgailua hormara hurbiltzen deneko erreferentzia aldaketak	112
Irudia 49: Ibilgailua hormatik aldentzen deneko erreferentzia aldaketak	113
Irudia 50: Pasilloko AprilTag-en lehenengo detekzioa	114
Irudia 51: Pasilloko gainontzeko AprilTag-en detekzioa	115
Irudia 52: AprilTag-en irudikapena espazioan	116
Irudia 53: Inguruko maparen irudikapena	116
Irudia 54: KITTI datu-basearen irudia eta puntu-hodeia	118
Irudia 55: LiDAR distantzia puntu-hodeia koloreztatuta	119
Irudia 56: Auzokide teknikaren bidez sortutako gelaxken irudikapena	119
Irudia 57: LiDAR irudia leunduta	120
Irudia 58: Datuen goiko partearen ezabaketa	120
Irudia 59: KITTI datu-basearen irudia eta puntu-hodeia	121
Irudia 60: Datu-basea osatuko duten irudi motak	121
Irudia 61: Distantziak kalkulatzeko sarearen egitura	122
Irudia 62: Sare neuronalaren entrenamendu grafika	124
Irudia 63: Sakontasunaren eta Sareak emandako eskalak [m]-tan	124
Irudia 64: Irudietan sortutako label-ak	127
Irudia 65: Oztopoen sareak detektatutako objektuak	128
Irudia 66: Oztopoen distantzia kalkulua	128
Irudia 67: Oztopoen distantzien emaitzak	129
Irudia 68: Sare neuronalaren etapak proiektuan	137

TAULEN AURKIBIDEA

Taula 1: Automatizazio mailak auto eta gizakiaren eginkizunekin [16]	13
Taula 2: Gida sistemak AGV eta AMR-entzako (goiko bista)([19]-ko irudian oinarrituta).....	16
Taula 3: Ibilgailu autonomoen erabilera ereduak industriaren barne ([19]-ko iruditik moldatua).....	18
Taula 4: Datu-base erabilienetakoak gidaketa autonomoan	63
Taula 5: Markatzaile fiduziarioen zerranda ([341]-ko taulatik moldatua).....	67
Taula 6: Antzezlearen sare neuronalaren ezaugarriak	92
Taula 7: Errefortzuzko ikaskuntzaren planteamenduaren algoritmoaren fluxua	95
Taula 8: RL-analitikoaren sare neuronala entrenatzeko balioak	95
Taula 9: Antzezlearen hasierako sare neuronalaren ezaugarriak.....	98
Taula 10: Kritikoaren sare neuronalaren ezaugarriak	100
Taula 11: Agentea definituko duten parametroak	101
Taula 12: RL eta GA algoritmoaren fluxua	101
Taula 13: Mapa sortzeko algoritmoaren fluxua.....	106
Taula 14: Kalkulatutako neurriak eta errealitatean neurtutakoak	117
Taula 15: Sare neuronalaren ezaugarriak.....	123
Taula 16: Sare neuronala entrenatzeko balioak	124
Taula 17: Distantzia sarearen entrenamenduaren emaitzak	125
Taula 18: Sarearen aldaketen datuak eta emaitzak	126
Taula 19: Oztopoen detekzioa egiten duen sare neuronala entrenatzeko balioak.....	127
Taula 20: Oztopoen detekzioa egiten duen sare neuronalaren emaitzak.....	128
Taula 21: Test-irudien objektuen neurri errealak.....	129
Taula 22: Oztopoen distantzia kalkulatzeko algoritmoaren fluxua	130
Taula 23: Autoreak egindako ekarpenak.....	133

AKRONIMOAK

Akronimoa	Definizioa	Itzulpena
AV	Autonomous Vehicle	Ibilgailu Autonomoa
NHTSA	National Highway Traffic Safety Administration	Bide Segurtasuneko Administrazio Nazionala
SAE	Society of Automotive Engineers	Automozio Ingeniarien Elkarteak
CAV	Connected Autonomous Vehicle	Ibilgailu Autonomo Konektatua
SLAM	Simultaneous Localization And Mapping	Aldibereko eta Lokalizazioa
AMR	Autonomous Mobile Robot	Robot Mugikor Autonomoa
AGV	Automated Guided Vehicle	Ibilgailu Gidatu Automatikoa
AI	Artificial Intelligence	Adimen Artifiziala
ML	Machine Learning	Ikaskuntza Automatikoa
EC	Evolutionary Computing	Konputazio Ebolutiboa
FZ	Fuzzy Logic	Logika Lausoa
NN	Neural Network	Sare Neuronala
EA	Evolutionary Algorithms	Algoritmo Ebolutiboak
GA	Genetic Algorithms	Algoritmo Genetikoak
DL	Deep Learning	Ikaskuntza Sakona
ANN	Artificial Neural Network	Sare Neuronal Artifiziala
CNN	Convolutional Neural Network	Sare Neuronal Konboluzionala
RNN	Recurrent Neural Network	Sare Neuronal Errepikaria
DNN	Deep Neural Network	Sare Neuronal Sakona
WSN	Wireless Sensor Network	Haririk Gabeko Sentsoreen Sarea
LiDAR	Light Detection and Ranging	Argi eta Irismen Detekzioa
RL	Reinforcement Learning	Errefortzu Bidezko Ikaskuntza
DQN	Deep Q-Network	Q-Sare sakona
DRL	Deep Reinforcement Learning	Errefortzu Bidezko Ikaskuntza Sakona
FPA	Flower Pollination Algorithm	Lore-Polinizazio Algoritmoa
DDQN	Double Deep Q-Learning	Q-Ikaskuntza Sakon Bikoitza
HRL	Hierarchical Reinforcement Learning	Errefortzu Bidezko Ikaskuntza Hierarkikoa
DDPG	Deep Deterministic Policy Gradient	Politika Determinista Sakonaren Gradientea
ROS	Robot Operating System	Roboten Sistema Eragilea
PPO	Proximal Policy Optimization	Politika Proximalaren Optimizazioa
IARL	Intervention Aided Reinforcement Learning	Interbentzioz Lagundutako Errefortzu Bidezko Ikaskuntza
RSS	Received Signal Strength	Jasotako Seinalearen Intentsitatea
RRT	Rapidly-exploring Random Tree	Miaketa Azkarreko Ausazko Zuhaitza
MLP	Multilayer Perceptron	Geruza Anitzeko Pertzeptroia
UWB	Ultra Wide Band	Banda Ultra Zabala
FNN	Feedforward Neural Network	Berrelkadura Sare Neuronala
DDP	Discrete Dynamic Programming	Programazio Dinamiko Diskretua
PSO	Particle Swarm Optimization	Partikula-Multzo Bidezko Optimizazioa
IMU	Inertial Measurement Unit	Neurketa Inertzialeko Unitateak

INS	Inertial Navigation System	Nabigazio Sistema Inertziala
GPS	Global Positioning System	Posizionamendu Globaleko Sistema
GNSS	Global Navigation Satellite System	Satelite bidezko Nabigazio Sistema Globala
Wi-Fi	Wireless Fidelity	Hari Gabeko Fideltasuna
BoW	Bag of Words	Hitz-Poltsa
WLAN	Wireless Local Area Network	Hari Gabeko Sare Lokala
SPAD	Single-Photon Avalanche Diode	Fotoi Bakarreko Olde Diodoa
ReLU	Rectified Lineal Unit	Unitate Lineal Zuzendua
GRU	Gated Recurrent Units	Unitate Errepikari Itxiak
OPP	Overlapping Pyramid Pooling	Piramide Gainjarriaren Bateratzea
LSTM	Long Short-Term Memory	Epe Laburreko Memoria Luzea
MPC	Model Predictive Control	Ereduen Kontrol Prediktiboa
YOLO	You Only Look Once	Behin Bakarrik Begiratzen Duzu
LVD	Local Voronoi Diagram	Voronoi Diagrama Lokala
IML	Incremental Maximum Likelihood	Gehitze Probabilitate Maximoa
FLC	Fuzzy Logic Controller	Kontrolatzaile Logiko Lausoa
FIS	Fuzzy Inference System	Inferentzia Lausoko Sistema
PFL	Probabilistic Fuzzy Logic	Logika Lauso Probabilistikoa
WDO	Wind Driven Optimization	Haizeak Bultzatutako Optimizazioa
MVS	Multi View Stereoscopic	Bista Anitzeko Estereoskopia
MOT	Multiple Object Tracking	Objektu Anitzen Jarraipena
VOC	Visual Object Classes	Ikusizko Objektu Klaseak
ViT	Vision Transformer	Bisio transformatzailean
AR	Augmented Reality	Errealitate Areagotua
SIFT	Scale Invariant Feature Transform	Eskala Ezaugarri Aldaezinen Transformatu
VIO	Visual Inertial Odometry	Ikus Odometria Inertziala
DIC	Digital Image Correlation	Irudi Digitalen Korrelazioa
ICP	Iterative Closest Point	Hurbileneko Puntu Iteratiboa
RFID	Radio Frequency Identification	Irrati Frekuentziaren Identifikazioa
DWA	Dynamic Window Approach	Leiho Dinamikoaren Ikuspegia
VFF	Virtual Force Field	Indar Birtualeko Eremua
PFF	Potential Force Field	Indar Potentzialaren Eremua
VFH	Vector Field Histogram	Eremu Bektorialaren Histograma
TFH	Traversability Field Histogram	Ibiltzeko Eremu Histograma
XAI	Explainable Artificial Intelligence	Adimen Artifizial Azalgarria
IPC	Industrial Personal Computer	Ordenagailu Pertsonal Industrialak
FC	Fully Connected	Gutziz Konektatua
MSE	Mean Squared Error	Batez besteko Errore Koadratikoa

ATALBURUA 1

SARRERA

*Pentsatzeko zure eskubidea defendatu,
era okerrean pentsatzea ere ez pentsatzea baino hobetagoa da.*

ALEXANDRIAKO HIPATIA

1. Hitzaurrea

Eguneroko bizitzan, gero eta nabariagoa da adimen artifiziala erabiltzen duten aplikazioen etorrera. Teknologia hau sektore askotan zabaltzea lortu du, gero eta gehiago erabiliz. Hau, azken urteotan jasan duen hazkunde handiaren ondorio zuzena da, gizaki eta makinek duten gaitasunen arteko harresia kasik hautsiz. Hain da handia izan duen eboluzioa, adimen artifiziala tresna oso baliotsu batean bilakatu dela, makinei gizaki gisa pentsatzeko eta jarduteko aukera emanez ere.

Garapen honen ondorioz jorratutako eremuetako bat ikusmen artifizialarena izan da, honen helburua makinei mundua gizakiek egiten duten bezala ikusaraztea izanez. Modu honetan, adimen artifizialak emandako ezagutza, irudi eta bideoen errekonozimendu atazetan, irudien klasifikazio eta analisisan edota inguruneen birsorkuntzan aplikatu daiteke, adibidez. Ordenagailu bidezko ikusmenaren teknikak denboran zehar aurrera egin dute baina sare neuronalen etorrerak eman die bultzada handiena. Ikaskuntza sakoneko teknikak izan dira hain zuzen ere hobekuntza handi hori sortu dutenak errendimendu aldetik, ohiko irudi prozesaketa metodoekin alderatuz.

Ikusmen artifizialak babestuta, ibilgailu autonomoen teknologiaren alderdiak bide egin du, gorakada ikaragarriarekin. Eboluzio etengabea jasaten hari den mundu honetan, mugikortasun seguru eta eraginkorra lehenetsuna hartu duen kontzeptu batean bilakatu da eta adimen artifizialak bultzatutako ibilgailu autonomoek, iraultza teknologiko bat suposatzen dute. Etorkizun handiko alor honek, ibilgailuen industria transformatzeko aukera dauka eta geroz eta gehiago nabarmentzen ari da. Hala eta guztiz ere, nahiz eta gidaketa autonomoak ibilgailuen gidatze automatizatuak ekar dezakeen gogora, ez da barneko ibilgailu autonomoen kontrola alde batera utzi behar. Nahiz eta bigarren honek ez jaso lehenengoaren adinako atentzio, ikasteko aukera paregabea eskaintzen du, ibilgailu hauek ulertzeko eta diseinatzeko okasioaz.

Tesi lan hau ibilgailu autonomoek aplikatutako adimen artifizialaren munduan barneratzen da, aurrerapen teknologikoak miatuz, eta horrek dakartzan erronkak eta ondorioak aztertuz. Barne ibilgailu autonomoetarako gidatze-sistemen garapenean jartzen da fokua, ikusmen artifizialeko algoritmoak erabiliz besteak beste. Kontzeptu berriekin hastea zaila denez, garapenaz haratago, lehendik existitzen diren sistemen hobekuntzan du ardatza. Helburu honetarako, bai ikaskuntza sakonean oinarritutako algoritmo konplexuagoak bai metodo sinpleagoak, trigonometria bezala, hartuko dira kontuan. Orokorrean tesi honen helburua arlo hau ezagutzen eta eztabaidatzen laguntzea da, gidatze autonomoa nonahi dagoen etorkizunerako bidean dauden aukerak eta erronkak azpimarratuz.

Proiektua Vitoria-Gasteizko ingeniarietza eskolan garatua izan da tesi honen zuzendariarekin batera, Ekaitz Zulueta Guerrero doktorea. Unibertsitate honetan, Sistemen Ingeniaritza eta Automatika sailean ibilgailu gidatu automatiko baten eredu bat dago eskuragarri, lanen bilakaeran lagungarri izateko. Ibilgailu hau, ibilgailu autonomo industrial baten prototipo bat da eta azken urteotan unibertsitateko ikasleek diseinatu eta garatua izan da. Eredua **Irudia 1**-ean ikus daiteke eta lan honetan egindako azterketak bertan aplikatzeko dira hain zuzen ere.



Irudia 1: Unibertsitatean eskuragarri dagoen ibilgailu autonomoa 1

Proiektu honetako algoritmoen garapena egiteko MATLAB programa erabili da eksklusiboki, MATLAB R2022b bertsioan. Algoritmo hauek ikaskuntza sakoneko kodigoek eta ikusmen artifizialekoek osatzen dute. Era berean, *webcam* baten erabilera ere egin da sentsore gisa eta markatzaile fiduziariorak inprimatu dira ikusmenaren euskarri bezala.

Automatizazio-elementu berri baten garapenean lan egin da, hala nola PLC edo robot industrialetan, fabriketan aplikagarria izan dadin. Beraz, ondoriozta daiteke tesi honen arazo nagusia barrualdeko ibilgailu industrialen, bai lokalizazio, bai nabigazio algoritmoak garatzea dela. Gainera, segurtasunaren kontzeptua ere aintzat hartu da.

1.1 Gaur egungo testuingurua

Ibilgailu gidatu automatikoak definizioz, *materialak manipulatzeko sistemak edo karga-eramaileak dira, eta biltegi, banaketa-zentro edo fabrikazio-planta batetik modu autonomoan mugitzeko ahalmena dute, langile edo gidaririk gainean eraman gabe* [1]. Ondorioz, bi punturen artean mugitzen doan robot bat bezala interpretatu daiteke, bitartean atazak betetzen doana eta ez duena behar gizaki baten kontrol zuzena.

Sektore industrialean ibilgailu mota hauen erabilera urteak daramatza, 1954-ean hasita [2], fabriketatik nabigazio librea eginez eta langileei euskarria emanaz. Begi-bistakoa da beraz, ingurunea ezagutzeko premian pentsatzea. Hau orokorrean mapaketa eta lokalizazioaren bidez gauzatzen da, sentsoreen bidez edota ikusmen artifizialaz egin daitekeena. Behin nabigazio ingurunea ezaguna denean, bi punturen arteko ibilbidea kalkula daiteke eta ibilgailuak hura jarraitu dezake. Gainera, oztopo dinamiko eta estatikoak antzematea ere garrantzi handia dauka talkak saihesteko, segurtasuna bermatuz.

Lehenengo ibilgailu autonomoak fabriketan lurperatutako kable baten beharra zuen, kanpo magnetiko bat sortzen zuena ibilgailuaren gida gisa [3]. Arrunki, hauek aurkitzen jarraitu daiteke, hau da, filo-gidatuak. Nabigazio sistema sendo bat diren arren, ikuspegi hau ez da batere malgua muntaketa-lerroetan zehar atazen aldaketak egiteko. Ondorioz, esan daiteke gaur egun orokorrean industrietan ez direla nabigazio guztiz librea duten robotak erabiltzen. Aipatu bezala, garrantzitsua da ingurunearen ezagutza ibilbidearen kalkulurako baina aktualitatean, gehienak gizakiak aurrezarriritako bideak jarraitzen dituzte. Honek, ezusteko gertakarien aurrean erantzuteko gaitasuna murrizten du.

Jakina da ibilgailua berari ibilbidea kalkulatzeko eta jarraitzeko gaitasuna eman nahi baldin bazaio, diseinatzen diren nabigazio sistemak hainbat informazio-iturritako datuen irakurketetan oinarritu behar direla. Hau askotan konplexua da eta industriek ekidin egiten dute. Alabaina badira nabigazio guztiz librea duten roboten erabilera babesten duten enpresak. Hauek kontuan daukate ibilgailu autonomoen zehaztasuna handitzeko datuen bilketa oso inportantea dela, lokalizazioa gauzatzeko eta robotek ibilbidea jarrai dezaten modu eraginkorrean. Duten fidagarritasun eta errendimenduaren ondorioz, pixkanaka, industria munduan tarte bat egin dute, aldi berea, hauen aplikazioak eta sofistikazioa handituz.

Ildo beretik, ibilgailu autonomo industrialek robotika mugikorraren arloan abantaila asko ekartzen dituzte. Alde batetik, beharrezkoa da inguruari buruz jasotzen den informazioaren kudeaketa, ibilgailua autonomoaren kokapena bera zenbatetsi ahal izateko. Izan ere, ibilgailuak espazioan duen posizioa ez ezik norabidea ere barneratu behar da, hau da, \overrightarrow{pose} -a. Honek datuen tratamendu metodologiari laguntzen dio, ibilgailuek iragazkiak erabiltzen bait dituzte ingurunearen ezaugarrien arteko bereizketak markatzen dituzten datuekin soilik geratzeko. Beste alde batetik eta aipatu den bezala, robotak ibilbidea kalkulatzeko eta, oztopoak saihestuz, aurrera egiteko duen gaitasunak, sistema oso malguak bihurtzen ditu.

Informazio-iturriak sentsoreak dira eta ohikoenen artean LiDAR-ak aurki daitezke. Gurpil odometria ere erabili ohi da posizioaren aurreikuspenean. Algoritmoei dagokienez, partikula iragazkia edo *Dynamic Window Approach* algoritmoa erabilienean artean daude. Teknika hauek oso konbentzionalak dira, ibilgailu autonomoen kokapen eta nabigazioa lortzeko, zenbait metodologia berritza eta sartzea beharrezkoa izanik. Azken urteetan adimen artifizialeko teknologia duten robotak garatu dira, *wecam*-ak sentsore bezala gehituz. Ordenagailu bidezko ikusmenak funtsezko zeregina betetzen ari da ikaskuntza automatikoko eta sakoneko ereduaren garapenean, ikuspertzepzioan oinarrituta. Optimizazioak ere garrantzi handia irabazten hari du sistemen hobekuntza bilatzeko lanetan. Hau, aplikazio askok beharrak betetzen ez dituztela ikusten hari delako da, ibilgailuekin batera teknikek ere eboluzio baten beharra bait dute.

Pertsonen dagokienez, ibilgailu autonomoen erabilerak industrien automatizazio-maila handitzen dute eta eramangarriagoak bilakatzen dituzte zenbait ataza. Lanpostu gogor eta arriskutsuenetan oso baliagarriak dira ere, hauetan pertsonen lana guztiz ordezkatzuz. Ondorioz, produktibitate fidagarri eta eraginkorragoa lortzen da.

1.2 Helburuak eta irismena

Proiektu honen irismena barneko ibilgailu autonomoen nabigazio algoritmo baten diseinu eta garapena da. Helburu orokorra nabigazio-sistema autonomo malgu bat lortzea da, ingurune edota ibilgailu ezberdinetan aplikagarria dena. Modu honetan, fabrikaz edota ibilgailu modeloaz haratago, barnealde industrialetan roboten kontrol fidagarri bat lortu nahi da. Proposatzen den konponbidea beraz, hiru atal nagusitan banatu daitekeen nabigazio algoritmo bat da: alde batetik, ibilbidearen plangintza eta kontsigna lortzea, bestetik, aldibereko mapatze eta ibilgailuaren lokalizazioa eta azkenik, segurtasun bermatzea.

Ibilbidearen plangintzak garrantzi handia jasotzen du, ibilgailuaren mugimendua gauzatzeko kontsignak lortzeaz gain, talkarik gabeko bideak kalkulatzeko dituelako. Ingurune osoko mapa emanda, gai izan behar da bi punturen arteko biderik optimoena kalkulatzeko. Badaude algoritmo desberdinak hau egiten dutenak baina errefortzuzko ikasketan fokua jarritz, ez da lortu guztiz egokia den metodorik. Ondorioz, errefortzuzko ikasketaren optimizazioa bilatzen da ibilbidea kalkulatzeko asmoarekin. Lehenengo diseinu hau *offline* exekuzioa izango du, ibilgailua martxan dagoenean kostu konputazionala arintzeko. Hala ere, ibilgailua galdu egiten baldin bada, *online* exekutatu ahalko den algoritmo bat ere bada, ibilbidea birkalkulatzeko.

Errefortzuzko ikasketa aplikatzeko mapa bat lortzean datza bigarren algoritmoa, ibilgailua lokalizatzeko modu bat aurkitzeaz gain. Algoritmo honen lehenengo parte, hau da, mapaketa, *online* egingo da baina miaketa egoeran. Behin ingurunea ezagututa ibilbidea kalkulatzeko aukera dago eta, ibilgailua lanean dagoenean, patroï fiduziarioen biden lokalizazioa *online* egitea. Denbora errealean eginda, ibilgailuaren pausuak mapatzeko aukera ematen du gainera. Azken algoritmoa segurtasuna gehitzen dio ibilgailuari, industria inguruko oztopoak identifikatu dezan eta horren arabera erantzun dezan, *online* exekutatu. Garrantzi handia du azken hau denbora errealean funtzionatzea, erantzun azkarra eman dezan edozein eragozpenaren aurrean. Azken bi hauek balio erantsia ematen diote nabigazio algoritmoari.

Diseinu hauek aurrera eramateko, adimen artifizialeko teknikak ikasi nahi dira batez ere errefortzuzko ikasketan eta ikaskuntza sakonean oinarrituta daudenak. Gainera, ordenagailu bidezko ikusmenak ere pisua dauka lan honetan. Horrela, erabat autonomoa den nabigazio sistema baten diseinua egiten saiatuko da, kostu txikioa, malgua, segurua eta era egokian funtzionatzen duena ibilgailu autonomoetan aplikatzeko. Oso garrantzitsua izango den alderdi bat entrenamendu datuen kopuru eta informazioa izango da, emaitza zehatzak lortzeko. Exekuzio denbora ere baloratuko da, etorkizunean algoritmoak behar bezain azkar exekuta daitezen erabilgarri dagoen *hardware*-an jartzeko.

Nahiz eta helburu orokorra ibilgailu autonomo baterako nabigazio sistema bat garatzea izan, zenbait azpi-ataza finkatu beharra ikusten da:

- a) Errefortzu bidezko algoritmoaren optimizazio era bat bilatzea. Horrela, mapa bat emanda gai izan beharko da puntu batetik bestera heltzeko era egokian, nabigazio kontsignak lortuz. Aurreikusten den ibilbidearen kontsignak gorde ahalko dira, t_N aldiunean (x_N, y_N) posiziora bideratzeko eta nabigazioa gauzatzeko.
 - a. Errefortzu bidezko teknikaren optimizazio analitiko bat egin, erabiltzen diren sare neuronalenak hain zuzen ere.
 - b. Sare neuronalak beste tekniken konbinazioaz optimizatu, kasu honetan eboluzioarekin.
 - i. Algoritmo genetikoaren bidez "sari" bidezko nabigazioa optimizatu.
- b) Inguruneko mapa modu sinplean lortzen duen algoritmo bat garatzea. Behin mapa ezaguna dela, ibilgailua nabigatzen dagoenean honen jarraipea eta lokalizazioa egiteko aukera egongo da.
- c) Objektu eta langileak detektatzen dituen algoritmo bat egitea. Modu honetan, oztopoa detektatzerakoan honen distantzia aurreikusiko da eta honen arabera ibilgailuak akzio ezberdinak erabaki dezake, adibidez, gelditzea oztopoa desagertu edo urrun egon arte.

Beste alde batetik, aurreko azpi-helburuak lortzeko lan ezberdinak egiteko beharra ere kontsideratu da:

- Barruko eta kanpoko ibilgailu autonomoen nabigazio-sistemei buruzko azken urteotako bibliografia-aurrekarien azterketa bat egitea.
- Nabigazio-sistemen beharren hausnarketa bat egin eta premia horiei irtenbidea nola eman dakioken aztertu.
- Nabigazio-sistema bat garatzea, ibilgailu autonomoa modu guztiz autonomoan mugitzeko eta hainbat ingurune eta ibilgailuetarako malgua izan dadin.
- Lortutako emaitzak aztertu garatutako algoritmoa baloratzeko.

Aipatu den bezala garapena MATLAB bidez egingo da, eta Vitoria-Gasteizko ingeniarietza eskolan dagoen ibilgailu eredua izango du oinarritzat.

1.3 Memoriaren egituraketa

Behin tesi honen gaiaren ikuspegi orokor bat aurkeztu ostean, hurrengo atalak era honetara antolatuta agertuko dira. Lehendabizi, bigarren atalburuak *“Teknikaren Egoera”* aurkezten du, non bibliografiaren analisia agertzen den, teknologiaren aurrerapenekin gaur egun arte. Hirugarren atalburuak *“Beharren Hausnarketak”* jasotzen ditu, gai honetan landuko diren algoritmoen arrazoiketa batekin batera. Jarraituz, *“Metodologia”* aurkitu daiteke laugarren atalburuan, algoritmoen garapen propioarekin. Atal hau hiru zati nagusitan banatuta dago, hauek *“Errefortzu bidezko ikaskuntzaren optimizazioa”*, *“SLAM bisuala”* eta *“Oztopoen distantzien kalkulua”* izanik. Ondoren, lan honen burutzeak ekarri dituen *“Autorearen Ekarpinak”* biltzen dira. Azkenik *“Ondorioak”* eta *“Etorkizuneko Ikerketa Ildoak”* aurkitu daitezke, eranskinei pausu emanaz.

ATALBURUA 2

TEKNIKAREN EGOERA

Mundu honetan ez zaio ezeri beldurra izan behar...ulertu egin behar da.

Orain da gehiago ulertzeko unea, beldur gutxiago izan dezagun.

MARIE CURIE

1. Gidatze Autonomoa

Azken urteetan, ibilgailu autonomoak (*AV-Autonomous Vehicle*) garrantzi gehiago irabazten ari dira etorkizuneko berrikuntza eta adimen ikur bezala, garraioaren esparruan. Teknologia berri honek atentzio handia jaso du hainbat sektoreetan, bai pribatu, publiko eta akademikoan, baina oraindik erronka bat suposatzen du. Nabaria da garraioak funtsezko eginkizuna duela gaur egungo gizarte eta ekonomian, eta etorkizuneko mugikortasun modu honek negozio-eredu berriak eskaintzea espero da.

AV-ak industrian integratzeari dagokionez, literaturak aurreikusitako hainbat abantaila eredu jasotzen ditu [4]. Alde batetik errentagarriak dira, ohiko erregaien ordeztu energia elektrikoa erabiltzeak kostuak murriztu ditzake eta, aldi berean fabrikazio eraginkortasuna handitu dezake, ingurumena errespetatuz. Bestetik, digitalizazioari dagokionez, ibilgailu autonomoek sentsoreen bidez jaso ditzaketen datuak enpresentzako lagungarri dira, informazio guztiaren kontrola izateko. Segurtasunaren aldetik, denbora errealeko jarraipen eta gainbegirapenak ibilgailuaren funtzionamendu egokia bermatzen dute. Honetaz gain, ekoizpen-kostu baxuagoak eta prozesuen estandarizazioak aprobetxatzeko aukera ematen dute [5].

Bibliografiak ere AV-en aplikazioen inguruan sortutako ardurak batu ditu. Izan ere, ekipamendu hutsegiteek edo datuen segurtasun eta pribatutasun arazoek adibidez, ibilgailu autonomoen eraginkortasuna zalantzan jartzen dituzte [6]. Gainera, inplementazio eta mantentze kostuak altuak dira, eta gaitasuna duen langile batengatik erabiliak izan behar dira [5]. Argi dagoena da, AV-en etorrera saihestezinak modu batean edo bestean eragina izango duela industriaren sektorean, eta hauen hartze orokortuak hainbat traba gainditzea suposatuko duela, hau da, ez da arriskurik gabeko ezarpen bat izango.

Egia da gaur egun oraindik, kanpoko ibilgailu autonomoek interes gehiago pizten dutela. Hala ere, teknologia, legedia, azpiegitura eta gizartea kontuan hartuz [7], oztupo ugari daude eztabaida gai direnak [8]. Alde batetik, erabiltzaile/gobernuaren ikuspuntua dago, erabiltzaileen onarpen eta jokaera, segurtasuna eta legedia barneratzen dituen. Beste aldetik, informazio eta komunikazio teknologiak daude. Puntu honetan, *hardware* eta *software* informatikoak, komunikazio sistemak, zibersegurtasuna eta posizionamendu eta kartografia zehatzak biltzen dira [9]. Jakina da, komunikazio sareak asmo gaiztoko erasoei ahulak direla, segurtasun zibernetiko eta fisikoa kaltetuz. Honi aurre egiteko, espezifikoa ez den legedi berri baten sarrera proposatzeaz gain, gai horiek aztertzen dituzten lan taldeak sortzea, zibersegurtasunaren inguruko ikerketa finantziatzea eta fabrikatzaileei oinarriak erakustea begiesten da.

Ikus daitekenez, lege eta arauak ez daude kanpo ibilgailuak ezartzeko guztiz prest, beraz industria eta ikerkuntzaren esku dago AV-en garapenarekin jarraitzea, barneko ibilgailuekin [10], non lan handia dago oraindik egiteko. Ibilgailu autonomoak guztik ulertu eta kontrolatzeko intentzioarekin, antzeman daiteke azken urteetan gaiaren inguruko literatura bolumenak gora egin duela, hainbat alor bilduz, [11]. Azken bi hamarkadetan hazkunde esponontzial bat izan dute, zientzia-agerkarien %50-a azken 17 urteetan argitaratuz. Hau, ondorioz, AV-en gaineko interesaren hazkuntzaren adierazgarri da.

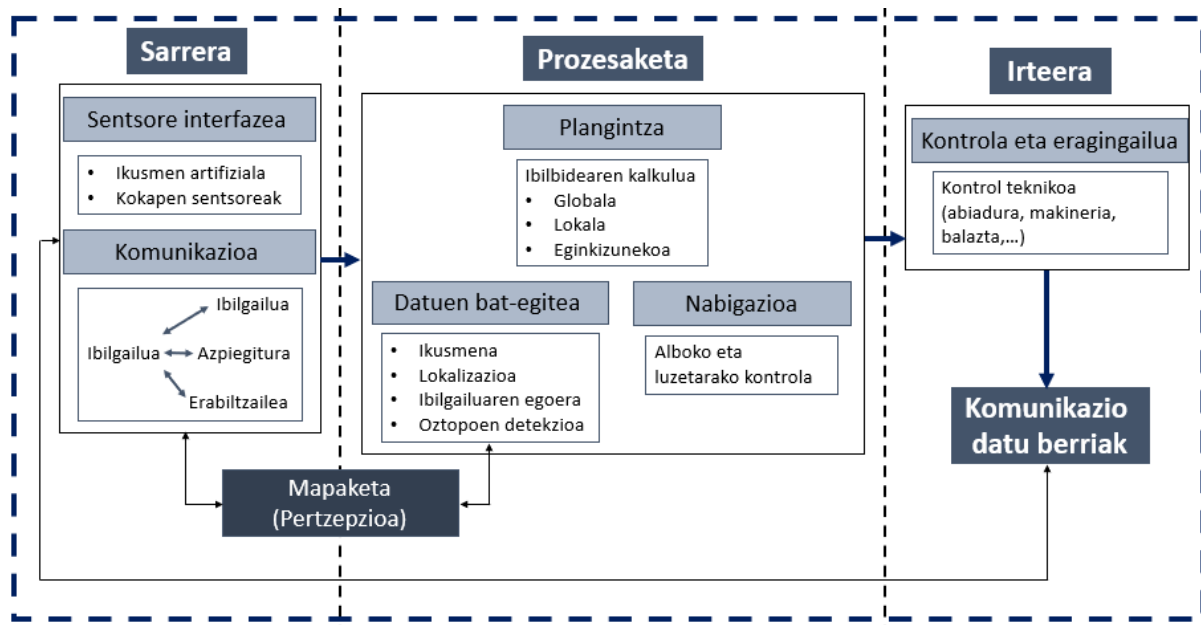
Ibilgailu autonomoen kontzeptua ulertzeko, lehendabizi autonomia izan behar da kontuan. 2016-an NHTSA-ak (*National Highway Traffic Safety Administration*) autonomia mailak definitu zituen automatizazioaren esparrurako [12]. Automatizazioaren barne gainera, badaude zenbait osagai kontuan izan behar direnak AV-en klasifikazioa egin ahal izateko. Garrantzitsuena erabiltzen hari den teknologiaren maila da, hau da, ibilgailuan behar den giza esku-hartzearen graduak honen sailkapena eta autonomia maila definitzen ditu [7]. Katēgoria hauek estandar bat bihurtu dira sektorean eta askotan aurkitu daitezke bibliografia akademikoan zehar [6], [7], [13], [14]. Hain zuzen ere, Automozio Ingeniarien Elkarteak (*SAE-Society of Automotive Engineers*) klasifikazio sistema bat garatu du maila hauek ebaluatzeko [15]. **Taula 1**-ek funtzioak deskribatzen ditu, gizakiaren zeregina automatizazio maila bakoitzean kontuan hartuz.

Taula 1: Automatizazio mailak auto eta gizakiaren eginkizunekin [16]

Automatizazio maila		Garraio-sistema automatizatua		Giza erabiltzailea	
		Funtzio operatiboa	Gaitasuna	Funtzio operatiboa	Gaitasuna
0. maila: Automatizazio gabea	Eskuz gidatzea, gizakia da erabat arduraduna.	-	-	Alboko eta luzetarako kontrola, lokalizazioa, pertzepzioa, plangintza, kudeaketa	Gidatzeko modu guztietan
1. maila: Lagundua	Funtzio gehienak gizakiak kontrolatzen ditu	Alboko eta luzetarako kontrola	Gidatzeko modu batzuetan	Lokalizazioa, pertzepzioa, plangintza, kudeaketa	Gidatzeko modu guztietan
2. maila: Partzialki automatizatua	Gizakiari laguntzeko sistema bat gutxienez automatizatuta dago	Alboko eta luzetarako kontrola	Gidatzeko modu batzuetan	Lokalizazioa, pertzepzioa, plangintza, kudeaketa	Gidatzeko modu guztietan
3. maila: Baldintzekin automatizatua	Gizakiak segurtasunerako funtzio-kritikoak transferitu diezazkioke ibilgailuari	Alboko eta luzetarako kontrola, lokalizazioa, pertzepzioa, plangintza	Gidatzeko modu batzuetan	Kudeaketa	Gidatzeko modu guztietan
4. maila: Oso automatizatua	Erabat autonomia, baina ez leku guztietan gidatzeko	Alboko eta luzetarako kontrola, lokalizazioa, pertzepzioa, plangintza, kudeaketa	Gidatzeko modu batzuetan	-	-
5. maila: Erabat automatizatua	Erabat autonomia, ibilgailuaren errendimendua gizakiarena bezalakoa da gidatze-egoera guztietan	Alboko eta luzetarako kontrola, lokalizazioa, pertzepzioa, plangintza, kudeaketa	Gidatzeko modu guztietan	-	-

Teorian, ibilgailu sistema automatizatu bat autonomo bezala definitzeko, gidatze dinamikoaren eginkizun guztiak, edozein ingurunean, ibilgailuaren sistema automatizatu batekin egin behar ditu. Honek, AV motak 3. eta 5. mailen artean dauden ibilgailuetara murrizten ditu [16].

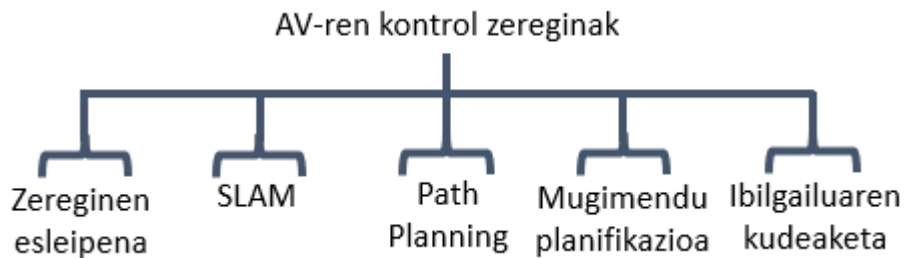
Beste alde batetik, gidaketa autonomoak funtzio askoren beharra dauka eta hauek elkarlanean jardun behar dute. **Taula 1**-an ikus daitezenez, lokalizazioa, pertzepzioa, plangintza, kontrola eta gestioa dira eginkizun hauek. Hasteko, alde aurreko baldintza bezala, ibilgailu autonomoak informazioa jaso behar du ingurunetik lokalizazioa eta pertzepzioa egin ahal izateko. Pertzepzioa, inguruaren mapaketa bezala ere izendatu daiteke. Gainera, AV-ak beste azpiegitura batekin komunikazio beharra badauka informazio gehiago edota kontrola zuzentzeko Ibilgailu Autonomo Konektatu (CAV-*Connected Autonomous Vehicle*) bezala ezagutzen da. Aipatzekoa da kanpo konexio honen erabilera ez dela nahitaezkoa. Honetaz gain, behin datu guztiakin, ibilbidearen plangintza egin beharra dauka, ostean, nabigazioa egiteko eta kontrola abiarazi ahal izateko. Hurrengo **Irudia 2**-an ikus daiteke ibilgailu autonomo baten beharrezko sistemen arkitektura, azaldutakoa laburbilduz.



Irudia 2: Ibilgailu autonomoaren sistemen arkitektura ([9]-ko irudian oinarrituta)

Guzti hau ahalbidetzeko badaude zenbait eginkizun gako direnak ibilgailu autonomoetan (ikusi **Irudia 3**). Lehendabizi, ibilgailu autonomoari egin behar bat edo batzuk esleitu behar zaizkio, zer egin behar duen jakin dezan. Honen ondoren, edozein AV-ak modu independentean bere ingurunea hautemateko gai izan behar da, beharrezko informazioa jasotzeko eta horren arabera kontrol erabakiak hartzeko. Inguruan dauden objektu guztiak hiru taldetan banatzen dira: eskualde hautespena, ezaugarrien ateratzea eta sailkapena. Kontuan izan behar da ikusmen errealean hainbat mugaketa eta zailtasun daudela, esate baterako, ibilgailua autonomoaren ikusmen mota partziala, albokoa edo angelukoa izan daitekeelako adibidez. Gainera, badaude antzerako itxura duten oztopoak distantzia ezberdinetan, tamaina ezberdinekoak iruditu ahal dutenak, errealitatea ez errepresentatuz. Guztiari argiztapena gehitu behar zaio [13]. Arrazoi hauengatik, funtsezkoa da aldebereko lokalizazio eta mapaketa egitea, SLAM (*Simultaneous Localization And Mapping*) bezala ezagutzen dena. Robotaren posizioa ezagutzeko ezinbestekoa da zeregin hau betetzeko, pose bezala izendatzen dena, $\vec{p}ose = [x, y, z, \theta]$ bektorearekin definituz.

Lanekin jarraituz, ezinbestekoa da ibilbide planifikatzaile bat izatea. Zeregin konplexu bat da, non bi puntuen arteko bide optimoa aurkitu behar den. Kalkulua AV-an erantsita dauden sentsore ezberdinetan oinarrituta dago. Ibilgailu autonomo on bat, lortutako datuen analisi egoki bat egiteko gai izan beharko litzateke, oztopo gabeko ibilbidea aurreikusteko. Kontuan izan behar da gainera konputazionalki eraginkorra izan behar dela. Honen arduradunari *Path Planning* deritzo. Ustekabeko oztopoei azkar aurre egitearen zeregina mugimendu kontrolari esleitzen zaio. Modu honetan, behar den ekintza exekutatu litzateke denbora errealean, norabidean edota ibilbidean moldaketak eginez horrela behar baldin bada. Gainera, kontuan izan behar da oztopoak dinamiko edo estatiko bezala sailkatu daitezkeela. Azkenik ibilgailuaren kudeaketa dago, non gertatutako hutsegiteak, bateria edo mantentze lanen beharra aztertzen den.



Irudia 3: Ibilgailu autonomoan ezinbestekoak diren egikizunak ([17]-ko irudian oinarrituta)

Badaude ibilgailu autonomoei gehitu dakioken beste zenbait lan, baina hurreko hauek dira funtzionamendu egokia izateko funtsezkoak direnak. Aipatzekoa da, ibilgailu autonomoekin erlazionatuta dauden teknikak bi multzo nagusitan banatu daitezkeela. Alde batetik, robotek mapen beharrik ez izatea gertatu ahal da, eremu ezezagun batean mugituz. *Online* bezala ezagunak dira kasu honen barnean dauden teknikak [18], eta sentsoreen informazioaz baliatuz, habia puntu batetik definitutako helmuga batera heltzeko gai dira talkak saihestuz. Bestetik, *offline* teknikak, eremua ezaguna izatea ezinbestekoa den lanak biltzen dituenak. Hauek ibilbide plangintza atazak biltzen dituzte batez ere, eta norantza eta abiadura bezalako parametroak zehaztu behar dira. Honek ez du esan nahi bi teknikak aldi berean erabili ezin direnik ibilgailu autonomoen nabigaziorako, bakoitzak, baliabide desberdinak ematen bait ditu.

1.4 Ibilgailu autonomoak industriaren barne

Industrian erabiltzen diren ibilgailu autonomoei buruz hitz egiteko, bi ibilgailu eredu definitu behar dira, nabigazio ahalmenaren arabera. Hasteko robot mugikor autonomoa dago (AMR-*Autonomous Mobile Robot*), *hardware* eta kontrol *software* aurreratuak dituenak, eginkizun autonomoak ahalbidetzen dutenak inguru dinamikoetan. Orokorrean sistema moldakorrak dira [19], ez dutelako gida-dispositibo fisikoen edo elektromekanikoen beharrik [20]. Beste aldetik, ibilgailu gidatu automatikoa (AGV-*Automated Guided Vehicle*) dago, non kontrol unitate zentral batek erabakiak hartzen dituen [17]. Hauek egonkorragoak dira sistemaren sinpletasunaren ondorioz. Beraz, antzeman daiteke bien arteko gida sistemek ezberdintasunak dituztela. **Taula 2**-an ikus daiteke ibilgailu mota bakoitzeko erabili ohi diren gida teknikak.

Taula 2: Gida sistemak AGV eta AMR-entzako (goiko bista) ([19]-ko irudian oinarrituta)

AGV			AMR			
Gidatze Mekanikoa	Gidatze Optikoa	Gidatze Induktiboa	Erreferentzia kartesiarra	Gidatze Inertziala	Laser gidatzea	Bisioan oinarritutako gidatzea

Aurreko taulan erreparatuz, AGV-en kasuan, marka eta erreferentzia puntu finkoak erabiltzen dira ingurunean zehar kokatuta [21]. Honen arazoa diru eta denbora inbertsioa da [22], ingurunearen aurre-analisi espezifiko bat egin behar delako fabrikaren barruan instalatu ahal izateko. Gainera, konfigurazioa diseinatu den ingurunerako soilik da erabilgarria, eta edozein aldaketak sisteman birkonfigurazio eta moldaketak egitea suposatzen du. Arrazoi honengatik, AMR-ak malguagoak dira, ez dutelako elementu zehatz horien beharrik, baina funtzionamendu algoritmoen konplexutasuna handiagoa da ere.

Aurrez aipatu den bezala, industriak garrantzi handia dauka ibilgailu autonomoen garapenaren eginkizunean. Industriak erabiltzen dituen ibilgailuak barrukoak dira eta hauek zenbait ezberdintasun dituzte kanpoko AV-ekin, honek suposatzen duen tekniken aldeekin. Aurkitu daiteken bereizketarik esanguratsuen errepideen erabilera da. Errepideek elementu arautuak izan ohi dituzte, legediaz araututa, eta hauek algoritmoen diseinuaren garapena errazten dute. Adibidez, bide-marken detekzioa norabidea definitzeko erabili daiteke. Barruko gidaketa autonomoan elementu bereizgarri horien gabezia dago eta garatu beharreko algoritmoek informazioa beste inguruneko elementuetatik lortu behar dute. Hala ere, kanpo gidaketa autonomoan erabiltzen diren teknologiak oinarri bezala erabil daitezke barruko AV-entzako.

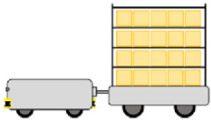
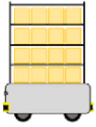

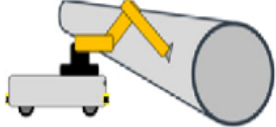


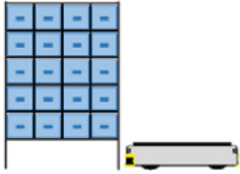
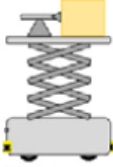

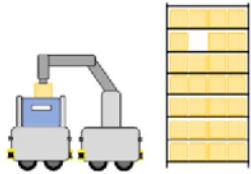
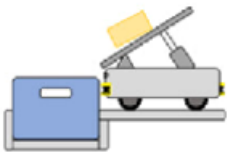
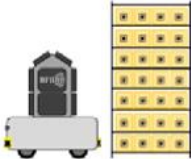
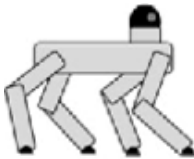

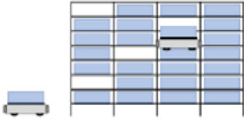

Industria 4.0 ibilgailu autonomoen sistemen garapen eta ospean influentzia handia izan du, gaiaren inguruko artikuluak sistema ziberfisiko eta hauen garapenean arreta jarriz [23]. Hiru azpimailatako sailkapen bat egin daiteke industriako AV-en ikerketa arloan. Alde batetik AV-en soluzio sistema ziberfisikoen egokipena dago. Horrela, Industria 4.0-aren ingurunean ibilgailuek izan behar duten eskakizun espezifikoak zehazten dira [24], [25]. Bestalde, ikerketa interes handiagoa sortu duen maila biki-digitalarena da, ibilgailu autonomoen sistemaren funtzionamendua hobetzeko intentzioarekin [26]. Horrela, nabigazioa gainbegiratu daiteke mugimendua kontrolatuz, lan ingurunea frogatuz eta gestioa eginez. Azkenik, AV-en erabilera automatizazio eta digitalizazio elementu bezala Industria 4.0-ren barne ikertzen dutenak dira. Helburua produkzioa, eraginkortasuna eta errentagarritasuna handitzea da. Multzo honetan sartu ohi dira lokalizazio, nabigazio eta kontrolaren inguruko artikuluak.

Esan beharra dago, industriako ibilgailu autonomoen erabilerak gizakia zerbitzatzeko burutua dagoela [27], lan fisikotan, lan arriskutsuetan edota lan errepikakorretan adibidez. Horrela, eginkizunen egitean hobekuntza bat ere sortzen da. Gakoa produkzio kooperatiboan dago, beste informazio sistemekin eta fabrikazio ekipoeekin, produkzioaren malgutasuna handitzeko [28]. Argi dago AV-en erabilerak lan sistemetan aldaketa sakonak suposatzen dituela, bai lantegi bai langileen aldetik,

industriako alor guztietan aplikagarriak bait dira eta egokitzapen prozesu bat behar delako hauekin batera lan egiten ikasteko.

Produkzioan, materialen manipulazioa lehenagien fluxuaren ezinbesteko parte bat da. Hauen garraio eta manipulazioa garrantzi handia dauka industrian eta ibilgailu autonomoak oso erabilgarri bihurtu dira zeregin hauetan [29]. Mugi dezaketen osagarri kopuruak, denbora laburrak eta lanpostuen artean piezen garraio azkarrak, produkzioaren igoera eragiten du. Gainera, hauetan robot kolaboratiboak akoplatzeko aukerak, zeregin askotan gizakiaren lana sinplifikatu du. Aldaketa hauek ekoizpen-lerro tradizionaletan malgutasuna eta eraginkortasuna sortu dute, lerro hauek interkonektatuz modu automatiko eta dinamiko batean. Kontuan izatekoa da, fabrikazio malgutasun horrek, enpresaren ahalmena handitzen duela bezeroen eskaeren denboren aurrean. Hurrengo **Taula 3**-an ibilgailu autonomoak eginkizun ezberdinetara aplikatuta ikus daitezke, fabrikazio eta biltegiatze lanetan hain zuzen ere.

Taula 3: Ibilgailu autonomoen erabilera ereduak industriaren barne ([19]-ko iruditik moldatua)

	Materialen manipulazioa		Lankidetzeta /elkarreragitea	Zerbitzu osoa
Fabrikazioa				
	Aingura/Atoia/Trena	Unitate propioa duen robota	Elkarlaneko robota	Erremintaz hornitutako beso robotikoa
				
	Altxatze ekipoa duen robota	Goiko garraiatzailea duen robota		
Biltegiatzea				
	Robotizatutako logistika mugikorrek instalazioetan eskaerak betetzeko robota	Biltzeko robota	Robot biltzaile kolaboratiboa	Robot biltzaile eta hartzailea
				
	Robot klasifikatzailea	Artikuluak aurkitzeko eta inbentariatzeko robota	Zaintza-robota	
				
	Puzzleetan oinarritutako biltegiatze sistemetakako robota	Ibilgailuak biltegiatze eta erauzteko sistema autonomoetako robotak	Eskaerak prestatzeko elkarlaneko robotak	

Argi dagoena da, teknologia sortu-berriek Industria 4.0-aren aroan sistema produkzio malgu berriak ahalbidetzen dituztela, eginkizunak sinplifikatuz.

- **KONPUTAZIO EBOLUTIBOA**

Adimen artifizialaren eremu zabalak, algoritmo ebolutiboak (*EA-Evolutionary Algorithms*) eta algoritmo genetikoak (*GA-Genetic Algorithms*) hartzen ditu. Nahiz eta bien artean ezberdintasun txikiak existitu, eboluzio biologikoan oinarrituta daude eta soluzio potentzialen eboluzioan dauden populazioen araberrako konponbideak garatzen dituzte. Horiek gaitasun funtzio baten bidez ebaluatzen dira. Askotan kostu konputazional altua duten algoritmoak dira eta hori arazo bat bezala ikus daiteke, baina parametroak doitzeko maiz erabiltzen dira.

- **LOGIKA LAUSOA**

Teknika honen izenak, egia partzialarekin lan egiteko duen arrazoibidea garatzeko ezaugarri nagusia islatzen du. *Fuzzy Logic*-aren abantailetakoa bat, kontrolatzaile bat diseinatzerako orduan ezagutza espezializatuak barneratzeko eta, beraz, hortik eratorritako kontrol eskemaren interpretagarritasuna egiteko ahalmena da.

2.1. Entrenamendu datuen garrantzia

Bai ikaskuntza automatikoan, bai ikaskuntza sakonean, garrantzi handia daukate entrenamendurako ematen zaizkion datuak, hauek bait dira emaitza egokiak izatea baldintzatuko dutenak. Jakin beharra dago, informazio kantitate handiek iragarpen zehatzagoak itzuliko dituztela “erakutsi gabeko” egoeren aurrean, beraz garrantzia du ere datu-basearen tamaina.

Bruin et al. [33]-en ikerkuntzan, tamaina honen inportantzia gain, datu-baseen antolamenduaren betebeharrak identifikatzen dira. Aipatzen duten bezala, oso interesgarria da egoera negatiboen garrantzia, hau da, aproposak ez diren politikak. Horrela, entrenamenduak distribuzioa mantentzen du emaitza anitzen artean.

Datu-baseek algoritmoen errendimendua baldintzatu dezakete, konparazio bide bat izanez. Gainera, Internet ate da irudi kopuru handi bat eskuratzeko, eredu desberdinen aplikazioetarako datu multzo bat eraikitzeo aukera emanez, ikusmen artifiziala erabiltzen duten algoritmoen jardueran hain garrantzitsuak direnak. Ikaskuntza automatikoan nahi diren datu motak erabili ahal dira, baina, nagusiki iragarpenak egiteko datu kategorikoak, numerikoak, denborazko-serieak eta testu eraokoak erabiltzen dira.

Datuak ikaskuntzarako baliatu aurretik prestakuntza prozesu batetik pasatu behar dira. Nagusiki, datuak bildu, garbitu eta handiagotu egin behar dira. Hasteko, datu bilketan, datu multzoa eraikitzen da, entrenamendurako. Garbiketan, zarata duten datuak iragazten dira, algoritmoaren errendimenduan eragina izan ez dezaten. Azkenik, handiagotzearekin ereduaren sendotasuna hobetu daiteke.

2.1.1. IKASKUNTZA TRANSFERITUA

Definizioz, *Transfer Learning*-a, beste problema batzuk ebazteko, arazoak ebazteari esker lortutako ezagutza transferitzeko metodoen multzoa da [34], [35]. Arazo berri batean, alde aurretik ikasitako eredu bat berrerabiltzeari deritzo, eta ikaskuntza sakonean bereziki ezaguna den teknika bat

da, datu kopuru txiki batekin sare neuronal sakonak entrenatzeko ematen duen ahalmenagatik. Ikaskuntza transferituaren gaitasuna bereziki garrantzitsua da datu-zientziaren alorrean.

Sare neuronal bat entrenatzerako orduan, neurona bakoitzak izan beharko duen pisu optimoa kalkulatu behar da, hau da, neuronen arteko konexioak erlazionatuko dituen balioa. Modu horretan, sare neuronalaren funtzionamendua definitzen da, konfigurazio hori gorde izanez beste eginkizun batzuetarako, bai arkitektura bai neuronek pisuak mantenduz [36]. Honek, sare neuronala etorkizun batean erabiltzeko aukera ematen du, sarea berriz entrenatzeko beharra izan gabe.

Gainera, jakina da sare neuronal baten entrenamenduak denbora asko suposatzen duela eta baliabide ugari behar direla konputazioaren aldetik. Aurrez entrenatutako sare neuronal baten erabilerak ondorioz, hau guztia aurrezten du. Modu honetan, jadanik entrenatuta dagoen ikaskuntza automatikoko eredu baten ezagutza, estuki erlazionatuta dagoen beste arazo bateri aplikatzeko aukera dago. Funtsean, lan batean ikasitakoa beste baten kontzeptuak hobeto ulertzen saiatzeko erabiltzen da. Konputazio koste altuena duten eginkizunak, orokorrean, ikusmen artifiziala aplikatzen den lanak dira eta hauetan da transferitutako ikaskuntzaren ohiko aplikazioa, batez ere jarraipen edo irudi klasifikazioaren ingurukoetan.

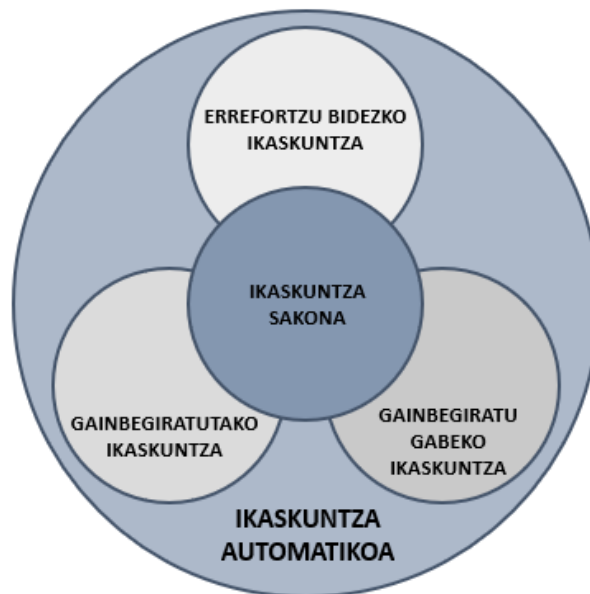
Transfer Learning-aren erabilerak nagusiki hiru zereginetan nabarmentzen dira [35]:

- Esan bezala, eredu bat entrenatu daiteke berrerabiltzeko, adibidez, ataza batek sare neuronal sakona osatzeko behar beste datu ez dituenean. Datu asko dituen beste lan bat aurkitzen da, aurrekoarekin zerikusia duena, eta hau entrenatzen da lehenengoa ebazteko. Bi lanek sarrera berberak izan ahal dituzte eta egin nahi den lanak berak erabakiko du zenbat geruza behar dituen beste eredutik eginbeharrekoa betetzeko.
- Beste aukera bat, jada entrenatuta dagoen sare bat erabiltzea da eta erabilpen hau ohikoena da ikaskuntza sakonean. Berrerabili eta entrenatu behar diren geruza kopurua eginkizunak berak zehazten ditu oraingoan ere, eta erakunde askok eskuragarri jartzen dituzte prestatutako ereduak.
- Azkenik, ezaugarrien ateratzean garrantzi handia dauka, eta ikusmen artifizialean bereizten da. Askotan, *Transfer Learning*-a ezaugarrien irudikapenetan erabiltzeak eskuz egindakoan baino emaitza hobekak eman ditzake eta automatikoki egitearen abantaila dauka.

Hiru zeregin hauek denbora eta baliabideak aurrezteko aukera ematen dute, zerotik entrenatzeko beharra ez dagoelako. Gainera, ikaskuntza sakonaren zenbait alorrean errendimendua hobetzen dute eta datuen gabezia konpentsatzen dute entrenamenduetan. Orokorrean ikaskuntza transferitua ohiko ikaskuntza sakonaren ereduen mugak gainditzeko barneratu zen, eredu berrien entrenamendu denbora eta errendimendua hobetzeko nahian.

2.2. Ikaskuntza automatikoaren teknikaren erabilera

Azken urteetan ibilgailu autonomoen algoritmo aurreratuenak sare neuronalen oinarritik sortu dira, errendimendu eta prezisio handiagoko emaitzak lortuz. Sare neuronalek ikaskuntza sakonaren funtsezko elementuak dira eta ML-ko teknika guztietan erabiltzen dira, baina lau prozesu desberdin burutu behar dira aurrez. Hasteko datuak kudeatu behar dira, non prozesatu egin behar diren, batuz, garbituz, zatituz, etab. Ondoren NN-en ereduak entrenatu behar dira, nahi den eginkizuna hautatuz eta behar den algoritmoa aukeratuz. Behin sare neuronala entrenatuta dagoela, lortutako ereduak ebaluatu behar da, lortutako emaitzak konprobatuz eta konparatuz. Azkenik ereduak zabaltzen da, aplikazio espezifiko batean exekutatzuz.



Irudia 5: Ikaskuntza automatikoa eta ikaskuntza sakonaren erlazioen eskema

Adimen artifiziala, ordenagailuei giza-adimena imitatzea ahalbidetzen duten teknikak dira. Ikaskuntza automatikoa aldiz, IA-ren nukleoa da (ikus **Irudia 5**) eta gainbegiratu gabea eta errefoztuzkoa biltzen ditu. ML-a sarrera datuen bitartez bere kabuz ikasteko entrenatuta dago eta ibilbidearen egoera optimoena aurkitu, larrialdietan erantzuna eman edo ingurunea topa dezake, adibidez [37].

Gainbegiratu gabea ikaskuntzaren algoritmoek emandako datuen multzoarekin lan egiten dute, erregresioa edo klasifikazioa eginez. Klasifikazioan esaterako, entrenamendu-datuen multzoa balio diskretuak ditu. Horrela, oztopoak, bidea edo edozein elementu esanguratsu identifikatu daiteke. Gainbegiratu gabekoa aldiz, entrenamendua etiketatu gabeko datuetan oinarritzen da eta honek konplexutasuna gehitzen die algoritmoei. Azkenik errefoztuzko ikasketa dago, non agente bati ematen zaion saria erabiltzen den ikasteko, ibilbideen planifikazioa ahalbidetuz.

Ikaskuntza sakona (DL-*Deep Learning*) azken hiru hauen nahasketa bezala definitu daiteke, eta funtsean sare neuronalekin artifizialekin (ANN- *Artificial Neural Network*) funtzionatzen du, sare neuronal bezala ere ezagunak direnak. Ohiko sare neuronaletako batzuk sare neuronal konboluzionala (CNN-*Convolutional Neural Network*) edo sare neuronal errepikaria (RNN-*Recurrent Neural Network*) dira adibidez eta oso erabilgarriak dira garraio autonomoen algoritmoetan, giza-burmuin bat bezala lan egin dezaten entrenatzen bait dira, garraioa osatzeko gaitasuna emanez.

2.2.1. GAINBEGIRATUTAKO IKASKUNTZA

Gainbegiratutako ikaskuntzan datu multzo bat erabiltzen da entrenamendua egiteko, ereduie nahi den irteera nola lortu erakutsiz. Gainbegiratutako ikaskuntzaren ohiko tekniketako bat sailkapena da. Teknika honek objektu baten klasea edo kategoria iragartzen du, etiketak eraginez. Objektua kategoria multzo baten barruan klase bakarrean egon daiteke, neurri edo ezaugarrien arabera adibidez. Beraz, helburua sailkapena egin dezaken eredu bat entrenatzea da, objektu bakoitza ondo klasifikatzeko ahalmenarekin .

Literaturan ikus daiteke teknika hau ez dela oso erabilia bere baitan ibilgailu autonomoen aplikazioetarako. Gehien bat bidean topa daitezkeen oztopoen identifikazio lanetan aplikatua izan da. Li et al. [38]-en hitzetan, ingurune ko objektu garrantzitsuen identifikazioa aurretiko baldintza da erabakien hartze segurua eta kalitatezko bat hartzeko, mugimenduaren planifikaziorako. Bi klase sortzen dituzte, "garrantzitsua" edo "ez-garrantzitsua" sailkapena eginez. Gainera, erdi-gainbegiratutako entrenamendu bat proposatzen dute, eredu etiketatu gabeko datu mugagabeetatik ikas dezan. 3D-ko oztopoen inguruan, Zhang et al. [39]-ek erdi-gainbegiratutako eredu bat proposatzen dute ere etiketak automatikoki sortzeko klase gehiago existituko direla kontsideratuz.

Gainbegiratutako ikaskuntzarekin jarraituz, erabiltzen den beste tekniketako bat erregresioa da. Erregresio algoritmoaren zeregina mapatze-funtzioa aurkitzea da, sarrera aldagaiak mapatuko dituen irteera aldagai jarraientzako. Rahman et al. [40]-ek erregresio orokorreko sare neuronal bat erabiltzen dute kokapena estimatzeko haririk gabeko sentsore sare (WNS-*Wireless Sensor Network*) batekin batuta , emaitza onak lortuz nodoen bitartez.

2.2.2. GAINBEGIRATU GABEKO IKASKUNTZA

Gainbegiratu gabeko ikaskuntzako ereduak, algoritmoak etiketatu gabeko datu batzuetatik ikasten du, zentzua izan dezaten ahaleginduz, ezaugarriak edo patroiak identifikatuz adibidez. Elkartzea da gainbegiratu gabeko ikaskuntzaren tekniketako bat, non klasifikazio etiketak ez dauden eskuragarri, beraz elkarketak sortu behar dira. LiDAR (*Light Detection and Ranging*) sentsorea erabiliz, 3D puntu-laino bat sortu daiteke, iragazi ostean elkarketak egiteko, oztopoak detektatuz (ikus Li et al. [41]). Domínguez et al. [42] gainera puntu-lainoan segmentazioa aplikatzen dute elkarketen aurretik. Honek elkarketa askotara eraman ahal duela diote Jiang et al. [43]-ek, distantzia filtro bat gehitzen dutenak gain-elkartzea ekiditeko.

Balaska et al. [44]-ek, oztopoak topatu ordez, lokalizazioa egitea proposatzen dute, gainbegiratu gabeko teknika semantikoekin. Hau, grafikoetan oinarritutako segmentazioaren bidez lortu dute, ibilgailuak bere ingurunea semantikoki definituz. Elkartze semantikoak, Louvian elkartearen algoritmoaz baliatzen dira neurri antzekoak dituzten taldeak identifikatzeko.

2.2.3. ERREFORTZU BIDEZKO IKASKUNTZA

Errefortzu bidezko ikaskuntzan (*RL-Reinforcement Learning*) agente batek ingurunearekin elkar-eragiten du eta bere errendimendua hobetzen ikasten du horri esker. Hobekuntza hori, sari metatzearekin lortzen du. Horrela, erabakiak sekuentzian hartzeko ahalmena dauka, denbora errealean. Zenbait elementu izan behar dira kontutan mota honetako ikasketetan. Alde batetik egoeren multzoa dago $s(t_k)$ bezala definitzen dena. Akzio multzo bat ere definitzen da $a(t_k)$ bezala ezaguna dena. Trantsizio funtzio baten beharra ere badago, U bezala definituz eta errefortzu funtzioa $reward(s,a)$, hau da, sariak erabakiko dituen funtzioa. Ikaskuntza antzezle π eta kritikoa Q sareen bidez egiten da, eredu bat emanez. Horrela, hurrengo egoerak $s(t_{k+1})=U(s(t_k),\pi(s(t_k)))$ balioa izango du non,

$a(t_k)=\pi(s(t_k))$ den. Sariak aldiz, kritiko sarearekin kalkulatu dira $Yreward(t_k)=Q(s(t_k),\pi(s(t_k)))$ erara, γ deskontu faktorea izanez.

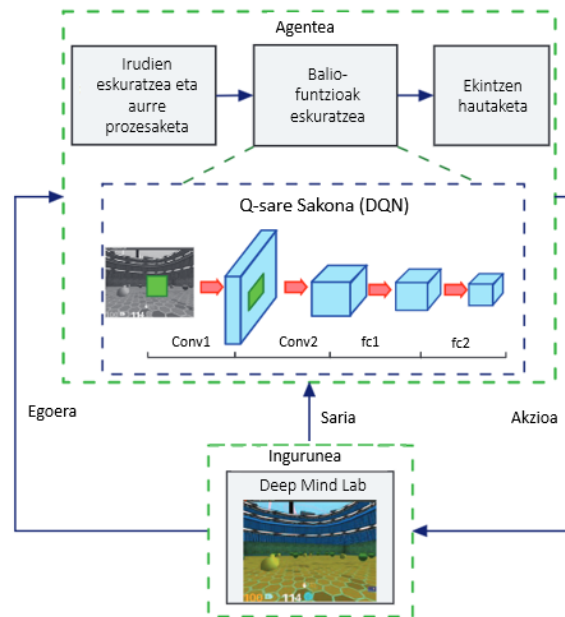
Aipatzekoa da, *Q-Learning*-a errefortzu bidezko ikaskuntzaren algoritmo bat dela, baina egoera jakin baterako akzioaren balioa ikasteko eredurik ez daukana. Modu honetan, trantsizio eta sari estokastikoen arazoak ebazteko gai da adaptazioen beharrik gabe.

Orokorrean, errefortzuzko ikaskuntzarako zenbait zailtasun existitzen dira. Hauetako bat, dimentsionalitatearen arazoa da. Horrela, sistema konplexutan aplikatu nahi bada, moldakortasunaz gain efizientzia konputazionala ere hartu behar da kontuan. Kondo et al. [45]-ek egoerak biltzeko antzele-kritikoaren egokitze-estrategia bat proposatzen dute NGnet sarea oinarri duena. Sistemak, horrela, bere egoera-espazioa berrantolatu edo zatitu dezake eginkizunaren zailtasunaren eta ikaskuntzaren aurrerapenaren arabera

Hau kontuan izanda, ibilgailu autonomoen eginkizunetan eraginkorra den ikaskuntza mota bat da. Aldibereko mapatze eta lokalizazioari dagokionez, eredu gabeko errefortzuzko ikasketa sakonaren bidez egin daiteke Placed et al. [46]-ek dioten bezala. Honetarako, sarietan, diseinu esperimental optimoaren teoriaraino oinarritutako erabilgarritasun-funtzio tradizionalak integratzen dituzte, ikuspegi klasikoaren kalkulu intentsiboak erlaxatuz. Horrela, entrenatutako agenteak nabigazio eta esplorazio politika bat eredu gabe ikasteaz gai izateaz aparte, beraien ezagutzak transferitu dezakete aurrez ikusi ez dituzten mapetan. Kontuan izan behar da, hau bi dimentsiotako inguruetan dela aplikagarria, hiru dimentsiotan, behagarritasun partziala edota saria duten inguruen gabezia suertatu daitekeelako. Beraz, abstrakzioa kontsideratu behar duen agente bat erabili beharra dago [47], non ikaskuntza sakoneko Q-sare (*DQN-Deep Q-Network*) baten datu-baseari objektu edo egitura-elementuen xehetasunak gehitu behar zaizkion.

Ibilbidearen kalkulua egiteko ere erabilgarria da errefortzuzko ikasketaren metodoa. Gao et al. [48]-en ikerkuntzan entrenamendu modu inkremental bat proposatzen dute DRL-an (*Deep Reinforcement Learning*) oinarrituta ingurune arin batean. Lehendabizi bilaketa grafikoaren algoritmoak eta errefortzuzko ikaskuntzaren algoritmoak ebaluatzen dituzte, ondoren DRL-ko algoritmoa diseinatzeko behaketa egoerak, sari funtzioa, sarearen egitura eta parametroen optimizazioa barneratuz. Gainera, hiru dimentsiotara transferitzeko gai dira, emaitzetan errendimenduaren hobekuntza detektatuz. Egia da baina, inguru arinetan informazio galera existitzen dela, hauek eskuz diseinatzen bait dira eta dimentsio baxukoak direlako. Hala ere, dimentsio altuko mapa bat sarearen sarrera bezala ere erabiltzeko aukera dago, ibilbide planifikazio politika arrakastatsu bat izanda AV-entzako (ikus *Kamoshida et al. [49]*).

Errefortzuzko ikaskuntza sakona eskuz karakterizatu gabeko ingurune batean erabiltzeko, Xin et al. [50]-ek DQN sare bat diseinatzen dute robotaren egoera-akzio balioen funtzioa gerturatzeko. Akzio bakoitzeko mugimendu balioak sareak determinatzen ditu, sarrera RGB irudi bat izanez. Horrela oztupoak ekiditen dituen ibilbide bat aurreikusi daiteke ibilgailuaren mugimendu optimoak errespetatuz. Egindako proposamena hurrengo **Irudia 6**-an ikus daiteke.



Irudia 6: Robot mugikorraren ibilbideak planifikatzeko eskema orokorra ([19]-ko iruditik moldatua)

Argi dago oztopoen ingurunearen konplexutasuna handiagotzen dutela. Kasu hauetan, Dueling DQN moldaketa ere erabilgarria da ibilbidea zehazterako orduan, objektu bai dinamikoak bai estatikoak aurrean Wen et al. [51]-ek frogatu duten bezala. Oztopoen erreferentzia puntu bezala erabiltzen dituzte Wen et al. [52]-ek, laser sentzore batek jasotako datuak multzokatzen dituen algoritmo bat erabiliz RL-aren aurretik, kokatzeko aukera lortuz ingurune ezezagunetan.

Kontuan izan beharrekoa da *Q-Learning* algoritmoek, esan bezala, eredu gabe ikasteko abantaila dutela, baina horrek emaitza optimoa lortzean konbergentzia motela duela. Muga hori gainditzeko, partzialki gidatutako *Q-Learning* kontzeptua proposatzen dute Low et al. [53]-ek, lore-polinizazio algoritmo (FPA-Flower Pollination Algorithm) bat erabiliz *Q*-ikaskuntzaren hastapena hobetzeko. Antzeman dutenez, *Q-Learning*-aren konbergentzian bizkortze bat existitzen da metodo honekin. Optimizazioarekin jarraituz, baina ibilbidekoarekin, Khamidehi et al. [54]-ek ibilgailuaren bidaia denbora minimizatzea bilatzen dute, betiere base batekin komunikazioa mantenduz. *Q* bikoitzeko ikaskuntza sakonaren (DDQN-*Double Deep Q-Learning*) metodoa erabiltzen dute, honek gainestimazio arazoak ez dituelako.

Robotaren beharrak kontuan izanda, badaude zenbait funtzio burutu ezin direnak eta Yu et al. [55]-ek errefortzu bidezko ikaskuntza hierarkikoa (HRL-*Hierarchical Reinforcement Learning*) proposatzen dute hauek bete daitezela. Honetarako, egin ezin diren akzioen uneko egoeraren mapaketa egiten dute HRL-aren bidez, ibilbide planifikaziorako eredu eraikitzaile. Gainera metodoen konparaketa egin dute, DDPG (*Deep Deterministic Policy Gradient*) ikaskuntza hierarkikoan oinarrituta dagoen algoritmoa eta DDQN algoritmoaren artean, non lehenak ibilbide planifikazio denbora laburragoa duela ziurtatuz.

Beste alde batetik, Botteghi et al. [56]-ek errefortzuzko hiru ibilbide planifikatzaileen prestazioak ebaluatzen dituzte, ez bakarrik entrenamendu inguruneetan, baizik eta ikusi gabeko eremu batean hauen generalizazio politiken ezaugarriak frogatzeko. Emaitzek, errefortzu bidezko ikasketarekin entrenatutako algoritmoak barruko inguruneetako mapak esploratu eta eraikitze gai direla esaten dute, modu egonkorrean. Gainera, etorkizuneko robotaren mugimenduetan atentzia jarritz, *Q* balioak kalkulatu daitezke uneko eta hurrengoko sari balioak erabiliz, denbora pausuko. Errefortzuan oinarrituz, ingurunea ikasi daiteke eta nabigazio ibilbidea aukeratu mugara helarazteko

(ikusi Tamilselvi et al. [57]). Hau Markov-en erabaki prozesuaren bidez egiten da, biderik optimoena aukeratuz.

Badago beste ikerkuntza lerro bat non mugimendu zinematika planifikatzeko *online* esparru algoritmiko bat erabiltzen den, asintotikoki optimoa den RRT (*Rapidly-exploring Random Tree*) algoritmoa maneiatuz denbora jarraiko Q-ikaskuntzarekin batera (ikusi Kontoudis et al. [58]). Eredugabeko Q-n oinarritutako abantaila funtzio bat formulatzen da eta errefortzuko ikaskuntza integrala erabiltzen da koste optimoaren *online* hurbilketa doikuntza legeak eta denbora jarraiko sistema linealen politika optimoa garatzeko. Gainera, Lyapunov-en bidez egonkortasuna ziurtatu daiteke. Horrela, oztopoak areagotzeko eta lekuko birplangintza egiteko esparru estatiko baten bidez, robotaren ibilbidea birkalkulatu daiteke talka gabeko nabigazio bat bermatuz.

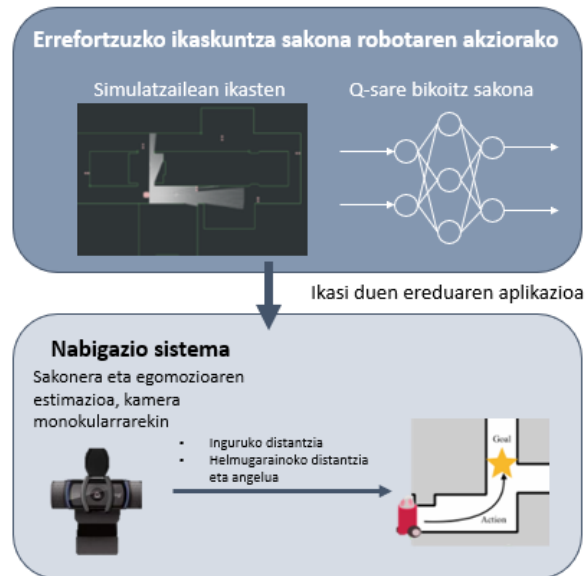
Behin ibilbidearen estimazioa egina dagoela, nabigazio algoritmoa aplikatu behar da. Honek mugimenduen planifikazio algoritmo bat garatzea suposatzen du ARM-etarako, non Van et al. [59]-ek errefortzuko ikaskuntza sakonaz baliatzen dira nabigazio adimentsu bat gauzatzeko bi dimentsiotako ingurunean, eraginkortasuna eta prezisioa lortuz. Honetarako, ROS (*Robot Operating System*) gainean eraikitako algoritmoa proposatzen dute. Politika proximalaren optimizazioaren (PPO-*Proximal Policy Optimization*) bidez oinarritutako errefortzu bidezko ikasketa sakonean ere entrenatu daitezke agenteak nabigazioa gauzatzeko (ikusi Sadhukhan et al. [60]). Sari funtzioa norabidearen arabera izanda, hau da, desbiderapen erroreak aintzat hartuz, agente bakoitza mugara heltzea lortzen da.

Nabigazio eremuan objektuak daudenean, Staroverov et al. [61]-ek objektu bidezko nabigazioa proposatzen dute non ibilgailua oztopo horietara bideratuko den inguruan ikuspen mugatua badago. Honekin algoritmo sendo eta optimo bat bilatzen dute, HRL-an eginkizunaren formulazio estandarrak aplikatuz eta objektuak, ingurune ezagutza gehigarri bezala, erreferentziatzen hartuz. Hierarkiaren maila baxuek intuiziozko trebetasunaz gaitzen dute, altuek aldiz, une oroko beharren erabakia hartzen dute. Hierarkia maila berdinak erabiliz, inguruneko objektuak semantikoki definitu eta horietara nabigatzea proposatzen dute Staroverov et al. [62]-ek, bi puntuen arteko sistema modularren errefortzuko ikasketetan azkenaldiko lorpen esanguratsuek kontuan izanda. Baina berriz ere, aurrerapauso handi baten beharra antzematen da sistemak sendoak izan daitezken. Esplorazio eskasari aurre egiteko eta semantikoki zentzua izan dezan, agenteari iriste errazeko erreferentzia puntuak ematen zaizkio, horrela agenteak politiken egitura hierarkiko bat eraiki dezan.

Aurretik zehaztu gabeko ibilbide edo mapa gabekiaz ere entrenatu daitezke errefortzu bidezko ikaskuntzako sareak, ibilgailu autonomoa ingurune murrizketei egokituz. Hala ere, praktika honek egonkortasun falta dauka entrenamenduetan. Gainera, oso garrantzitsua da mapa gabeko nabigazioaren errendimendurako beti ingurune espezifikoko batean egitea. Toan et al. [63] PPO-ko sare konboluzional bat eraikitzea proposatzen dute mapa gabeko nabigazioaren arazorako. Gainera Boltzmann politika gehitzen dute balantza bat sortzeko esplorazio eta esplotazioaren artean, robota ingurune konplexuak esplotatzen laguntzen duena. Dolu-sareen arkitekturan oinarrituta Ruan et al. [64]-ek mapa gabeko nabigazioan ere egin dute ikerkuntza. *Dueling DQN* eta Q bikoitzaren errefortzuko ikaskuntza sakona erabiliz, sare bat egokitzen dute. Horrela, ibilgailuak inguruko ezagutza jaso dezake gradualki, nabigazioa eginez RGB kamera baten laguntzaz.

Marchesini et al. [65]-ek aldiz, espazioko egoera diskretuen algoritmoak mapa gabeko nabigaziorako soluzio diren aztertzen dute. DDQN bidez entrenamendu asinkrono paralelo bat eta sorta anitzeko *Priority Experience Replay* bat proposatzen dute entrenamendu denbora laburtzeko. Horrela DDPG eta PPO algoritmoak gaintzea lortzen dute, entrenamenduan %40-ko azkartasuna lortuz besteen aurrean, arrakasta maila antzekoa mantenduz. *Deep Q-Network* bikoitzarekin batera

kamera monokular bat ere erabili daiteke, Yokoyama et al. [66]-ek adierazten duten bezala (ikusi **Irudia 7**). Kamerarekin irismen datuak estimatzen dira sarearen sarrera izateko. Lortutako sakontasun estimatuko irudiak bi dimentsiotako irismen datutan bihurtzen dituzte, 2D-ko planoan oinarrituz ikasitako ereduaren sartzen direnak.



Irudia 7: Kamera monokular era Q-Sare sakon bikoitzarekin garatutako nabigazio sistema([66]-ko iruditik moldatua)

Surmann et al. [67]-en hitzetan, errefortzu bidezko ikasketetatik ateratako ereduak oraindik segurtasun eta sendotasun falta dute edo/eta ingurune egituratu baten falta daukate. Horretarako, mapa eta planifikatzaile gabeko robot baten nabigaziorako autoikaskuntza proba aurkezten dute. Ibilgailuaren sarrera, bi dimentsiotako laser eskaner bat RGB-D kamera batekin fusionatutako informazioa da, helmugaren orientazioarekin batera. Sarrerak aldiz, abiadura lineal eta angeluarra dira, *Asynchronous Advantage Actor-Critic* sare batetik eskuratuak. Gainera, gehiegizko doikuntza ekiditeko, sare txikiak entrenatzen dituzte, zorizko soinu Gaussiarra gehituz sarrera laserrari. Laser bidezko irismeneko 10 datu DDPG sare bateko sarrerak izan ahal dira ere, aurretiko abiadura lineal eta angeluarra eta robotaren posizio eta angelu erlatiboekin batera (ikusi Jesus et al. [68]). Horrela ikus daiteke errefortzu bidezko ikaskuntza, akzio jarraituekin, eraginkorra dela ibilgailu autonomo baten erabaki hartzearekin. Arazoa betiere, sari sistema on bat sortzea da.

Interbentzioz lagundutako errefortzuzko ikasketak (*IARL-Intervention Aided Reinforcement Learning*) robotaren inguruko interakzioa gizakiarenarekin batera erabiltzen du politika hobetzeko (ikusi Wang et al. [69]). Sare neuronalak sentsoreen irakurketak kontrol seinaleetan bihurtzeko politika bezala erabiltzen da, nabigazioaren errendimendua hobetuz eta aldi berean, segurtasuna bermatuz.

Helmuga, seinaleak bidaltzen dituen dispositibo bat izatea proposatzen dute Chowdhury et al. [70]-ek, periodikori irrati-frekuentzia seinaleak bidaliz hari-gabeko sarrera puntu gertuetara. Ibilgailuaren posizioa eta jasotako seinaleen intentsitatea (*RSS-Received Signal Strength*) *Q-Learning* algoritmoraren sartzen dituzte, helmugako inguruetara nabigatuz. Modu honetan ez da ingurunearen informazioa behar, seinaleen intentsitateak sortutako bero-mapetan oinarritzen bait dira robotaren mugimenduak.

Khriji et al. [71]-ek situazio errealean oztopo askok parte hartzen dutela diote, eta horrek *Q*-ikaskuntzaren ohiko ikuspuntuak bi arazo nagusiekin topatzea eragingo duela neurritz gainera egoera-espazio baten ondorioz. Alde batetik, tabula-formako *Q* balioen ikasketa bideraezina izan daiteke

gehiagizko memoria baten beharrik. Bestetik, sariak egoera-espazioan eskasak izan daitezke eta oso motel begiztatuko ziren zorizko esplorazio batekin. Traba hauek gainditzeko aurre ezaguera Q-Ikaskuntzan erabiltzen den ikuspegi bat proposatzen dute. Jokaeran oinarritutako nabigazioa akzio txiki sinpleagotan banatzen dute logika zehaztugabearen bidez. Ondoren, mugimenduak koordinatzeko errefortzuzko ikasketa aplikatzen da, ikaskuntzaren konbergentzia denborak murriztuz.

Atzeman den bezala oztopo saihestea nabigazio egoki baten ezinbesteko parte da. Ribeiro et al. [72]-ek *Q-Learning*-eko bi ikuspuntu proposatzen dute oztopoak ekiditeko, hiperparametroen azterketa zabal batekin batera. ROS komunikazioa erabiliz kontrola eta robotaren artean iterazio-komplexutasuneko hiru labirintotan nabigatzeko gai dira. Eraitza horietara heltzeko baina, hiperparametroak oso kontutan izan ziren ehundaka simulazio eginez. Bi Q-Ikaskuntza emaitzek estrategia ezberdinak garatzen dituzte, ikaskuntza ahalmena indartuz eta situazio ezberdinetarako hurbilketak eginez, azkenik ingurune konplexutan talka gabeko nabigazioa gauzatzuz. *Q-Learning*-ak balioak biltegitartzeko kapazitateaz baliatuz ordea, Huang et al. [73]-ek robotaren izaera autonomoa bermatzen dute ere.

Oztopo dinamiko eta estatikoei aurre egiteko Duguleana et al. [74]-ek robotak talka gabeko ibilbide batekin hornitzea ikertzen dute, lan eremu ezezagun batean. Q-Ikaskuntza eta ibilbide arazoak ebazteko sare neuronal planifikatzaile batekin, talka gabeko nabigazioa egin daitekeela frogatzen dute. Gainera, robotaren abiadura aurrez zehazteko aukera ematen dute, denbora murriztuko aplikazioetarako. Chewu et al. [75]-ek aldiz, SLAM-etik lortutako informazioaz baliatzen dira. Kinect sensore bat erabiltzen dute mapa eraikitze eta kodetzaile birakari bat lokaliziorako. Errefortzuzko ikasketak horrela, oztopo dinamikoak ekiditen dituen nabigazio bat egiteko gai da, ibilbidea birplanifikatuz eta optimoagoa den bat bilatuz, aurrez ezarritakoa moldatuz.

Deep Q-Learning teknikari logaritmoetan oinarritutako sari balioaren eremu funtzio bat sartzen diote Mohanty et al. [76]-ek, hau da, agenteak jasotzen duen saria bere posizio erlatibo, oztopo eta helmuga kontuan dituen funtzio bat. Arrakasta tasa altua duen algoritmo bat aurkezten dute horrela, oztopoak ekiditerako orduan. Wicaksono et al. [77]-ek Q-Ikaskuntza, ikaskuntza mekanismo gisa erabiltzen dute ibilgailu autonomoaren nabigazio portaerak barnertzeko. Horrela, konportamenduetan oinarritzen dira inguruaren elementuetan egin ordez.

Badaude zenbait eginkizun roboten kolaborazioa ezinbestekoa dena. Honek industriaren arloan konplexutasuna gehitzen du, produkzioan flexibilitatea eskatzeaz gain. Popper et al. [78]-ek esaten dutenaren arabera, sortzen den lan denboren programazio malguaren arazoa, AV-en koordinazioaz batera zaila da optimizatzeko. Agente anitzeko errefortzu bidezko ikasketa proposatzen dute honetarako, eginkizun eta ibilgailuen programazioa modu kooperatiboan eginez. Takahashi et al. [79]-ek aldiz, errefortzuzko metodologia ebaluatzeko simulazioz robot bat entrenatzen dute lehendabizi eta sare horrekin beste ibilgailu bat optimizatzen dute.

Corrales et al. [80]-ek errefortzuzko ikasketa, erlekume heterogeneo robot sistema batean aplikatzen dute politika ezberdinak lortzeko, zeregin logistikoetarako. Bi agente ezberdintzen dituzte, pakete garraiatzaileak eta ibilgailu kopurua arautzen dituzten semaforoak. Horrela bi kontrol politika ikasten dira simultaneoki, agente bakoitzerako, garraioaren izaera eta trafiko fluxua erregulatu ahal izanez talka gabeko elkarlaneko nabigazioan. *Online* ibilbide planifikazio eta exekuzio metodoa proposatzen dute Kozjek et al. [81]-ek. Aurrez kalkulaten dute ohiko jasotze eta entrega kokapen arteko ibilbidea, AV-en mugimendua mugatuz talka egin ez dezaten. Metodo honek errendimendu eta fidagarritasunean emaitza onak eskaintzen ditu, ohiko ibilbide motzeko tekniken aurrean.

2.2.4. IKASKUNTZA SAKONA

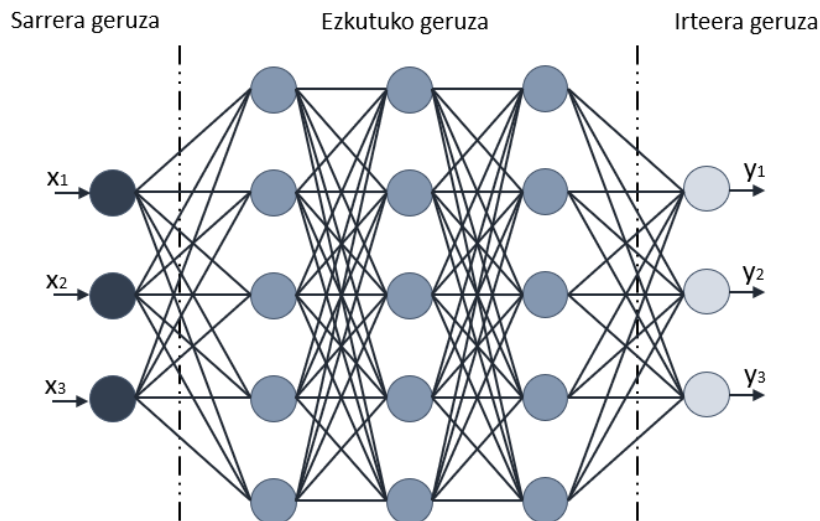
Deep Learning-a *Machine Learning* mota bat da, prozesamendu geruza ugari osatutako eredu konputazionalak abstrakzio maila ugari datuen errepresentazioak ikastea ahalbidetzen duena. Datu-multzo handietan barneratutako egiturak aurkitzeko ahalmena du, atzerahedapeneko (*Backpropagation*) algoritmoa erabiliz barne parametroak nola aldatu behar diren adierazteko, hau da, aurreko geruzako irudikapenetik abiatuta geruza bakoitzeko irudikapena kalkulatzeko erabiltzen diren parametroak [82].

Gidatze autonomoaren zereginetan ere ikaskuntza sakonaren bidez garatutako algoritmoak aplikagarri izan dira. *Deep Learning*-ak ordea, ez dauka egitura bakarra beraz modu asko begiztatu dira AV-en gabeziei aurre egiteko.

Deep Neural Network arkitektura

Sare neuronal sakonak (DNN-*Deep Neural Network*) konplexutasun maila bat duten sare neuronalak dira, hau da, bi geruza baino gehiago dituen sare bat. Beraz, eredu matematiko sofistikatuak erabiltzeko ahalmena dute forma konplexutako datuak prozesatzeko. Hala ere, sare neuronal sakonak benetan ulertzeko, hobe da eboluzio bat bezala interpretatzea, elementu ezberdinak eraiki behar izan zirelako DNN-ak sortu aurretik.

Orokorrean, *Deep Neural Network*-en definizioa, sarrera geruza bat, irteera geruza bat eta, gutxienez geruza ezkutu bat duten sareak bezala uler daiteke, **Irudia 8**-an eskematikoki errepresentatzen den erara. Geruza bakoitzak sailkapen eta antolamendu espezifikoetaz arduratzen da, "eginkizunen hierarkia" bezala ezaguna dena. Sare neuronal sofistikatu hauen funtsezko erabilerak etiketatu edo egituratu gabe dauden datuak tratatzearekin dute erlazioa eta, esan daiteke, sare neuronal konplexuen oinarria dela arkitektura hau.



Irudia 8: Deep Neural Network sare neuronalaren arkitektura

Lokalizazioan hari gabeko metodoek dakartzaten seinale aldagarri eta iragarrezinei aurre egiteko erabili daiteke DNN bat, adibidez (ikus Zhang et al. [83]). Modu honetan, posizionamendua aurrez entrenatutako lau geruza dituen sare neuronal sakon batean oinarritu daiteke, soinua deuseztatuz eta ondorioz, zarata asko dute lagin multzo handi batetik ezaugarriak ikasteko ahalmena izanez. Gainera, denbora koherentziari eusteko, Markov ezkutuaren eredian oinarritutako lokalizatzaile fin bat erabili daiteke, DNN-an oinarritutako lokalizatzaile lodiak lortutako posizionamenduaren zenbatespena leuntzeko.

Path planning-a aurrera eramateko Wu et al. [84]-ek sare neuronal sakonetan oinarritutako metodo bat proposatzen dute, ingurune desordenatu ezezagunetan denbora errealean ibilbideak *online* planifikatzeko. Muturretik muturrerako DNN arkitektura diseinatzen dute, ibilbide plangintza lokalak egiteko politikak ikasi dezan. Chai et al. [85]-en kasuan, proposatzen duten esparrua bi mailatako egitura batean dago banatuta: alde batetik, ibilbidearen optimizazio desentsibilizatua egiten dute eta bestetik, sare neuronal sakonetan oinarritutako kontrola. Goiko mailan, sistemaren kontrol optimoa eta egoera ibilbideak dituen ibilbide datuen multzo bat sortzen da. Beheko mailan aldiz, DDN-ak aurreko mailako ibilbideen multzoa erabiliz entrenatzen dira, sistemaren egoera optimizatuen eta kontrolen arteko erlazio funtzionala irudikatzen.

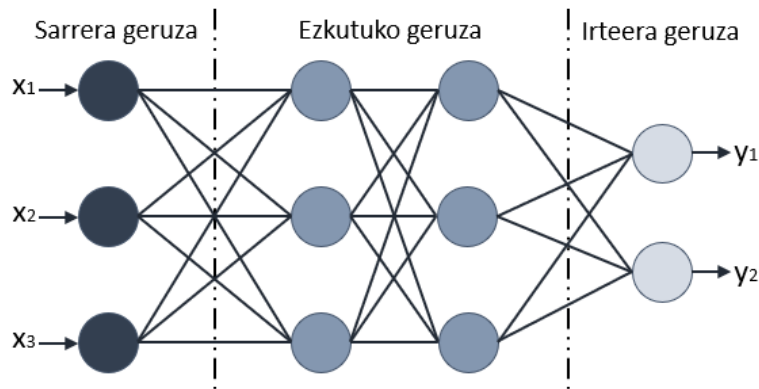
Lai et al. [86]-en lanean, denbora errealeko kontrol-ikuspegi optimoa aztertzen da, DNN-tan oinarrituta, ingurunean oztopoak dituen AGV baterako. Ibilgailu autonomoaren ibilbide plangintzaren arazoa kontrol optimoko problema bat bezala planteatzen da, emaitza numeriko bat lortuz. Horretarako, lehendabizi Gauss-en metodo pseudoespektrala erabiltzen da, leuntze teknika bat barneratuta duena. Ondoren, sare neuronal sakon bat garatzen da, aurreko pausoaren bidez lortutako egoera eta ekintza bikote optimoen arteko erlazio funtzionalak ikasteko. Modu honetan, DNN-ak denbora errealean AGV-aren nabigaziorako kontrol-jarraibide optimoak sor ditzake. Ren et al. [87]-ek aurreko hauen ildo beretik, denbora errealerako kontrol bat ere proposatzen dute, DNN-ek ibilgailu autonomoen nabigazio kontrolaren autonomia eta adimena hobea dezaten. Mugimendu planifikazioaren arazoa kontrol optimo ez-lineal bat bezala planteatzen dute ere, baina kasu honetan, oztopo dinamikoak eta estatikoak aurki daitezke ingurunean. Egoera-ekintzen lagin optimoak lortzen dira eta datu-multzo bezala gordetzen dira DNN-a entrenatzeko. Ondoren, sare neuronal sakonaren arkitektura diseinatzen dute eta hau entrenatzen da erlazio optimoak ikasteko.

Nabigazioarekin jarraituz, Chai et al. [88], ibilgailu autonomoetarako, mugimendu komando optimoak aurreratzeko gai den sare neuronal sakon baten diseinua aurkezten dute. Proposatutako ereduak geruza anitzeko egitura erabiltzen du, lehenengo geruzan ibilbideak optimizatzen metodo desentsibilizatu bat erabiliz iteratiboki. Behin ibilbideak finkatuta daudela eta DNN-en errendimendua hobetzeko, datuak gehitzen dituen metodo bat diseinatu eta aplikatzen dute. Sareak entrenatuta daudenean, mugimendu kontrolatzaile gisa erabiltzeko aukera ematen dute, denbora errealean aginduak sortzeko. AGV-ak inklinatuta dauden inguruneetan ere nabigazioa egin dezakete sareak portaera eredu ikasi ezker (ikusi Badgujar et al. [89]).

Sare hau oso mugatuta geratzen da zenbait eginkizunetarako eta “sakontasunaren” kontzeptua beste arkitekturetan barneratzen da, ikerketa landuagoak aurrera eramateko aukeraz, hurrengo ataletan antzemango den bezala. Horregatik, ibilgailu autonomoen arloan ez da egungo bibliografia ugari aurkitzen sare neuronal sakonen erabilera hutsarekin.

Feedforward arkitektura

FeedForward (FNN) sareak, geruza anitzeko pertzeptroi (MLP-*Multilayer Perceptron*) bezala ere ezagunak direnak, ikaskuntza sakoneko ereduak dira, non f^* funtzio baten hurbilketa egitea duten helburu bezala [30], Φ parametroen balioa ikasiz. Horretarako, sare mota honek mapaketa funtzio bat definitzen du $y=f(x, \Phi)$ erakoa, non x sarrera datuak diren eta y irteera datuak, f eta f^* berdina izan daitezke. Eredu hauek ordea ez dute atzeraelikadurazko konexioak egiteko aukera eta egitura **Irudia 9**-an ikusi daitezke, ezkutuko geruza gehiago, sarrera edo irteera izan ahal izanez.



Irudia 9: Feedforward sare neuronalaren arkitektura

Feedforward sareak ibilgailu autonomoen lokalizazioa egiteko erabili dira. Krapež et al. [90]-ek *Feedforward* sare eredu samur bat aurkezten dute hiru dimentsioko espazioko orientazioaren araberrako oszilazio-errorea ematen duena, sarrera bezala neurtutako distantzia eta kalkulaturako orientazioa, goratzea eta azimuta daukana. Hau neurketak UWB (*Ultra Wide Band*) modulu baten bidez egiten dutelako proposatzen dute, modulu horiek bi norabideko komunikazio-protokolo bat erabiltzen bait dute, orientazioaren mendeko errore bat eragiten duena. Payal et al. [91]-ek aldiz WSN bat erabiltzen dute eta hauen lokalizazio-esparru eraginkor bat garatzeko *Feedforward* sarea aplikatzen dute ere. Proposatutako lanak ainguratze-nodoek jasotako seinalearen intentsitatea erabiltzen du, ibilbide anitzeko hedapen-efektu batzuen bidez. Ondorioz, FNN-aren sarrerak hiru dimentsiotan diseinatu dira. Geruza ezkutukoak tansigmoide transferentzia funtzio bat erabiltzen du, irteerak transferentzia lineala erabiltzen duen bitartean, prezisio eta zehaztasun handiagoa emanez posizioan.

Pose inertzialaren ikus-estimazio estereoaren metodoa aurkezten dute Chen et al. [92]-ek, non posearen estimazioa kontrol eredu batekin egiten den. Sistemaren kontrol egonkor hori lortzeko *Feedforward* begiztatan sartzen dira, behearazko gradienteko berrelikadura begizta, kulunka eta zabuka berrelikadura begizta eta alborapen zenbatespen berrelikadura begizta duena. Sistema *Feedforward-feedback Loop-based Visual Inertial System* izendatzen dute eta prezisio eta sendotasuna lortzen dute SLAM-eko beste ikuspegi batzuekin konparatuz. Lee et al. [93]-ek, irudien bidezko prozesamendua egiten dute SLAM-erako irudi fusionatze eta berreraikitzearen ezagutza kuantitatiboa lortzeko prozesua, erregresio-arazo bat dela justifikatzen dute. Kasu honetan beraz, erregresio arazoari soluzioa emateko *Feedforward* sare sakona aplikatzen dute, honek duen funtzio ez-lineal zailak esleitzeko gaitasunarengatik.

Sistema dinamiko malguetan ibilbidearen kalkulua egiteko aurrez zehaztutako ikuspuntu optimizatu bat erabili daiteke. *Feedforward* sarearen komando profilak zehazten dira ibilgailua atsedenean dagoen maniobretarako eta angelu handietan bibrazioak minimoan mantentzeko (ikus Kologain et al. [94]). Gainerako sarearen agente-profilak aldiz, kontrol-arazo bat denbora diskretuan konponduz sortzen dira, Programazio dinamiko diskretuaren (DDP-*Discrete Dynamic Programming*) bidez. DDP ikuspegiak zenbaki-optimizazioko aukera bat dela demostratzen dute, aurrerapen-ibilbide optimoko profilak sortzeko.

Feedforward sareekin gehien ikertu den esparrua nabigazioarena da. Bui et al. [95]-ek definitutako ibilbideentzako jarraipen-kontrolatzaile bat aurkezten dute. Lyapunov teknika ez-linealaren bidez sendotasuna ematen diote diseinuari sentsoreen soinu eta perturbazioaren aurrean. Laser sentsore bat erabiltzen dute robotaren posizioa estimatzeko, *Feedforward*-aren sarrera izango dena, AV-aren kontrol komandoak irteerak diren bitartean. Oztupoak laser baten bidez detektatu

beharrean PSO (*Particle Swarm Optimization*) algoritmoa erabili daiteke *Feedforward* aurreratze sare batekin sinkronizatuta (ikusi Pandey et al. [96]). Horrela, PSO algoritmoak FNN balio errealen eta aurre-esandakoen arteko batez besteko akats koadratikoa minimizatzen du. Sareak sarrera bezala sentsore ultrasonikoek emandako oztopoaren distantzia jasotzen du eta robotaren diferentzialaren direkzio angelua kontrolatzen du irteeran.

Barakat et al. [97]-ek *Feedforward* kontrol eskema bat aurkezten dute korrante zuzeneko motor baten korrante beharra aurreikusteko, motorraren dinamikan eta funtzionamendu-ingurunean gertatzen diren aldaketen arabera. Horrela, motorrak kontsumitzen duen korrantea mugatzen dute tentsioa doitzea mugatu beharrean. Erabiltzen duten anplifikadoreak, nahi den abiadurara iristeko behar den tentsioa ematen du. Motorrak kontsumitzen duen korrantea, berriz, alderantzizko eredu dinamikoak baimentzen duen baliora mugatzen da, abiaduraren kontrol ona lortuz. Norabidearen kontroladore bat garatu dute Kapania et al. [98]-ek, berrelikadura duen *Feedforward* batean oinarrituta, ibilgailuaren egonkortasuna berma dezan maniobragarritasun limiteetan. Gainera, alboko desbiderapena ere murriztu dute ibilbidearen jarraipenean. Diseinuak atzeraelikadura aurreratuko eskema bat duen erreferentziako kontrolatzaile baten errendimendua eta ibilgailuaren erabilera-diagrama ez-lineal batean oinarritutako aurrera-egite algoritmo bat kontsideratzen ditu. Honek egonkortasuna emateaz gain ibilbide desbiderapenak sortzen ditu, baina, nahi den ibilbidearen alboko jarraipen portaera *Feedforward*-eko begiztan sartzeak norabide kontrolatzaile sendo bat sortzen du, ibilbidea zehaztasunez jarraitzeko gai dena.

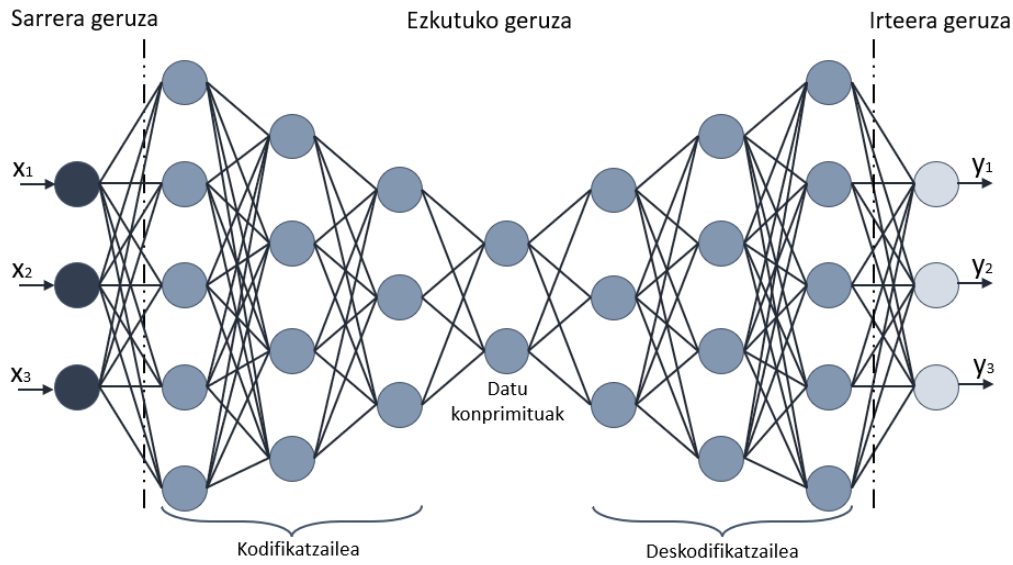
Oztopoen saihestea hiru eginkizunetan banatu daitezke: pertzepzioa, ibilbide plangintza eta norabide kontrola. Hiru horiek arteko erlazioa kontsideratzen duen arkitektura global bat proposatzen dute Laghmara et al. [99]-ek, non proposatutako kontroladorea *Feedforward* akziotan eta egoera erreazio sendoetan oinarrituta dagoen, alboko-errorean perturbazioen inpaktua murrizteko eta alboko egonkortasuna bermatzeko. Alboko-kontrolak luzetarako abiadura eta sarrerako posizio komandoaren influentzia handia duela diote Suráez et al. [100]-ek. Hori kontuan izanda, eskema moldatzaile bat proposatzen dute, non erreferentzia eredu bat *Feedforward* doikuntza arau batekin konbinatzen den aurrerapen irabazia doitzeko. Horrela bi parametro sintonizatzen dituzte sendotasuna lortzeko abiadura eta erreferentzia seinalearen bariazioei aurre egiteko: egokitze irabazia eta doikuntza arauaren ordena deribatiboa.

Gertatu daiteke ibilgailu autonomoaren nabigazioa lurzoru inklinatu batean egin behar izatea. Kasu horietan dauden erreferentzia ibilbideak jarraitzeko proposatzen dute Bian et al. [101]-ek alboko kontrola egitea. Lurzoru inklinatuek ibilgailuaren mugimenduan duten inpaktua aztertzen da, grabitatearen alboko perturbazioa modelatzeko eta *Feedforward* konpentsagailu bat diseinatzeko efektua arintzeko. Modu honetan alboko kontrolatzailea aurrerapen-kompentsagailu batez eta berrelikadura-kontrolagailu batez osatua dago, denboran aldagarria duen eredu linealaren kontrol prediktiboan oinarritua. Aireak ere perturbazioak eragiten ditu eta hauen efektua, grabitatearekin batera, perturbazio-efektuak dituen eredu ez-lineal bat sortu daiteke. PID kontroladore bat *Feedforward* kontrolarekin erabili daiteke perturbazio horiek ezabatu ahal izateko (ikusi Sailan et al. [102]).

Beste alde batetik, neurketa inertzialeko unitateak (IMU-*Inertial Measurement Unit*) oso erabiliak dira nabigazio inertzialeko sistemen nukleo gisa, baina erroreak existitzen dira azelerometro eta giroskopioaren seinaleetan. Gailuak kalibratzea errore deterministak ezabatu ahal ditu baina ez estokastikoak. Horretarako Luu et al. [103]-ek *Feedforward* Kalman filtroa erabiltzea iradokitzen dute, INS-aren (*Inertial Navigation System*) zaratadun irteera GPS-aren (*Global Positioning System*) irteera zaratatsurekin konbinatuz, nabigazio sistemen kalitatea hobetzeko.

Autoencoder arkitektura

Autoencoder sare neuronalak kodifikatzaile batez eta deskodifikatzaile batez osatutako arkitektura da oinarrian, sarrera datuen errepresentazio konprimatu bat topatzea helburutzat duena. Prozesua kodigo edo errepresentazio bat aurkitzean datza, sarrera datuen transformazio erabilgarriak egiteko [30]. Kodifikatzaileak, x sarrera datuak $h=f(x)$ -etan bihurtzen ditu, hau da, dimentsio txikiko irudikapen latentetan. Horrela sarrera datuen ezaugarri garrantzitsuenak ateratzen dira, dimentsio baxuagoko bektore batean. Deskodifikatzaileak aldiz sarrera datuak berreskuratzen ditu irudikapen latentetik $y=g(h)$, $y=g(f(x))$ izateko helburuarekin x eta y berdinak izan daitezzen. Oro har, kodifikatzailea eta deskodifikatzailea funtzio ez linealak dira. **Irudia 10**-an ikus daiteke arkitektura honen egitura.



Irudia 10: Autoencoder sare neuronalaren arkitektura

Autoencoder erako sare neuronal batekin lokalizazioaren eginkizuna betetzeko orduan sentzore mota ugari erabili daitezke. Sentzore anitzen eta *Autoencoder* sarearen fusioa ikertzen dute Gao et al. [104]-ek, sentzoreen seinalea oztopoekin eragozten bait da lokalizazioaren zehaztasuna galduz. Denbora errealeko satelite bidezko nabigazio-sistema global zinematikoko sentzorearen datuak, IMU datuak eta kodetzaile birakari dual baten datuak, Kalman iragazki hedatu batekin fusionatzen dira eta azken honen zarataren matrizea optimizatzeko aplikatzen dute *Autoencoder*-a. Emaitzek ohiko sentzoreen prezisioa hobetzen dela adierazten dute, AMR-en errendimendua hobetzeko aukera emanez. Seinale eragozpenen arazo berdinerako Schlichting et al. [105]-ek, satelite bidezko nabigazio-sistema globaleko (GNSS-*Global Navigation Satellite System*) sentzorearen arazoak aztertzen dituzte eta 2D laser eskaner batez ordezkatzea proposatzen dute. Horrela, lokalizazioa sentzoreak detektatutako eskaneatze-lerro bertikalak inguruko erreferentzia-puntuen hodei batetik eratorritako eskaneatze-lerroekin alderatuz egiten da, sarea lerroen formen ezaugarri esanguratsuak lortzeko erabiliz.

Wi-Fi (*Wireless Fidelity*) seinalearen erabilgarritasuna ere frogatu da posizionamenduaren informazioa lortzeko barrualdeko inguruneetan. Chidlovskii et al. [106]-ek *Autoencoder* sare sakon bariazional batean oinarritzen dira, non sarea egiturazko proiektzio osagai gehigarri batekin osatzen duten, oztopo ezberdinak dituen ingurune konplexu batean kokapenaren zehaztasuna hobetzeko gai dena. Gainera, sarearen hainbat konposizio kontsideratzen dituzte, klasifikazio eta erregresio azpi-eginkizunak konbinatuz errendimendu optimoa lortzeko. Ingurune konplexutan Wi-Fi seinalearen prezisioa okerrera jotzen duela detektatu dute Liu et al. [107]-ek, RSS aldakor bat eraginez. *Autoencoder*-a erabiliz RSS zaratazua aurre-prozesatu eta lokalizazioaren zehaztasuna hobetzen dela

baieztatzen dute, entrenamendu *offline* eta lokalizazio fase *online* baten bidez. Zarata neurtzea helburua bada, algoritmo egokitzaile bat erabili daiteke, transformatzaile bat eta hondar-zarata ezabatzen duen auto-kodifikatzaile batez osatuta dagoena (ikusi Xu et al. [108]), zarataren neurriaren kobariantza dinamikoki balioztatu dezakeena. Gainera, hondar-modulua gradientearen degradazio-arazoa ebazteko erabiltzen da, *Autoencoder*-a zarata-dun datuen funtsezko ezaugarriak ikasteko erabiltzen den bitartean. Emaitzek erakusten dute nola posizionamenduaren zehaztasun handia lor daitekeen.

Sateliteko irudiak erabiltzea proposatzen dute Bianchi et al. [109]-ek lokalizatorako, metodoa frogatuz. Orokorrean, satellite irudiak erabiltzeak konputazio eta memoria kostu handia suposatzen du eta ezin dira denbora errealean exekutatu. Aurkezten duten lanean aldiz, *Autoencoder* bat entrenatzen dute irudi horiek dimentsio txikiko irudikapen bektorial batean konprimatzeko, gako diren ezaugarriak mantenduz. Ibilgailu autonomoak ateratako irudiak konprimatutako irudiekin konparatzen dira pisuen distribuzio bat emanaz eta posea lortuz. Kameraren erabilerarekin ere, lokalizazioaz gain, *inguruko mapaketa* egin daitekeela ikusi da aurrez aipaturiko ikerketetan baina askotan sarrera hau hondatuta dago zarataren ondorioz (ikusi Papachristodoulou et al. [110]), adibidez. Arazo honi aurre egiteko denbora-eremu kodifikatzaile arin bat erabili daiteke, irudien segmentazio prozesua sendotzeko.

Helburua aldibereko *mapatze eta lokalizazioa* egitea baldin bada, oso garrantzitsua da begiztaren ixte zuzena detektatzea. Sistema hauetan, begizta itxiera hori robot mugikorra aurrez bisitatutako kokaleku batera itzultzea bezala ulertu behar da. Begizta hori ez antzematea ez da arazo bat baina itxiera desegoki batek arazoak dakar, AV-aren galera eragin ahal izanez. Leonardi et al. [111]-ek algoritmo sendo eta paralelizagarri bat aurkezten dute begizten itxiera globala detektatzeko. SLAM monokularra KITTI datu-basearen bidez egiten da eta *Autoencoder*-etik pasatako irudi hauek begiztak detektatzeko algoritmoa gidatzen dute.

Wang et al. [112]-en kasuan grafoak erregularizatzeko *Autoencoder* bat proposatzen dute SLAM-ean detekzio zehaztasun handia lortuz eta fidagarritasuna hobetuz. Horrela diseinu enpirikoko algoritmoen menpe egotea saihesten dute, hala nola, kalitate baxuko ezaugarriak lortzen dituzten algoritmoak. Aurkezten dituzten esperimenduetan, proposatutako algoritmoak errendimendu handiagoa lortzen duela erakusten dute, irtenbide bideragarri bat eskainiz begizta itxia detektatzen duen SLAM-aren parterako. Aurreko hauek bezala, Luo et al. [113]-ek ohiko SLAM begizten ixte detekzioa egiteko metodoak BoW-etan (Bag of Words) oinarrituta daudela antzeman dute ere, baina hauekin arazoa ingurunearen itxuraren aldaketa txikiek eragin asko dutela da, sendotasun eta prezisio txikia sortuz. Hauen proposamena *Autoencoder* pilatu bat erabiltzea da, auto-kodifikatzaile berdinen hainbat geruzaz osatuta dagoena. Sare neuronala *denoising* auto-kodifikatzailearen, auto-kodifikatzaile konboluzionalaren eta *sparse* auto-kodifikatzailearen hainbat geruzaz osatuta dago, irudiaren ezaugarrien sendotasuna hobetuz, informazio espaziala bermatuz eta ezaugarrien dimentsionalitatea murriztuz, hurrenez hurren. Gainera begizta ixtearen zehaztasuna eta sendotasuna ere era eraginkorrean hobetzen da metodo honekin.

Behin ibilgailuaren ingurune eta posizioa ezaguna dela, aurrez aztertutako metodoekin bezala, *ibilbidea aurreikuszeko aukera dago* *Autoencoder*-a erabiliz ere. Jagadish et al. [114], [115]-ek aurkeztutako bi lanetan inguruko ibilgailuen portaera ulertzearen garrantzia azpimarratzen dute, hauen ibilbidea aurreikusiz eta ibilgailu autonomoaren mugimenduak erabaki ahal izanez. *Autoencoder* bariazionalak erabiliz, inguruko AV-en ibilbide multimodalak eskuratzen dituzte, sarea iraganeko ibilbideekin eta zirkulazio egoerekin baldintzatuz.

Feng et al. [116]-ek ere, inguruko ibilgailuen mugimendua aurreikusteak, robot mugikorrarenaren aurretik, gidaketan segurtasuna eta efizientzia hobetzen duela aipatzen dute. Beraiek auto-kodifikatzaile bariatzional kondizional bat aplikatzen dute ibilbide multimodal aukerak iragartzeko, interpretagarritasun handiarekin. Horrela, hobekuntza detektatzen dute ibilbideen alboko eta luzetarako mugimenduan. Gainera, sekuentziatik sekuentziarako maniobretan oinarritutako eredu bat modifikatzen dute ibilbidea iragartzeko eta konparaketak egiteko. Huang et al. [117]-ek bi ereduak alderatu beharrean, hau da, sekuentziatik sekuentziakoa eta *Autoencoder* bariatzionala, bateratu egiten dute auto-kodifikatzaile sekuentzial bariatzional bat eraikiz, inguruko trafikoaren arazoari irtenbidea emateko. Horrela, proposatutako ereduak inguruko mugikortasunaren ezaugarri garrantzitsuenak antzeman ditzake, arrazoizko ibilbideak proposatuz.

Nabigazioaren kasuan Walker et al. [118]-en ikerkuntzan, *Autoencoder* sakonaren erabilera aztertzen da irudi baten errepresentazio konprimatua aurkitzeko zein ikusmen nabigazioan erabili daiteken. Errepresentazio konprimatutik irudia berreraiki eta frogatzen da ea nahiko informazio duten ikusteko. Horrela, sare sakonak etorkizun handikoak direla frogatzen dute ikusizko kodifikazio optimoak aurkitzeko, bai berreraikitako irudiak bai auto-kodifikatzailearen geruza ezberdinetako errepresentazio konprimatuak nabigazioan erabili daitezkeela baieztatuz.

Gu et al. [119]-ek nabigazioa egiten den bitartean erre aldaketaren arazoa aurkezten dute. Horri aurre egiteko, erabaki eredu bat sortzen dute ibilgailuari deliberamendu gaitasuna emanaz. *Autoencoder* sare sakon bat eta *XGBoost* algoritmoa konbinatzen dute, auto-kodifikatzailea aldagai anitzeko berreraikitze-eredu sakon bat eraikitze erabiliz, non sentsoare ugariko denbora-serieen datuekin entrenatutako sarearen errorea aztertzen den. *XGBoost*-a ordea, erre aldaketen erabakiak hartzeko prozesurako aplikatzen da, arazo hau parametro anitzekoa eta ez-lineala izanez.

Amini et al. [120]-en hitzetan, entrenatutako ereduak alde aurretik ohartarazi gabe huts egin dezakete, entrenamendu datu eskasak edo partzialak dituzten egoeretan. Arkitektura berri bat da hauen konponbidea, aldagai ezkutuekin entrenatua, nahiko entrenamendua ez duten egoerak detektatzeko, entrenamendu datuen desoreka lantzen duten bitartean. *Autoencoder* bariatzional batez baliatuz, kodifikatze parte sentsoaren datu gordinen eta ingurunearen dimentsio txikiko kodifikatzailearen arteko korrespondentzia ikasteaz arduratzen da. Deskodifikatzailea aldez, alderantzizko mapaketa ikasten du, ingurunearen lagin bat hartuz jatorrizko sarrera berreraikitze gaitasuna izanez.

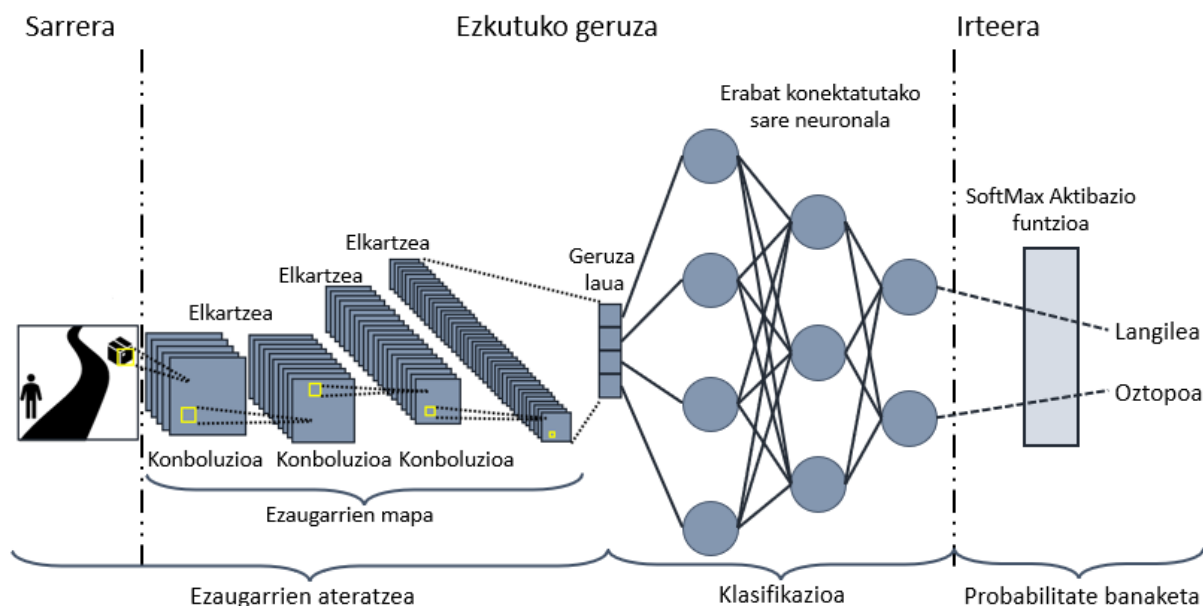
Datu eskasen arazoa ere Gebauer et al. [121]-ek detektatu dute nabigazioa aurrez entrenatutako auto-kodifikatzaile batekin egiterako orduan. Horretarako informazioan aberatsa eta trinkoa den ingurunearen irudikapen baten sorkuntzan egiten dute azterketa. *Autoencoder* bariatzional bat entrenatuz, LiDAR-bidimentsionalaren eskaneatzeak egiten dituzte, ingurunea nabigazioaren errefortzuzko ikaskuntzarako erabiltzeko. Okupazio bitar lokala duen irudi baten aurre-prozesamenduaz, 2D-lidar eskaneatzearen auto-kodifikazioaren testuinguruan, edozein ingurunetan nabigazioa gauzatzeko gai dira. Richter et al. [122]-en kasuan, iragarpen erratikoak edo ez-segurua sortzen diren ingurune horietarako, *Autoencoder*-a egoera noiz den berria antzemateko erabiltzen dute, aurretiazko portaera seguru batera itzultzeko. Gaitasun horrekin ere, ikaskuntza sakon autonomoko sistema zabaltzen da ingurune arbitrarioetan, entrenamendu egokiaz arduratu gabe.

Argi dagoena da nabigazioa gauzatzeko datu asko bildu, gorde eta komunikatu behar direla, eta honetan arreta jartzen dute Perera et al. [123]-ek, honek nabigazioaren errendimendua baldintzatzen bait du. Hauen proposamena kodetze automatikoko arkitektura duen sistema bat erabiltzea da, parametroak konprimatu ahal izateko. Aldagai nagusien analisi bat egiten da, datu-basearen egitura osatuko dutenak eta hauen konprimatzea egiten da *Autoencoder* lineal baten bidez.

Convolutional Neural Network arkitektura

Sare neuronal konboluzionalak (CNN-Convolutional Neural Network) adimen artifizialean erabilera anitza duten sare arkitekturak dira, batez ere informazio espaziala prozesatzeko. CNN-ak ikaskuntza sakoneko erreminta oso ahaltsuak dira, automatikoki, sarrera irudi batetik abiatuta ikusizko ezaugarri diskriminatzaileak antzemateko duten ahalmenagatik. Izan ere, gaitasun horrek patroien miaketan, klasifikazioan, objektuen detekzioan edota beste ikusmen artifizialeko aplikaziotan baliagarria bilakatze du [124]. Sare mota hauek konboluziotan oinarrituta daude, matematikako eragiketa lineal mota espezializatuak direnak. Ohiko sare neuronalek matrizeen biderketa parametro independente batekin erabiltzen duten bitartean, hau da, sarreraren eta irteeraren arteko elkarreragina deskribatzen duen parametroa, CNN-ak iterazio sakabanatuak dituzte, hala nola, sarrera datuetatik lortutako ezaugarri espezifiko eta esanguratsuak erabiltzen dituzte.

Sare neuronal konboluzionalak orokorrean pisu-bektore batengatik parametrizatuta daude, W neuronaren arteko pisu talde batez eta b alborapen-balioen multzo batez osatuta, $\theta=[W, b]$ bezala definituz. Gainera, geruza bakoitzeko sarrera hiru dimentsio izan behar ditu: zabalera, altuera eta sakontasuna. Entrenamenduaren prozesuaren hasieran, konboluzio geruzetan, maila altu eta balioa duten ezaugarriak ateratzen dira, ondoren, elkartze geruza erabiliz, lortutako ezaugarri-maparen tamaina murrizteko kostu konputazionala murriztuz ere. Laburbilduz, geruza bakoitzak sarrera eraldatzen du ezarritako parametroei dagokien irteera bat lortuz. Hauen emaitzari aktibazio funtzio ez lineal bat aplikatzen zaio eta konexio osoko geruzara pasatzen da, zeinetan neuronek dagokien pisu eta alborapenekin batera, beraien arteak konektatzen dira. Azkenik sareak iragarpen bat egiten du **Irudia 11**-an ikus daitekeen bezala.



Irudia 11: Convolutional Neural Network sare neuronalaren arkitektura

Sare mota hauen arkitekturaren ezaugarriak direla eta, ohiko erabilera lokalizazio eta mapatze eragiketak izan dira, irudiak prozesatzeko duten ahalmenagatik. Gainera CNN-ak ingurune baten deskriptore holistikoak eman ditzakete, ingurune dinamikotetan sendoak direnak.

Barruko ingurune batean robot mugikorraren lokalizazio hierarkiko bat egiteko CNN-a erabiltzea proposatzen dute Ballesta et al. [125]-ek, norabide orotako irudiak erabiliz sarrera gisa. Transferentzia bidezko ikaskuntzaz baliatuz, CNN-a eraldatzen dute erregresioa egin dezan, horrela, gela konkretu baten barruan posizio koordinatuak estimatzeko gai izanez. Norabide orotako kameran

erabilerak sare neuronalaren aldaketa ere ekar dezake (ikusi Wang et al. [126]), kameren jarrera aldakuntzei aurre egiteko. Gelaren identifikazioa egin ordez, eredu honek leku hurbileneko ale bat berreskuratzen du eta leku horren eta sarrera irudiaren arteko distantzia erlatiboa kalkulatu du. Horrela politika heuristikoa bat proposatu ahal da robota leku hurbil horretara gidatzeko, zehaztasun handia lortuz ingurune birtualean. Xu et al. [127] aldiz, sentsore anitzetan oinarritzen dira. Proposatzen duten sistema CNN-an oinarritutako irudi berreskuratzeko lagunduriko lokalizazio bisuala integratu du, kokapen probabilistikoaren ikuspegiarekin. Kamera monokular bat erabiltzen dute hasierako lokalizazioa gauzatzeko, non aurrez entrenatutako sare neuronal konboluzionalatik ateratako ezaugarri estandarrik hartzen diren hautagai diren kokalekuak identifikatzeko. Ondoren, laser telemetro bat erabiliz pose zehatza onesten da, Monte Carlo lokalizazio moldagarriarekin aurrez lortutako kokaleku hautagaien bidez.

Ibrahim et al. [128]-ek diotenaren arabera, nodo mugikorren barrualdeko lokalizazioak interes handia jasotzen hasi da, gailu mugikorretan izan berri diren aurrerapengatik eta kokapenean oinarritutako zerbitzuen gero eta kopuru handiagotatik. Aurkezten duten lanean, sare neuronal konboluzional bat erabiltzen da hari gabeko sare lokaleko (WLAN-*Wireless Local Area Network*) sarbide-puntuen RSS denbora-serieak erabiliz barnealdeetan lokalizatzeko. Ikerketa ildo beretik, Njima et al. [129]-en helburua sentsore bat zehaztasunez aurkitzea da eta horren kokapena zehaztea. Horretarako, kokapen esparru bat garatzen dute, sare neuronal konboluzionala oinarrituta, non barnealdeko kokapen-arazoa RSS azterna digitaletatik eraikitako 3D irudietan oinarritutako eskualdeen miaketa gisa formulatu den.

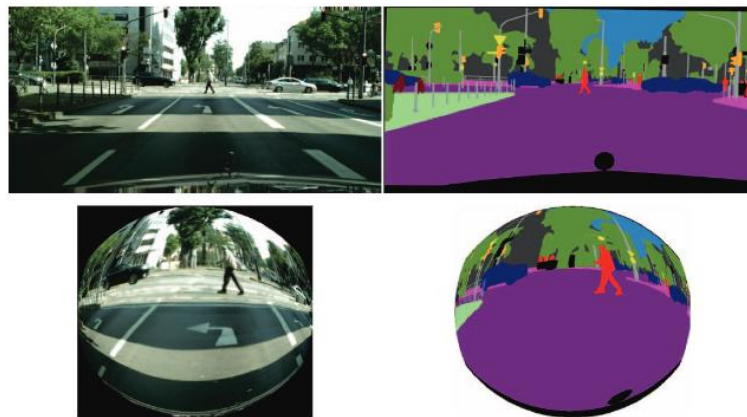
LiDAR sentsorearen aplikazioa ere erabilgarria da SPAD (*Single-Photon Avalanche Diode*) diodoarekin batera baliatzen denean. LiDAR eta SPAD-aren funtzionamenduaren teoriarik oinarrituz, Ito et al [130]-ek diote fusio honek, aldi berean, irismen-irudiaren datuak eta irudi monokularren datuak igortzen dituela koordinatu-sistema berarekin, eta ez duela irteeren arteko kanpo-kalibratze eskatzen. Beraz, CNN sakon batekin batera lokalizazio metodo bat aurkezten dute, guztien irteerak batuz, hein-irudiaren datuak, irudi monokularren datuak eta intentsitate goreneko irudiaren datuak lortuz. LiDAR eta SPAD-aren ikerketarekin jarraituz Ito et al. [131]-ek, CNN-a erabiliz, lokalizazioa egiteko datuen kalitatean atentzia jartzen dute, hauen mende dagoelako emaitza ona lortzea. Kalitate indize bat finkatzen dute datuentzako, fotograma jarraien arteko korrelazioan oinarrituta, pixel garrantzitsuak identifikatu. Indize horren ebaluazioa LiDAR SPAD-aren bidez egiten da datuen kalitatea sailkatuz.

Mapatzearen zati garrantzitsu bat ere, oztopoen klasifikazioa da eta barrualdeko kasuetarako Afif et al. [132]-ek sailkapen sistema berri bat proposatu dute CNN sakonean oinarrituz, ibilgailu autonomoetarako. Protasov et al. [133]-ek aldiz, sarea optimizatzeko algoritmo bat proposatu dute oztopoen klasifikazio binarioa egiteko baina 6 kameren erabilera inplikatu du, 360º-tako ikusmenarekin. LiDAR sentsorea erabili nahi baldin bada, sare neuronal konboluzionala eta irudien laginketaren goi-mailako teoria aplikatu behar da (ikusi Gao et al. [134]). LiDAR datuen laginen puntu-hodeia sortzean eta hauek pixel mailako sakontasun informazioan bihurtzean sarea hornitu daiteke oztopoak aurkitzeko.

Nabigazio-ingurunearen mapatzea soilik egiteko, segmentazio semantikoa da baliabiderik hedatuena. Irudiekin egin daitekeen prozesamenduaren ondorioz beraz, sare neuronal konboluzionalak oso aproposak dira eginkizun honetarako. Liu et al. [135]-ek eszenaren errekonozimendua egiteko metodo bat proposatu dute segmentazio semantikoa egiten duen CNN sakon baten bidez. Gainera, biltegi guneetan, AV-ak ikaskuntza sakona erabili dezake inguruneko ezaugarriak ikasteko eta errekonozimendu zehatz bat egiteko, erreferentzia puntuak erabili gabe. Kasu honetan ereduaren aktibazio geruzan gradiente-errendimendu onena duen aktibazio funtzioa

erabiltzen dute, ReLU (*Rectified Linear Unit*) funtzioa baino zehaztasun gehiago izatea demonstratu duelako. Unitate errepikari itxiak (*GRU-Gated Recurrent Units*) eta sare neuronal konboluzionalak batuz ere ibilbideen segmentazioa egin daiteke nabigazio eskualdea lortzeko (ikusi Lyu et al. [136]). Ibilgailu autonomoan kokatuta dagoen aurreko kameraren datuekin CNN-GRU sarea entrenatzen da, eta kasu honetan, sareak, sekuentzia espazial luze bat lortzen du, *Autoencoder* arkitektura tradizionalak baino konputazio konplexutasun txikiagoa duena. Honek, zehaztasuna lortzen du segmentazioan, denbora errealean prozesatzeko aukeraz gain.

Chiodini et al. [137]-en ikerkuntza, hiru dimentsiotako mapa semantikoak sortzeko teknikan izan da. Sarrera bezala, oraingoan, irudi estereoskopiko bat erabiltzen da, robot mugikorraren gainean instalatuta dagoen kamera batek lortua. Irudiak etiketatu eta kodifikatu eta deskodifikatzen duen CNN bat erabiltzen dute. Horrela, segmentazio semantikotik lortutako etiketak sakonera estereoko mapetan konbinatzen dira. Deng et al. [138]-ek aldiz arazoa ibilgailu autonomoek duten ikuspegi-eremuan finkatzen dute, beraz, kamera kasu honetan arrain-begi objektiboduna da, ikuseremu zabala hartzen duena. Sare neuronal konboluzionalarekin OPP (*Overlapping Pyramid Pooling*) modulua erabiltzen da, informazio lokala eta globala esploratu ahal izateko. Horrela segmentatutako irudietan klaseen bereizketa eraginkorra lortzen dute, **Irudia 12**-an ikusi daiteken bezala.



Irudia 12: Arrain-begi kamerarekin lortutako mapatzea ([138]-ko iruditik moldatua)

Sare neuronal konboluzionalarekin oinarrituta, bi esparru osagarri aurkezten dituzte Yang et al. [139]-ek, sarrera eszenetako eskualde kontributiboan automatikoki zehazteko. Lehenengo esparruan ezaugarri mapetan oinarritutako metodo bat proposatzen da, dekonboluzioa erabiliz, garrario-eszenak ulertzen gehien laguntzen duten ezaugarriak ateraz. Bigarren esparruan, eskualdeen garrantzi maila sailkatzen etiketatu eta aurreikusitako kontrol sarreraren arteko errore mapa erabiliz. Errore mapa, sarrera eszenaren zati ezberdinak oklultzean sortzen da, pixelen garrantzi klasifikazio bat lortuz. CNN sarean sartzen dira proba datu multzoak, kontribuzio eskualdeekin batera eta emaitzek, lehenengo esparruan identifikatutako ezaugarriek eta bigarrean identifikatutako eskualdeek garrantzi handia dutela sareak egiten duen ezena ulerkeran erakusten dute.

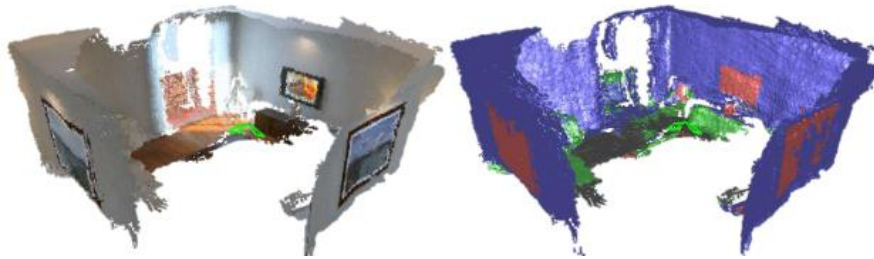
CNN sareek duten irudiak sailkatzeko eta ikusmen lokalizatorako datu-ezaugarrien mapak ateratzeko gaitasuna ustiatuz Foughi et al. [140]-ek barrualdeko ingurunetarako SLAM-a proposatzen dute. Aurkezten duten sistemak, robot mugikorraren lokalizazio-eremua zehazten du inguruko mapa topologiko eta sare neuronal konboluzionalen eremuan zentratuz. Gainera galtze-funtzio berri bat kontsideratzen dute sare mota hauek datu-multzo txikietan duen orokortzeko gaitasun eskasari aurre egiteko. Datu horiek LiDAR bat erabiliz lortu daitezke, Qin et al. [141]-ek egiten duten bezala baina azpi-mapekin konbinatzeko beharra ikusi dute sentsoaren sentzibilitateak ekiditeko. LiDAR datuak irudi bihurtzen dituzte kameraren datuekin, begiztaren itxiera sendo eta

zehatza detektatuz. Kasu honetan, CNN-ak sortutako deskribatzaileak ezaugarri gisa erabiltzen dira irudien konparaziotan. Song et al. [142]-ek LiDAR bezalako sentsore bakarra erabiltzeak zaratak sortzen dituela diote, estimazio desegokiak eraginez. Horretarako, sentsore anitzen bat-egite konfigurazio hibrido bat aurkezten dute, kameraren neurketa asinkronoekin, LiDAR-arekin, odometriarekin eta IMU datuekin. Bi atal bereizten dira sistema honetan, paraleloki funtzionatuz. Alde batetik, LiDAR-a eta kamera CNN ereduarekin konbinatzen dira klasifikazioa eta segmentazioa egiteko. Bestetik, odometria eta IMU sentsoreak Kalman iragazkiaren bitartez batuz, objektuak detektatzen dira. Bi pausu hauek lokalizazio errorea gutxiagotzen laguntzen dute.

Ikusi ahal izan denez, SLAM-aren barruan, objektuen hautematea zati garrantzitsu bat da. Gainera, Liu et al. [143]-ek diotenez, odometriak zenbatespen zehaztugabea sortzen du eta LiDAR-en mapaketan erroreak sortzen dira, objektu dinamikoak detektatzeko arazoak sortuz ekipamendu hau duten AV-etan. Hauen proposamena eredu gabeko objektu dinamikoen detekzio metodo azkar bat erabiltzea da LiDAR-ean soilik oinarrituta, sare neuronal konboluzional baten bidez, puntu-hodei bateko informazio espaziala eta denborazkoa erabiliz. Hasteko, puntu-lainotik nabigazio lurzorua lortzen da ondoren, lurzoru horretan dauden objektuek mugimendua ondoz-ondoko fotogrametik kalkulatu. Helburu berbera dute Zhang et al. [144]-ek baina kasu honetan SLAM esparru semantiko berri bat proposatzen dute, mugikorak izan daitezkeen elementuak *Mask R-CNN* baten bidez detektatuz, RGB-D kamera baten eszena dinamikoetan sendotasuna lortzeko. Modu honetan, jarraipena, mapaketa lokala eta begiztaren itxiera jarraitzen da. Kamera mota berdinez baliatuz Torres Angonese et al. [145]-ek, laser sentsore bat gehituz, bi pausotako objektu identifikazio sistema aurkezten dute SLAM-arekin bat, objektu anitz edo bakarra aurkitzeko. SLAM-arekin batera ROS nodo bat jartzen dute funtzionamenduan eta identifikazioa CNN-aren bidez egiten dute.

Steenbeek et al. [146]-en ikerketan aldiz, SLAM-a denbora errealean egiteko ibilgailuak duen gaitasuna aztertzen da. Hauek irudiak erabiltzen dituzte sarrera bezala eta mapaketa eta lokalizazioarekin batera sakontasun estimazioa egiten duten algoritmoak aplikatzen dituzte, CNN-tan oinarrituta. Modu honetan, datu loditze eta eskalatzea lortzen dute, esplorazio ingurune egokiaren mapa bat eskainiz denbora errealean. Sakontasuna erabiltzea lokalizazioa lortzeko beste modu bat da eta Lee et al. [147]-ek ideia horrekin, CNN arkitektura proposatzen dute SLAM sistemak sakontasun mapa aintzat har dezan, informazio espazial eta denborazkoa erabiliz. GRU zelula konboluzional baten bidez aurreko konboluzio geruzen pisuak gogoratzea lortzen dute eta geruza gehigarrien erabilera sartzen dute sakonera-seinale sakabanatuak maneiatuz eszenen egitura zaintzeko.

Jin et al. [148]-ek kamera monokularretan oinarritutako SLAM sistemek ezin dutela zuzenean sakontasunaren informaziorik lortu diote, eta gehienek eskalaren ziurgabetasunak dituztela. Honi erantzuna emateko, sakontasun estimazioa DenseNet eta CNN baten bidez egitea proposatzen dute. Modu honetan, *Autoencoder* arkitektura erabiltzen da transferentzia-ikaskuntzan oinarrituz eta sare neuronal konboluzionala RGB irudien sakontasuna estimatzeko. Aldi berean, sakontasunaren informazioaz, posible da kameraren pose zehatzak eta mapaketa trinko bat lortzea barrualdeko inguruneetan. Behin sakontasun mapa CNN-aren bidez kalkulatu dagoela Tateno et al. [149]-ek mapa horiek modu natural batean zuzeneko SLAM monokularren bidez lortutako sakonera-neurketekin fusionatzea proposatzen dute. Horrela, etiketa semantikoak berreskuratzen dira semantikoki ingurua berreraikitze aukeraz, **Irudia 13**-an ikus daitezkeen erara.

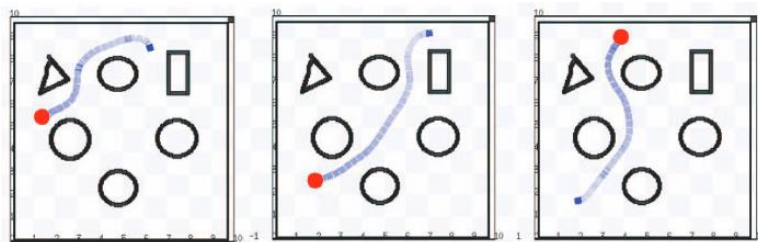


Irudia 13: Hiru dimentsiotako SLAM berreraikitzea klasifikazioarekin ([149]-ko iruditik moldatua)

Bai lokalizazioak bai mapatzeak konputazio kostu handia badute, logikoa da pentsatzea SLAM sistemen kostua handiagoa izango dela, aurreko bien fusioa bait da. Karpyshev et al. [150]-ek kamera ugari duten robot mugikorretarako, SLAM algoritmoen eraginkortasun konputazionala eta sendotasuna hobetzeko ikuspegi bat deskribatzen dute, bitarteko geruza bat ezarriz sare neuronal konboluzionalan. Geruza horretan irudiak sailkatzen dira, ResNet18-an oinarrituta, robotaren kokapenean duten aplikagarritasunaren arabera. Proposatutako ikuspegiak SLAM algoritmoak lokalizatu ditzakeen irudi kopurua nabarmenki handitzen lagundu dezake, eta orokorrean, SLAM bisualaren sendotasuna hobetzen. Konplexutasun konputazionalak ere zaildu egiten du CNN-en denbora errealean integratutako SLAM sistemen aplikazioa. Xu et al. [151]-ek, arazo horri aurre egiteko denbora errealean puntu bereizgarriak ateratzeko metodoa proposatzen dute, *Softmax* geruzaren fluxua optimizatuz eta murriztuz. SLAM-a optimizatzeko jatorrizko konboluzio geruza bakarraren ordezkari paralelismoa ezartzea eta ereduaren parametro kopurua murriztea da Wang et al. [152]-en irtenbidea. Gainera, azken ikuspegi honek sistemen gradientearen desagertzearen arazoa murrizten du.

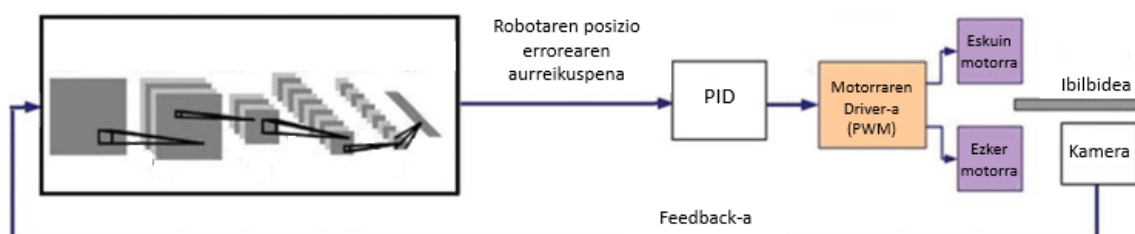
Ibilbidea kalkulatzearen arazorako Bae et al. [153]-ek robot anitzen kasurako ikaskuntza sakona eta sare neuronal konboluzionalak konbinatzen dituzte. *Deep Q-learning*-a aplikatzen da ikaskuntza algoritmoa indartzeko, CNN algoritmoarekin bat, beharrezkoa dena inguruneke egoera modu eraginkorrean aztertzeke. Sare neuronal konboluzionalak egoera zehatza aztertzen du inguruko irudiaren informazioa erabiliz eta ibilbide bat ezarri dezake Q-ikaskuntzak aztertutako inguruko informazioarekin batera. Ma et al. [154]-en kasuan errefortzuzko ikasketa konbergentzia moteleko abiadura arazoei, gehiegizko iterazioei eta konbergentzia errendimendu ezegonkorrari aurre egiteko erabiltzen da. Aurrekoek bezala, CNN sarea *Q-learning* algoritmoan sartzea proposatzen dute. Errefortzuzko ikaskuntzaren ordezkari, Theta* algoritmo tradizionalarekin lan eginda, Sartori et al. [155]-ek ere ibilbidearen estimazioa egiten dute. Sareak ingurunearen irudiak jasotzen ditu, oztipoen banaketarekin eta hasiera eta helmugako puntuen kokapenekin. Azkenik, ibilbidea osatzen dute puntuen koordinatuak lortzen dira irteera gisa, ibilbide seguru eta fidagarriak sortuz.

Zhang et al. [156]-ek maila-ikaskuntzan oinarritutako algoritmo bat proposatzen dute ibilbideen plangintzaren arazoa konpontzeko, non bi adarreko sare neuronal konboluzional ebolutibo bat aurkezten dute. Sare honek zuzenean ibilbidea zehaztu dezake ingurunearen lurraren irudietatik abiatuz, mapaketaren beharrik gabe. Mapen beharrik ez izateak konputazio kostea arintzen du eta horretan oinarritu dira Lu et al. [157] ere. Horretarako CNN-a epe luzerako memoria batekin elkartzen dute ibilbidea kalkulatzeko. Sareak sentsoreen informazioaz baliatuz inguruko ezaugarriak eta aginte-abiadura egokia kalkulatu ditu. Memoriak kasu honetan, ekintza jarraituen arteko erlazioa ikastea eta robotaren abiadura leuntzea du helburutzat. Inguruko datuak berriz ere LiDAR baten bidez eskuratzen dira eta **Irudia 14**-an ikus daitezkeen emaitzak lortzen dira, LiDAR-ak ingurunearen informazio abstraktua jasotzen bait du eta orokortze ahalmena dagoelako delako informazioarekin.



Irudia 14: Ibilbide plangintza oztopoak dituen ingurune batean abiapuntu ezberdinetarako ([157]-ko irudia)

Nabigazio aplikazioak ere ohikoak dira *Convolutional Neural Network* arkitektura erabiliz. Ibilgailu autonomoen kontrol aurreratua diseinatzeko aukera ematen dute sare hauek, robota ibilbide bat jarraitzeko gai izanez abiadura kontrolari esker. Bi aldibereko kontroladoreen erabilera ere aplika daiteke: alde batetik, sare neuronal konboluzionala, objektuen jarraipena eta ibilbidearen iragarpena egiten duena eta bestetik, PID kontrolagailu bat, norabide automatikoa eta abiadura erregulatzeko (ikusi Farkh et al. [158]). Sistema hau hurrengo **Irudia 15**-an ikusi daiteke era eskematikoan.



Irudia 15: Kontroladore aurreratua CNN-an oinarrituta ([158]-ko iruditik moldatua)

Chuxin et al. [159] ordenagailu bidezko bisioan oinarritzen dira AV-ak ikaskuntza sakona egin dezan. Sare neuronal koboluzionalak entrenamendu jarrai bat egiten du jasotako irudiez baliatuz eta robotaren norabide angelua determinatzen da, nabigazioa gauzatuz. Kim et al. [160]-en ikerkuntzan, CNN-en entrenamendu integral bat proposatzen da eta sistema honek zuzenean abiadura lineal eta angeluarra sortu dezake pauso bakar batean, sarrera iruditik abiatuz. Horrela denbora errealean funtziona dezaken nabigazio eredu bat lortzen da, kamera ez den beste sentsorerik erabili gabe. Kameraz bakarri baliatzen den beste lan bat Rausch et al. [161]-ena da, muturretik muturrerako kontrolagailu bat erabiliz, CNN-an oinarrituta. Modu honetan ez da errailak, objektuak edota ibilbideak detektatzeko algoritmo espezifikoaren beharrik, entrenamendutako sareak zuzenean pixelen datuak mapatzen dituelako norabide komandoak lortuz.

Segmentazio semantikoz baliatuz ere nabigazioa gauzatu daiteke Teso-FZ-Betoño et al. [162]-ek egiten duten erara. Nabigazio lurzorua lortzen dute ResNet18 sare batekin, transferentzia bidezko ikaskuntzaz baliatuz. Lurzorua maskara beraz, robot autonomo mugikorraren mugimenduaren kalkuluak egiteko erabiltzen da, aurreikusitako ibilbidea eta maskara horren erdialdeko lerroaren ezberdintasuna ateraz. Mugimendua gauzatzen denean, lurrarekin eraisteko arriskua dela eta odometriaren oinarritutako ibilgailu autonomoetan erroreak sortu daitezke (ikusi Quan et al. [163]). Odometriaren datuak IMU datuekin konbinatu daitezke eta dimentsio bateko sare neuronal konboluzional bat erabiliz, entrenamendu abiadura handia duena, posizionamendu zehaztasuna lortzen da erroreak minimizazioaren ondorio zuzen bezala.

Orokorrean AV-ei eginkizunak esleitzen zaizkie eta horien informazioa, espazio-tenporalarenarekin batera erabiltzea da Chen et al. [164]-en ideia, AV-ak dinamikoki programatzeko. Ikaskuntza sakonean oinarritutako esparru bat proposatzen dute, denbora errealeko erantzunen abantailak eta CNN-aren malgutasunaz baliatuz, DDPG algoritmoarekin batera, dinamikoki AV-en

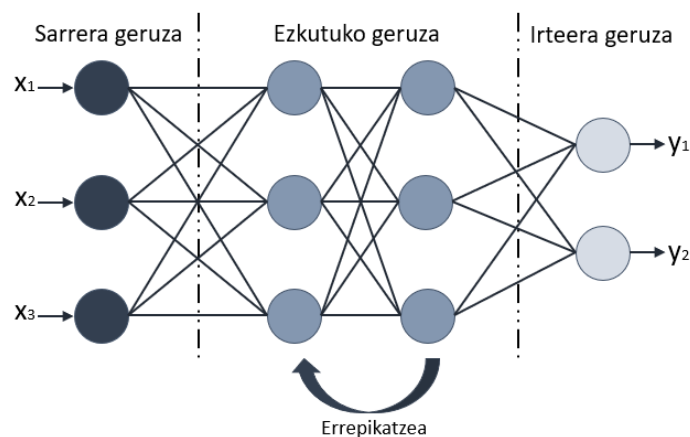
programazio estrategiak doitu daitezkeelako. Horrela, nabigaziorako erabakiak hartzen dituen robot mugikor bat diseinatu da. Errefortzuzko ikaskuntzaren ideiarekin ere, Issa et al. [165]-ek gainbegiratutako ikaskuntzarekin fusio bat ikertzen dute nabigazio autonomorako ingurune dinamiko batean. R-CNN-aren inplementazioaz inguruko oztopoak identifikatzen dira ibilgailuak arazo gabe maniobra dezan. Horrela, *Q-learning* sakon bikoitzaren politikez baliatuz agenteak erabakiak hartzen ikasten du nabigazio eraginkor baterako.

Azkenik, ingurunean oztoporik egonez gero, nabigazioa horiek saihesteaz arduratu behar da. Liu et al. [166]-ek eginkizun hau betetzeko muturretik muturrerako ikasketa eredu bat aurkezten dute CNN-an oinarrituz, kameratik zuzenean hartutako irudia erabiltzen duena sarrera bakar bezala. Metodoak zuzenean pixel gordinak direkzio agindutan bihurtzen ditu, prezisio handiaz objektuak saihestuz entrenamendu egoki batekin. Aurrera orientatuta dagoen kamera baten erabilerarekin Chakravarty et al. [167]-ek sare neuronal konboluzionala irudi bakar baten sakontasuna zenbatesteko entrenatzen dute. Sakonera mapa ondorioz, kontrol algoritmo batekin hornitzen da, jokabide-arbitrajean oinarrituta oztopoetatik urruntzeko. Chang et al. [168]-ek ordea, oztopoak ekiditen dituzten metodoek kontsumitzen duten energia eta konputazio kostea altua dela diote eta sare neuronal konboluzionalak fusionatzen dituen esparru bat aurkezten arazo hori erregulatzeko.

Recurrent Neural Network arkitektura

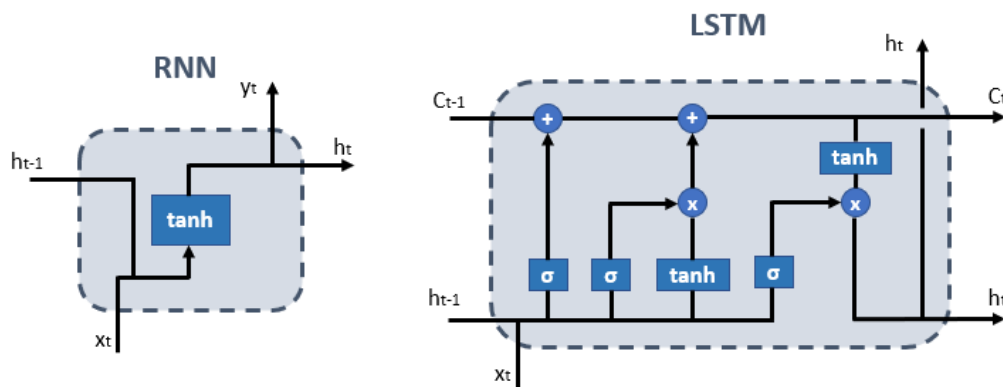
Sare neuronal errepikariak (RNN-*Recurrent neural network*) ikaskuntza sakonaren barne dagoen arkitektura mota bat da, denborazko eta sekuentziako datuak prozesatzeko ahalmena dutenak. Sare mota hauek modu desberdinean partekatzen dituzte parametroak, irteera bakoitza aurreko irteeren funtzio bat bait da, beste era batera esanda, aurreko iterazioko irteera oraingo iterazioko sarrera bat da. Beraz, RNN-ak atzeraelikadura begizta bat daukate datu sekuentzialen denbora-fasetan iteratzeko. Gainera, nodoen arteko loturek grafo bat osatzen dute denbora sekuentzia baten zehar, beraien barne-egoera erabiliz sarrera sekuentziak prozesatzeko, portaera dinamiko izanez. Portaera dinamiko honek beraz, sare neuronal errepikariei barne-egoerak mantentzeko ahalmena ematen die, denbora-pauso bakoitzaren informazioa gorde eta "memoria" kontzeptuaz aprobetxatzeko aukeraz.

Arkitektura honen kasuan, sarrera geruza t denbora-pauso bakoitzaren bektore sekuentzia bat da $\{x_t, x_{t+1}, \dots, x_{t+n}\}$ bezala izendatuta. Ezkutuko geruzaren kasuan, nodoak elkarrekin konektatzen dira lotura errepikakorren bidez, geruza hori sare osoaren memoria definituz. Hau $h_t = f_{ezkutu}(k_t)$ bezala definitzen da, non $k_t = W_{h1}x_t + W_{hN}h_{t-1} + b_h$ den, W geruzen arteko pisuak eta b alborapen-balioa eta h *hiden* (ezkutukoa) geruza izanda. Irteera geruzak $y_t = f_{irteera}(W_{irteera}h_t + b_{irteera})$ balioa hartzen du. **Irudia 16-** an ikusi daiteke sare mota hauen arkitektura era eskematikoan.



Irudia 16: Recurrent Neural Network sare neuronalaren arkitektura

Arkitektura hau duten sare tradizionalen arazo nagusietako bat, entrenatzen den bitartean, gradiente ebaneszentea da, non sarearen pisuak ez diren behar bezala eguneratzen sarearen sakontasunaren ondorioz. Honek pisu-balio oso txikiak dituzten emaitzak eman ditzake, sarearen ikasteko gaitasuna murriztuz. Horregatik oso erabilia da RNN-aren eredu handitu bat, epe laburreko memoria luze (LSTM-Long Short-Term Memory) bezala ezaguna, datu sekuentzialak modu eraginkorragoan erabiltzeko eta hobeto ikasteko gai dena. RNN sinplea eta LSTM-a konparatuz, azken honek "ateak" ditu sareko informazio-fluxua kontrolatzeko eta honek praktikoagoak bihurtzen ditu. **Irudia 17**-an ikus daiteke bi sare motak osatzen dituzten neuronen arteko ezberdintasunak. x_t balioak sarrera bektoreari egiten dio erreferentzia, y_t irteerakoari egiten dion bitartean. h_t eta C_t geruza ezkutuko bektoreak dira. Antzeman daitekeenez, RNN sarearen neurona oso sinplea da, \tanh aktibazio funtzio bakarrarekin. LSTM egituran aldiz, beste aktibazio funtzio bat agertzen da, σ bezala errepresentatuz, eta hauen arteko eragiketak sortzen dira memoria lortzeko.



Irudia 17: RNN eta LSTM neuronen konparaketa

Ibilgailu autonomoen aplikazioetan erreparatuz, sare neuronal errepikaria erabiliak izan dira bibliografian ikus daitekeen bezala. RNN sakonen erabilerarekin Bait et al. [169] lokalizazioa egin daitekeela egiaztatu dute. Hauek bi RNN erabiltzen dituzte, lehenengoak, behaketa historikoaren seinaletan oinarrituz kokapena kalkulatzeko duena eta bigarrenak, berriz, lokalizazioa iragazten duena zenbatetsitako kokapen historikoan oinarrituz, sistemaren errendimendua hobetuz. Barrualdeetako Wi-Fi seinalea aplikatuz Hoang et al. [170]-ek *Recurrent Neural Network*-arekin ibilbidean posizionatzea dute helburu eta bertan jasotako seinaleen intentsitatea hartzen dute kontuan. Gainera, seinaleen denbora gorabeheren arteko zehaztasuna hobetzeko, batez besteko iragazki haztatu bat proposatzen dute, sarearen irteerako kokapen sekuentzian ere aplikatuz.

Gomez Chavez et al. [171]-en kasuan, LSTM sare neuronal errepikaria erabili dute puntu globaletako hodeien deskribatzaileen arteko denbora erlazioak antzematen dituen posea lortzeko. Egiten duten azterketa ikusizko informazioan oinarritu ordez, denbora-sekuentzietan oinarritzen denez, esparrua sendoagoa dela diote eta denbora errealeko errendimendua itzultzen du. LSTM-aren hobekuntza bat proposatzen du Yan [172], ibilgailu autonomoen posizionamendu eta nabigazioan aplikatu daitekeena. Gainera AV-entzako posizionamendu sistema bat eraikitzen du LSTM hobetuan oinarrituta. LSTM ereduarekin ere Föster et al. [173]-ek algoritmo berri bat deskribatzen dute roboten lokalizaziorako, bai memorian, bai prozesamendu denboran eraginkorra dena. Laser sensore baten datu-fluxua robotaren kokapen kalkulatu probabilitistikoan bihurtzen dute, ingurunearen egitura ikasteko ahalmenaz. Ahalmen horrek gainera mapa inplizitu bat eraikitzeko aukera ematen du ere.

Ingurunearen mapa eraikitzeko ikerkuntza egin dutenak Liang et al. [174] izan dira. Objektu semantikoen ideiarekin LSTM sare grafikoa proposatzen dute, datu sekuentzial edo dimentsio anitzeko datuen LSTM egituraren orokortzen bat dena, egitura grafikoko datu orokorretara. Zehazki, LSTM

egitura multidimentsionaletan irudi bat modu uniformean zatitu eta pixeletan finkatu ordez, superpixel bakoitza modu arbitrarioan hartzen dute, semantikoki koherentea den nodo gisa. Horrela, irudi bakoitzerako zuzendu gabeko grafiko bat eraikitzen da, superpixelen espazio erlazioak ertz gisa erabiliz. Xiang et al. [175]-ek mapaketa egiteko RNN tradizionala aplikatzen dute, datuei elkartuta. 3D eszenen mapatze bateratua eta etiketatze semantikoa egiteko esparru berria aurkezten dute, RGB-D bideoi aplikatzen dena. Sarearen irteera kasu honetan, *Kinect Fusion* mapatze teknikekin integratzen dute 3D ingurune berreraikian etiketa semantikoak sartzeko. Mapetan sakontasun estimazioa egiteko, Wang et al. [176]-ek bista anitzeko sakonera-mapa dentsoa eta agentearen mugimendu mapa zenbatesteko metodo bat aurkezten dute, sare neuronal errepikariekin erabiltzen duten ikaskuntzan oinarrituta. Kasu honetan ere, 3D berreraikitzea egiten dute bideo batetik abiatuz, zeinetan sarrera fotogramak aldi baterako korrelazionatuta dauden.

Jada jakina denez, **aldibereko lokalizazio eta mapatzean** prezisio handia lortzea oso garrantzitsua da. Asmo horrekin, Chancán et al. [177]-ek CNN arkitektura entrenagarri bat proposatzen dute sare neuronal errepikari batekin batera. Modu honetan SLAM sakon bat egiteko gai dira, non ikusmen eta posizio errepresentazioak ikasi ahal diren irudi monokularren sekuentzia bakar batetik abiatuz. Liu et al. [178] aldiz, RGB-D irudien erabilera planteatzen dute, zeinetan IMU bat integratzen duten sare neuronal errepikari eta konboluzionalean. Horrela pose estimazio eta puntu-hodei baten erregistroa hobetzen dute, IMU-aren datuekin abiadura azkarreko datu inertzialak neurtzeko eta deriba akatsak murrizteko aukera ematen bait dute.

Segmentazio semantikoaren erabileraren ildotik, Gao et al. [179]-ek metodo berri bat aurkezen dute RGB-D bideoetan segmentazioa gauzatzeko RNN-en bidez. SLAM-a beraz, bideoko fotogramen arteko erlazioak antzemanegiz egiten da, fotograma horien arteko informazioa transferituz. Hiru dimentsiotako mapen kasurako, Cheng et al. [180]-ek mapak informazio semantikoarekin eraikitzeke sistema bat proposatzen dute. Sare neuronal errepikaria segmentazio semantikoa egiteko erabiltzen dute, SLAM algoritmo bat integratuz mapa eraikitzeke. Horrela, eskala errealeko hiru dimentsiotako mapak lortzen dituzte RGB-D datuak erabiliz sarrera gisa.

Honek, datu kantitate ugarien beharra izango duela argi dago, eta hori ekiditeko Tu et al. [181]-ek LiDAR datuak konprimatzeko metodo bat aurkezten dute, hiru dimentsioko formatuan dauden LiDAR datuetarako hain zuzen ere. Sare neuronal errepikariaren bitartez, hondar-blokeak erabiltzea proposatzen dute progresiboki informazioa konprimatzeko bi dimentsiotara. Horrela SLAM prozesuaren konputazio kostea murrizten da eta datuak deskonprimatu daitezke mapa eraikitzeke beharra egon ezker. Azzam et al. [182]-en lanean erroreak murrizteko metodo berri bat garatzen da, LSTM arkitekturaren oinarrituta, SLAM semantikoaren zehaztasuna hobetzeko erabilitako sentsore moten aurrean. Entrenamendu eta testatze datu-multzoak eraikitzen dituzte gainera, simulatutako frogan bidez, denbora errealean.

Ibilbide plangintza sare neuronal errepikariekin egiterako orduan robotak talkak saihestu beharko lituzke eta bere helbururako biderik onena eta laburrena aukeratu. Hau, lokalizazio-algoritmo bat garatuz egin daiteke marko kartesiar batean, AV-aren posizioa definituz eta helburuarekiko bere posizioa aurreikusteko gaitasuna emanez (ikus Brahmi et al. [183]). Oztopoen artean mugituko den ibilgailu autonomo baten ibilbide plangintzarako ikuspegi berri bat proposatzen dute Min et al. [184]-ek. Sentsoreen datu sekuentzialak analizatzeko RNN sarea erabiltzen dute, hauen sarrera hiru sentsore ezberdinetatik etorritik: LiDAR-a, kamera eta GPS-a. Aurreikusitako ibilbidea jarraia da eta epe-labur eta epe-luzeko bidea lortzen da algoritmo bakar batekin. S. Nair et al. [185]-ek gainera, sare neuronal errepikariak oso egokiak direla ibilbide plangintza egiteko diote. Izan ere, RNN-ak iraganeko sarrerak uneko irteerarako erabiltzen dituzenez, denbora datuentzako aplikagarri bihurtzen ditu. Beraz,

oztopoak saihestu behar diren ingurune dinamiko batean, metodo honek CNN-ak baino azkarrago ibilbidea aurrez aurre lagun dezakeelako.

Konputazio koste eta informazio behar handia duten sistemei aurre egiteko Kong et al. [186]-ek logika lehenetsi bat proposatzen dute RNN-etan oinarrituta. Proposatutako metodoa hiru zatitan dago banatuta: lehena, ibilbidearen plangintzaren teoria lehenetsia, sortutako ibilbideetan begiztak sortzeko. Bigarrena aldiz, RNN-an oinarritutako arrazoibide lehenetsia da, arauak lehenespenez gomendatzen dituena. Azken zatia, mapa ereduarekin eguneratzen den algoritmoa da, ingurune dinamikoaren informazioa kontutan hartzeko beste bi pausuetan. Inoue et al. [187]-en metodoarekin ere, ibilbide ugari eta onak sortzen dira robot mugikorraren konfigurazio-espazioan, RRT metodoaren bidez. Gainera, *Autoencoder* konboluzional bat eta LSTM sareak entrenatzen dituzte, entrenamendu-datu askoren eskuratzearen arazoa gaindituz.

Nicola et al. [188]-en lanean LSTM arkitekturaren aplikazio bat proposatzen da *online* bilaketa-agente batekin. Beraz, agentea ibilbidea erabakitzeko, infragorri laser sentsore batekin egindako mapa lokalean, eta ibilgailu autonomoaren eta helmugaren arteko informazio erlatiboan soilik oinarrituko da. Kontrajartzean, plangintza dinamiko egiteko metodo bat diseinatu dute Yuan et al. [189]-ek dimentsio bakarreko sare neuronal errepikariaren ereduaren oinarrituta, ingurune ezezagun batean kokatuko den robot mugikorrarentzako. Sarrera bezala sentsoreen informazioa izango duen sare bat erabiltzen dute eta irteera aldiz, mugimendua zuzenduko duen kontrol estrategia da. Horrela oztopoak saihestuko dituen ibilbide bat lortzen da, ohiko algoritmoak baino sendoagoa. Xie et al. [190]-ek ikaskuntza sakonerako ikuspegi bat proposatzen dute LSTM-a eta CNN-a batuz, hiru dimentsiotako inguruneetan ibilbidea planifikatzeko. Tokiko informazioa eta distantzia erlatiboa aplikatzen dute, partzialki behagarria den Markov erabaki prozesu bat bezala ibilbidea defini ahal izanez. Sarea kasu honetan, behagarritasun partzialaren arazoari aurre egiteko erabiltzen da, egoera-ekintza sekuentzia historikoaren funtsezko informazioa mantenduz.

LSTM arkitektura fuzzy kontroladorearekin erabiltzeak ere emaitza onak ekarri ditu, Guo et al. [191]-ek adierazi duten bezala, ibilbide plangintza lokaleko zereginetan. LSTM sarean oinarritzen dira kontroladorearen konplexutasuna sinplifikatzeko, ikasteko, orokortzeko gaitasuna handitzeko eta eraginkortasuna hobetzeko. Hasteko, sarearen diseinua egiten dute, ibilbidearen planifikazioa zereginetik inposatuko duten baldintzetan oinarrituta egoteko. Ondoren, entrenamendu-datuak biltzen dira fuzzy kontroladorearen algoritmoan oinarrituta, kalkulu abiadura hobetuz. Emaitza adierazgarriak lortzen direla ikusiz, errefortzuzko ikaskuntza ere aplikatzen dute aurreko sare eta kontroladorearekin batera, bakoitzaren abantailak aprobetxatzeko intentzioarekin. Aurreko kasuko funtzionamendu berbera du metodo honek baina errefortzuzko ikaskuntzak inguruneen arau berriak modu autonomoan ikasteko konbinatzen da, eszenatoki desberdinetan egokipen hobetuegia lortuz (ikus Guo et al. [192]).

Chan et al. [193]-ek RNN-an oinarritutako ikuspegi bat proposatzen dute, denbora errealean oztopoak saihesteko hiperparametro gutxi ajustatzeko behararekin. Datuetan oinarritutako sistema denez, era autonomoan sortzen dituzte entrenamendu-datuak, non bi dimentsiotako okupazio mapak erabiltzen diren. Ibilbidearen datuak eraginkortasunez sortzeko, A* algoritmoa erabiltzen dute RNN ereduaren entrenamendu autonomoaren prozesuaren hasiera gida gisa. Horren ondoren, entrenamendutako ereduak, tokiko ibilbide-plangintza egiteko gai da oztopoak saihestuz.

Eszenatoki horietan oztopo dinamikoak egon daitezkeela argi dago eta Bin et al. [194]-ek, RNN sare antolatuen eredu berri bat oso eraginkorra dela ibilbideak planifikatzeko diote, oztopoak saihestuz aldi berean. Topologikoki ordenatutako neuronek mapa ezezaguna errepresentatzen dute eta bizilagunaren kokapena eta helburuko posizioaren distantziaren datuekin, robotak bide egokia

kalkula dezake esplorazio prozesu erredundanterik gabe. Wang et al. [195]-en kasuan LSTM sarea entrenatzen dute definizio altua duten datuekin eta ibilbideak aurreikusteko erabiltzen dute. Beste era batean esanda, ibilbideak modu lokalean planifikatzeko algoritmo bat proposatzen da, inguruko ibilgailuetan aurreikusitako ibilbideei buruzko informaziotik abiatuta.

Vijayakumar et al. [196]-en lanean gidatze autonomorako jarraibide bat diseinatzen da hiru modulu dituen portaera ereduetan oinarrituta: miaketa, planifikazioa eta aurreikuspena. Modulu bakoitza, sarrerako ibilbide bideo-datuen prozesamendua erregulatzeko diseinatu dute, ikaskuntza sakoneko sailkapenak integratuz, gaitasun automatizatua hobetzeko. Aurreikuspen modulua sare neuronal errepikari batekin diseinatuta dago, etorkizuneko eroapen-ereduak aurreratzeko, egungo baldintzetan eta aurreko gidatze-emaitzetan oinarrituta. Altché et al. [197]-ek ere ibilbide trinkoa aurreratzeko lehen urrats bat aurkezten dute, epe laburreko memoria luzearen bidez. Gainera, sare horrek errepideko gainontzeko oztopoen etorkizuneko luzetarako eta albotarako ibilbideak zehaztasunez aurreikusteko ahalmena duela diote.

Portaera semantikoaren aurreraketa aztertzen dute Patel et al. [198]-ek, pausu hori bait da ibilbidearen aurreikuspenaren aitzindaria. Sare neuronal errepikarien eta eredu grafikoaren potentzia aprobetxatuz, esparru bat sortzen da ingurunean dauden oztopoak etorkizunean izango duten portaera aurreratzeko. Horretarako, LiDAR, GPS eta datu inertzial eta kartografikoak erabiltzen dituzte. RNN-aren eredu konposatu bat diseinatzen da, funtzio faktorialak ikasteko eta sare hauen eredu grafikoaren goi-mailako egitura nahiz modelatze sekuentzialerako gaitasuna aprobetxatzeko. Lv et al. [199]-en kasuan, nahiago izan dute soilik GPS seinaleen erabilera inplementatzea eta posizioa zuzentzen duen eredu bat proposatzen dute RNN hibrido batean oinarrituta, ohiko nabigazio algoritmoen ereduaren beharrik gabe. Erabiltzen duten sarea bi kapa hibrido dituen RNN bat da, irteera LSTM bikoitza duena. Honi sarrera normalizatuaren sekuentzia ematen zaio eta ondoren, erregresio geruza baten bidez, posizioaren zuzenketa lortzen da.

Nabigazio algoritmoen eraiketa ere aurrera eramane da arkitektura honetaz baliatuz, Xing et al. [200]-ek demostratzen duten bezala. Hauek, kontrol metodo sendo bat proposatzen dute, sare neuronal errepikarien egituren oinarrituta, AGV ibilgailuen ibilbideen jarraipena hobetzeko. Eredu dinamikoa eta ibilbide jarraipeneko ereduak aintzat hartuz, alboko desplazamenduak denbora mugatu batean zeroan errorea mantentzen duela bermatzen da, kontroleko bibrazioak kenduz. RNN-a beraz, sistemaren dinamika ezezagunaren sekzioa, *online* nabigazioan hurbiltzeko erabiltzen da, parametroak egokituz Lyapunov-en teorema eta Taylor-en linealizazio tekniken bidez. Modu honetan begizta itxiko sistemaren egonkortasuna bermatzen da. Bai et al. [201]-ek, LSTM-az baliatuz mugimenduen planifikazio eredu bat proposatzen dute ikaskuntza sakonean oinarrituta, espazio eta denborari buruzko informazioarekin. Lehenik eta behin, LSTM konboluzional bat aplikatzen dute irudi sekuentzialen ezkutuko ezaugarriak lortzeko, ondoren, horietatik informazio espaziala ateratzeko. Azkenik, guztiz konektatutako sare neuronala ibilgailu autonomoen angelua kontrolatzeko eredu bat eraikitzeko erabiltzen dira, emaitza zehatz eta sendoak lortuz.

Quan et al. [202]-en lanean gutxi gorabeherako ereduaren kontrol prediktiboa (*MPC-Model Predictive Control*) aztertzen da. MPC-a metodo eraginkorra dela ibilgailuen alboko kontrolerako jakina da eta erreiaren mantentze sistema ugartan aplikatua izan dela ere, baina, konputazionalki oso esijentea ere dela detektatu dute denbora errealeko kontroleretan. Arazo horri aurre egiteko, sare neuronal errepikaria erabiltzen dute MPC estandarren gutxi gorabeherako irteera emateko, *offline* entrenatutako haztapen matrize baten bidez. Horrela, errendimendu handia duen metodo bat aurkezten dute puntu jarraipena egiten duen nabigazioan.

Inertzia bidez lagundutako nabigazioa egiteko metodo bat proposatzen dute Zhang et al. [203]-ek, RNN-ak maneiatur. Orain arteko nabigazio inertzialerako metodoak odometriari zentratzen direla diote, errendimendu altuko emaitzak lortu gabe. Honek zehaztasun eta sendotasunaren galera dakar eta hori ekiditeko esparru bat diseinatzen dute, non IMU-aren integrazio termino behagarriak kalkulatu daitezke RNN-ak erabiliz. Ondoren, pose numerikoaren integrazioa eta sentsoreen fusioa egiten dute errendimenduaren irabazia lortzeko, nabigazio egoki bat eginez. IMU datuez baliatu ordez, INS/GNSS datuekin nabigazio metodo integratu bat aurkezten dute Dai et al. [204]-ek. Modu honetan, INS-aren kalkulatu printzipioaz eta RNN-aren memoria funtzioaz baliatzen dira INS-aren erroreak zenbateteko eta, horrela, nabigazio jarraitu, fidagarri eta zehaztasun altukoa lortzen dute.

Jiang et al. [205]-ek, bideotan agertzen diren objektu anitzen jarraipena egiteko algoritmo bat proposatzen dute, epe laburreko memoria luzean eta errefortzuzko ikaskuntza sakonean oinarrituta. Lehenik eta behin, objektu anitzak detektatu egiten dira YOLO-V2 (You Only Look Once) algoritmoaren bidez, objektuen detektagailu dena. Bigarrenik, objektu bakarraren jarraipenaren arazoa Markov-en erabakitze prozesutzat hartzen da, planteamendu horrek erabaki sekuentziatuak hartzen dituen agente bat moldatzeko estrategia formala ematen bait du. Objektu bakarraren arakatzailak kasu honetan, sare neuronal konboluzional bat eta LSTM bat ditu, eta, arakatzaila bakoitzak, agente bezala kontsideratuz, errefortzuzko ikasketaren bidez entrenatzen da. Modu honetan, kontrol bat osatzen da nabigazio mota hau gauzatuz. Robot industrial mugikorretarako sare lokalaren kontrol kooperatiboko arkitektura bat proposatzen da Yan et al. [206]-en ikerketan, robotaren kontroladorea bi zatitan dituen: robotak barruan integratuta duen kontrola eta sare kontrolatzailea. Horrela, oinarrituko kontrol funtzioak eta aplikazioaren mendeko funtzioak bereiztea lortzen dute nabigazio koordinatu bat egin ahal izanez garatu duten politikaren bidez, RNN-an oinarrituta.

Roboten arteko koordinazioa lortzeko, inguruko mugimendu aurreraketa egin beharra dago eta horretarako algoritmo bat aurkezten dute Jeong et al. [207]-ek LSTM baten bidez. Mugimenduen aurreikuste hau egiteko, sarea, ibilgailua beraren eta inguruko egoeren informazioaz entrenatzen da, hauek guztiek daukaten sentsoreek jasotako informazioaz hain zuzen. Modu honetan, mugimendu planifikatzaile bat diseinatzen da MPC-an oinarrituta, luzetarako azelerazio agindua zehazteko inguruko ibilgailuen egoerak aurreikusiz. MPC-tik eratorritako ibilgailuen etorkizuneko egoerak sarrera ezaugarri gisa erabiltzen dira, sistemak AV guztien elkarrekin islatzeko. Li et al. [208]-ek intentziorako inferentzia-eredu bat proposatzen dute, RNN-tan oinarrituta, denborazko serieen aurreraketa duten arazoak ekiditeko, hala nola, ez-linealtasuna. Inguruko ibilgailuen arteko elkarrekin dinamikoa kontuan hartuta, aurkezten duten ereduak, inguruko robotek egiten duten mugimendu sekuentziari buruzko informazioa hartzen du sarrera gisa. Ibilbideen pilaketa kalkulatzeko lortzen dute, AV-aren egoerarekin integratuta. Eredu ez-linealaren kontrol prediktiboa, MPC-a, ezartzen dute LSTM-arekin eta GRU-arekin batera, nabigazio seguru eta optimo bat eginez, non murrizketak dauden talkak saihesteko.

Inguruko robotekin elkarlanean funtzionatzeko ideiaz ere, Jeong et al. [209]-ek eredu interaktibo bat aurkezten dute, ibilbidearen norabide mantentzea gauzatzeko. Proposatzen duten ereduak, errailen markak ez ezik, inguruko AV-en elkarrekin ere kontsideratzen du norabidearen sarrerak zehazteko. LSTM-an oinarritzen dira eta honen neuronak jasotako garraio datuez baliatuz konfiguratu dira. Datu horiek jasotzeko, ibilgailu autonomoa LiDAR eta kamera batekin hornituta dago eta sare neuronal errepikariaren sarrerak sentsore hauek jasotako errailaren datuak, inguruko AV-ak eta ibilgailuaren egoerak dira. Horrela, sarearen irteera robotaren angelua da, norabidea mantendu dezan. Sare neuronal estruktural errepikariak baliatzen dira Liu et al. [210], gainera deszentralizatu dena beraien lanean. Sare berri honek espazio eta denbora erlazioei buruzko hausnarketak egiten ditu, robotak oztopo dinamikoen arteko nabigazioan erabakiak hartzeko. Ereduz

libre den errefortzuzko ikasketaren bitartez entrenatzen dute sarea eta gainbegiratu gabe. Horrela ingurune konplexuetan errendimendu altua duela ziurtatzen dute.

Song et al. [211]-ek inguruneko oztopo dinamikoak identifikatzeko lana egin dute ere, ibilgailu autonomoei abiadura eta norantza modu dinamikoan ajustatzeko aukera emanaz. Oztopoen norabidea behar bezala aurreikustea beharrezkoa dela diote AV-ak gaitasun horietaz hornitzeko. Horretarako, inguruko oztopoen mugimenduaren aurreikuste kontzientea egiten duen eredu bat aurkezten dute, mugimenduak eragina duten bi faktore nagusiak kontuan hartuz: distantzia erlatiboa eta norabidea. Eredu eraginkor bat sortzeko nahian, kontzienteki taldekatzen duen geruza bat gehitzen diote LSTM-ari, oztopoen arteko erlazioa eraikitzeke, oraingo posizioa eta mugimendu joera ikasiz. Bestalde, Chen et al. [212]-ek pilaketa bat egiteko eredu bat erabiltzea nahiago izan dute, LSTM eta mugimenduetan oinarrituta, oztopoen ibilbidea auresateko. Modu horretan, MPC-an oinarritutako ibilbideen planifikazio eta jarraipen sistema bat garatu dute, oztopoak saihestuz hauen ibilbidea kontuan izanda.

Oztopoak saihesteko gaitasun autonomoa eta plangintza adimena hobetzeko, Lin et al. [213]-ek konboluzioa duen sare neuronal errepikaria proposatzen dute. Metodo honetan, konboluzioak sare neuronal arrunt estandarreko konexio osoa ordezkatzeko, parametro kopurua murriztea lortuz eta ezaugarriak ateratzeko gaitasuna hobetuz. Entrenamendu datu multzoak sortzen dituzte ikaskuntza sakon prozesu honetarako eta, etorkizuneko ikuspegia duen sonar sentsoare batekin batera, ikaskuntza sistema honek oztopoak automatikoki ekidin ditzake *online* egoeran. Illdoarekin jarraituz, beste lan bat aurkezten dute zeinetan *online* plangintza metodo bat aurkezten den ibilgailuak oztopoak saihestu dezaten, erloju sare neuronal errepikarian eta epe laburreko memoria luzean oinarrituta, hurrenez hurren. Funtsean, oztopoak saihesteko plangintza, sekuentzia espazialaren plangintza arazo bat bezala planteatzen dute, sentsoaren datu espazialen sekuentzia sarrera gisa erabiliz. Irteera aldiz, kontrolatzailearentzako aginduak dira, AV-ak mugimendua gauzatu dezan. Gainera, datu-multzo bat sortzen dute sareak entrenatzeko, *offline* probatuta dagoena eginkizuna ondo betetzen duela ziurtatzeko (ikus Lin et al. [214]).

2.3. Konputazio ebolutiboaren teknikaren erabilera

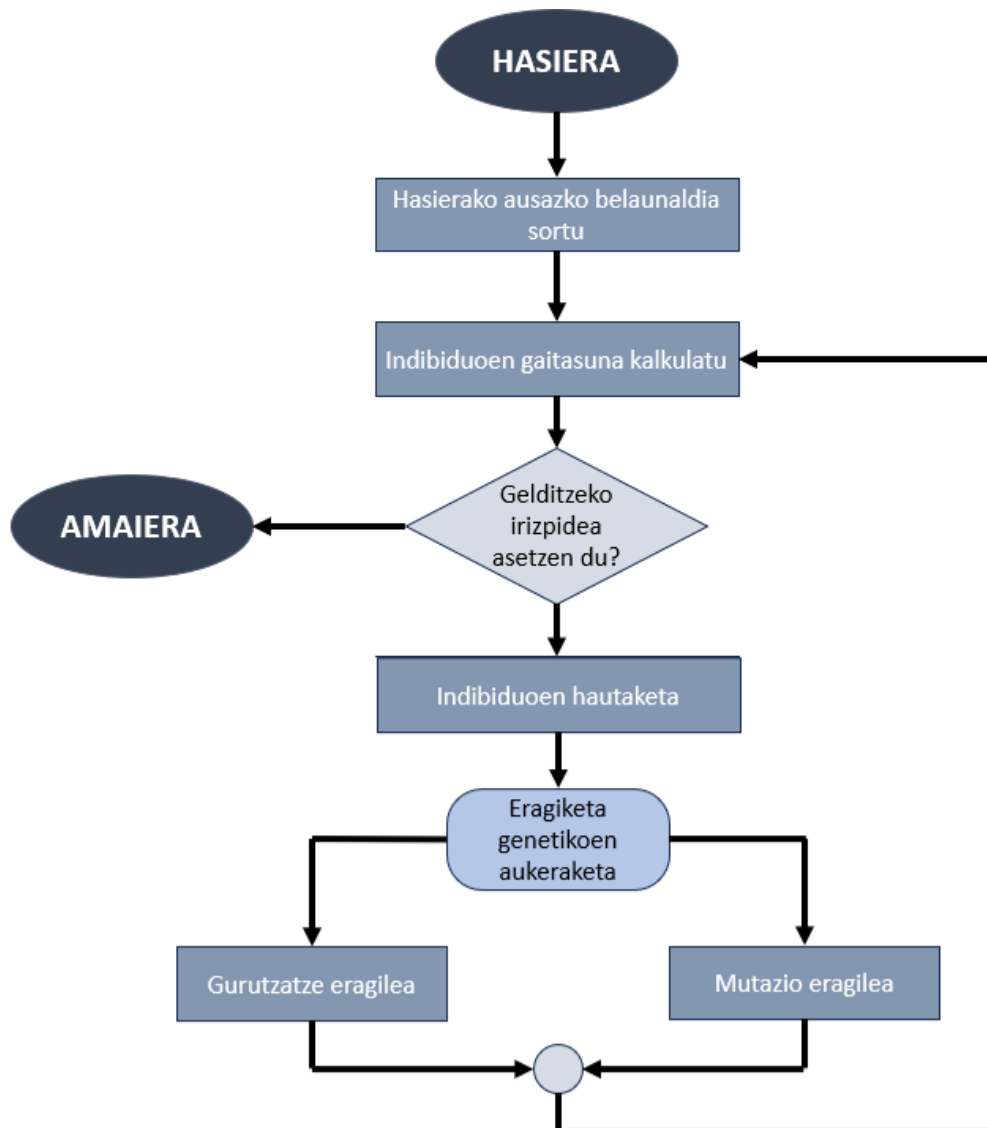
Konputazio ebolutiboa adimen artifizialaren azpi-arlo bat da, optimizazio arazo konplexu eta optimizazio jarraituko oso erabilia den teknika bat da hain zuzen ere. Orokorrean, algoritmo tradizionalentzat aldagai gehiegi dituzten problemak ebazteko aplikatzen dira. Algoritmo ebolutiboak erabiltzen dituzten eredu konputazionalek prozesu ebolutiboak erabiltzen dituzte problema konplexuak ebazteko eta, eboluzio prozesu horiek eboluzio biologikoaren teorian oinarritzen dira. Teoria horretako printzipioetako bat, adibidez, aurreko belaunaldi arrakastatsuenaren herentzia eta hautespen naturala da, irtenbide onenek beren ezaugarriak hurrengo belaunaldiei transmititzeko.

Konputazio ebolutiboaren funtzionamendua orokortuz, hasteko eta kalkulu ebolutiboa abiarazteko, irtenbide posibleen hasierako multzo bat sortzen da. Ondoren, irtenbideak behin frogatuta daudela, soluzio ahulenak estokastikoki ezabatzen dira emaitza fintzeko eta ausazko aldaketa txikiak sartzeko hurrengo belaunaldietan. Ondorioz, belaunaldiak igaro ahala, konponbideak gero eta zuzenagoak bihurtzen dira. Azkenik konputazio ebolutiboak sortutako soluzioak gaitasunaren arabera optimiza daitezke.

Konputazio ebolutiboaren barnean, nagusiki lau teknika mota aurki daitezke: estrategia ebolutiboak, programazio genetikoak, programazio ebolutiboak eta algoritmo genetikoak. Azken hau da ibilgailu autonomoen aurrerapenean eragin gehien izan duena.

2.3.1. ALGORITMO GENETIKOAK

Algoritmo genetiko batek biologia ebolutiboan inspiratutako teknikak erabiltzen ditu, hautaketa, herentzia eta birkonbinazioa adibidez, arazoak ebazteko [215]. GA-etan gehien erabiltzen den metodoa, indibiduo talde bat ausaz sortzea da, populazio jakin batetik abiatuta. Modu horretan eta programatzailerak definitutako ebaluazio funtzio baten bidez, sortutako indibiduoak baloratu egiten dira. Ondoren, indibiduoek puntuazio bat esleitzen zaie, emandako egoerarekiko duten egokitasunaren arabera. Horrela, bi indibiduo onenak aukeratu daitezke eta ondorengotza bat gutxienez sortzeko erabiltzen dira, ausazko mutazioak ere txertatuz. Aplikazioaren beharren arabera, prozesuak jarraitu egiten du soluzio onargarri bat sortu arte edo belaunaldi kopuru bat igaro arte (ikusi **Irudia 18**).



Irudia 18: Algoritmo genetikoek fluxu diagrama

Aurreko irudian ikus daitekeen bezala bi eragiketa mota egon daitezke [216]. Alde batetik gurutzatze eragiketa dago. Honetan, guraso batetik ondorengo bat sortu ahal da, bi gurasotik ondorengo bat edo bi sortu ahal dira edo, bi guraso baino gehiagotatik ondorengo bat edo gehiago sortu ahal dira. Gainera, ondorengoak sortzeko, hautatutako gurasoen informazioa galtzen ez duten gurutzaketak egiteko aukera dago, ezaugarri onenak mantenduz. Eragiketa hau aplikatzen ez bada, ondorengoa gurasoen kopia bat besterik ez da izango. Mutazio eragilearen kasuan, material genetiko berria txertatzeko modu bat bezala ikus daiteke, soluzioen genetikan ausazko aldaketak eraginez.

Orokorrean, algoritmo genetikoaren erabilera abantailak sortzen ditu sistemetan, egonkorragoak direlako, sentiberak izateaz gain sarreren aldaketa txikiak edo zaratek sor ditzaketen matxurei. Gainera, emaitza hobea eta esanguratsukoak sor ditzakete egoera espazio multimodal handiagotan lan egiten dutelako. Arrazoi hauengatik, bibliografian zehar ikus daiteke ibilbide plangintzan oso erabilia izan den teknika bat izan dela, baina ez da hauen aplikazio bakarra.

AV-en posizionamendua zehazteko lanetan, Moreno et al. [217]-ek sentsore ultrasonikoen bidez lokalizazioa gauzatzeko sistema bat aurkezten dute, barrualdeko inguruneetarako. Garatzen duten algoritmoa iragazki ez-lineal iteratiboan oinarritzen da, mapa baten eta forma geometrikoen artean bateragarritasunak bilatuz ibilgailuaren posizioa eta orientazioa zuzentzeko. Iragazki ez-lineala algoritmo genetikoan sostengatzen da, posizio optimoak aurkitzeko metodo bezala. Biltegi baten moduko barneko ingurune batean, Gao et al. [218]-ek AGV-a lokalizatzeko erabil daitezkeen ezaugarri argirik ez daudela diote. Beraz, forma geometrikoak erabili ordez, erreferentzia puntuen kokapen optimo bat egitean zentratzen dira, lokalizazio fidagarri bat egiteko. Lehendabizi, biltegiaren ingurune praktikoa hartzen dute kontuan, erreferentzia-puntuak kokatzearen arazo optimizazio problema batean bihurtuz. Horrela, lokalizazioko oinarritzeko unitate bakoitzaren arteko aldea maximizatzen dute. Ondoren, algoritmo genetiko bat erabiltzen dute optimizazio arazo hori ebazteko. Azkenik, lortutako erreferentzia puntuak eta mapan gordeta daudenak konparatzen dira triangulazio metodo bat erabiliz lokalizazioa lortzeko.

Nan et al. [219]-ek, sare bateko nodo sentsoreen kokapena ezagutzea funtsezkoa dela zeregin askotarako diote, eta beraiek aurkeztutako lanean haririk gabeko sentsore-sarean, nodoak kokatzeko arazoaren zehaztasuna ebaluatzeko erabili ohi den algoritmo genetikoaren bertsio kodetu erreala bat deskribatzen da. Gainera, doikuntza funtzio bat eta eragile genetikoak ere diseinatzen dituzte. Ildo beretik, Najeh et al. [220]-ek, aplikazio errealean haririk gabeko sentsore-sareen bidez lokalizatzeko duen garrantzia kontuan izanda, irismenik gabeko algoritmo berri bat aurkezten dute, GA-en abantailak aprobetxatzen dituen, nodo baten posizio ezezagun baten kalkuluan erabiltzen diren funtzioak optimizatzeko.

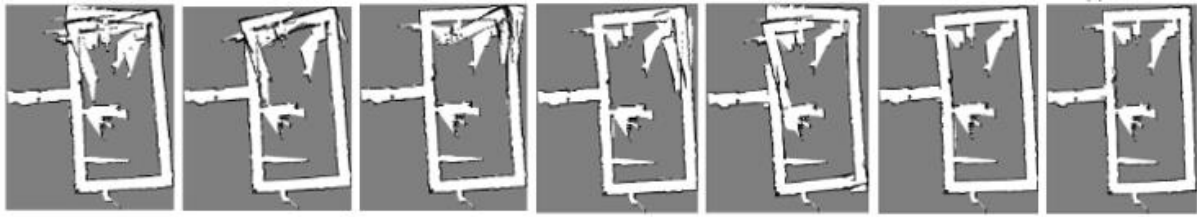
WSN gailuen *hardware* mugak direla eta, irismenik gabeko lokalizazio soluzioak aztertzen ari dira, irismenean oinarritutako ikuspegi garestien alternatiba errentagarri gisa. Hala ere, teknika horiek lokalizazio akats gehiago izaten dituzte heinetan oinarritutako algoritmoekin konparatuz. Distance Vector Hop edo DV-Hop algoritmoa, tartarik gabeko lokalizazio algoritmo tipiko bat da, jauzi distantziaren zenbatespena erabiltzen duena. Honetaz baliatuz, Peng et al. [221]-ek, DV-Hop algoritmoaren hobekuntza bat proposatzen dute algoritmo genetikoan oinarrituta, lokalizazioaren zehaztasuna hobetzeko. Algoritmo berbera erabiliz haririk gabeko sentsoreentzako, Sharma et al. [222]-ek, nodo-zabaleren batez besteko tamaina aldatzea proposatzen dute, zuzenketa faktorea eguneratuz eta jauzien tamaina aldatua algoritmo genetikoaren bidez optimizatuz. Lokalizazioaren lehen fasean arrakastaz aurkitu diren nodoak gainera, nodo laguntzaile bilakatzen dituzte eta energetikoki eraginkorra den sistema bat garatzea lortzen dute. Ikuspuntu energetiko hori Romozi et al. [223]-en lanean ere kontsideratze da, non algoritmo genetikoaren erabiltzen den nodoen posizionamendua sortzeko WSN sareetan.

Lokalizazioan oinarritutako algoritmo genetiko bat proposatzen dute Zhang et al. [224]-ek, bi eragile genetiko berri hartzen dituen: erpin bakarreko mutazioa eta jaitsieran oinarritutako gurutzatze aritmetikoa. Simulazioan lortzen dituzten emaitzek gainera, posizionamendu zehatza lortzen dutela erakusten dute. Tokiko kokapen-sistemetan, aldi baterako neurketetan oinarritutako kokapen-arkitektura asinkronoek gero eta garrantzi handiagoa hartzen ari direla diote Díez-Gonzalez et al. [225]-ek, arkitektura hauek garrantzi berezia izanez AGV-en zehaztasuneko aplikaziotan. Ideia horrekin, hainbat inguruneetara egokitzeko behar den malgutasuna duen algoritmo genetiko bat proposatzen dute. Algoritmo hau beraz, nodoen lokalizazioa hobeto zehazteko erabiltzen dute, emaitza onak lortuz.

Inguruneen mapa lortzeko, algoritmo genetikoetan oinarritu dira Zhao et al. [226], robot anitzek kooperatiboki mapak sortzeari buruzko bi ikerketa eremu aztertuz: alde batetik, esplorazio eskualdeen banaketa metodoa eta, bestetik, esplorazio robotikoaren nabigazio estrategia. Horrela, behin eta berriz gune berbera esploratzea ekidin daiteke, eskualde hori optimizatzeaz gain. Mapen fusioa egitea ere proposatzen dute Sun et al. [227]-ek, aldagai anitzeko geruzen algoritmo genetiko gurutzatu batean oinarrituta. Hiru dimentsiotako LiDAR sentsoreak erabiltzen dituzte informazioa jasotzeko, lauki-mapen mosaikoaren arazoa irudi bizidunen bat-etortze problema batean bihurtuz, maparen eraikitzea aurrera eramateko. Hau egiteko, ingurunearen eredu matematikoa eraiki behar da eta helburu funtzioa definitu, azken hau optimizatuko duen algoritmo genetiko diseinatzeaz gain. Irudiak erabili nahi izanez gero, kontuan hartu behar da pixel nahasiak egon daitezkeela eta, sailkapen arin batek pixel horiek lurzoruko zein klasetakoak diren zehaztu dezakeela. Azpi-pixelen mapaketa, pixel misto horietan dagoen informazioa erabiltzeko teknika bat da eta horrela, irudi garbiago bat lortu daiteke. Hori jakinda, Mertens et al. [228]-ek, pixelak azpi-pixeletan zatitzen dituzte, lurzoruaren klase zatikiak adieraziz. Ondoren, mapa eraikitzeko, espazio mendekotasunaren ustearekin konbinatutako algoritmo genetiko bat erabiltzen dute, kokapen bat esleituz azpi-pixel bakoitzari.

Mapa topo-geometriko lokalak konbinatuz mapa todo-geometriko global bat eraikitzea lortzen dute Boada et al. [229]-ek. Mapa lokalak Voronoi Diagrama Lokalak (LVD-*Local Voronoi Diagram*) erabiliz eraikitzen dira, telemetro laser baten neurrietan oinarrituz. Sortutako mapa horietan gordetzen den informazioa eta sentsorearen neurketak, robotaren posizioa zuzentzeko erabiltzen dira. Ondorioz, mapak eta lokalizazioa batuz, bi pausutako SLAM algoritmo bat lortzen dute. Gainera, lokalizazio algoritmoak GA-etan oinarrituta dago. SLAM-ari irtenbidea aurkitzeko Duckett [230]-ek algoritmo genetiko bat deskribatzen du, zeinetan soluzio hautagaien populazioa pixkanaka doitzen den globalki optimoa den irtenbide bat aurkitu arte. Optimizazio global honetan, helburua, espazioan robotaren mapa posibleen bilaketa da. Honetaz aparte, algoritmo genetikoan, gaitasun balioak funtzio heuristikoen baten bidez kalkulatu dira, mapa hautagaien trinkotasuna neurtuz.

Elkartze genetikoak bilatuz, Mingas et al. [231]-en algoritmoak aldebereko mapatze eta lokalizazioa egiten du *scan-matching*-a eginez gaitasun genetiko ebaluatzeko. *TinySLAM* izenpean, El Hamzaoui et al. [232]-ek ere beraien algoritmoa aurkezten dute baina kasu honetan IML (*Incremental Maximum Likelihood*) printzipioan oinarritzen dira, laser sentsore baten datuak erabiltzen dituen gainera, robotaren posizio gertagarriena estimatzeko bi dimentsiotako mapa batean. Begum et al. [233]-en kasuan, proposatzen duten algoritmoa SLAM ebolutibo bezala definitzen dute eta uhartereduko algoritmo genetiko batean oinarritzen da. SLAM-eko korrespondentzia arazoa hautespen naturalaren propietatea ustiatuz konpontzen da eta horri esker, emaitza hobekak lortzen dituzten norbanakoek baino ez dute bizirik irauten. Modu horretan, **Irudia 19**-an ikus daitezkeen emaitzak lortzen dira generazioak aurrera egin ahala. Gainera, datu sentsorialak modu inkrementalean prozesatzen dituzenez, *online* lan egin dezakeen sistema bat da.



Irudia 19: SLAM genetikoaren eboluzioa generazioak aurrera egin ahala ([233]-ko iruditik moldatua)

Liu et al. [234]-ek ingurunea kurba multzo gisa irudikatzen dute, non optimizazio problemaren aldagaiak robotaren helburuak eta kurbak deskribatzen dituzten parametroak diren. Laser baten bidez egiten dituzte irakurketak eta guzti hauek, irismen balio maximoa itzultzen duten laser-izpiak barne, xede funtzioan sartzen dituzte. Optimizatu beharreko azken funtzio hau etenuneak dituenez, ezin dela gradiente estandarretan oinarritutako ikuspegi bidez ebatzi diote, beraz, algoritmo genetikoan oinarritutako metodo bat aplikatzen dute. Ibilgailu autonomoan kokatuta dagoen laser sentzore batekin ere Wu et al. [235]-ek GA bat erabiltzen duen SLAM bat diseinatu dute, laserretik ingurune ezaugarriak deskribatzeko informazioa lortuz. Kromosoma forma berri bat aurkezten dute robotaren mugimenduak laser puntuetan eragindako aldaketak irudikatzeko eta, modu horretan, algoritmo genetikoak, inguruko ezaugarriak eta robotaren mugimenduak lotzea lortzen da. Gainera, berrabiarazteko mekanismo bat ere garatzen da, kromosoma guztiak berrabiaraziz soluzioaren aniztasuna handitzeko, parekatze prozesuarekin bat lan egiten duena. Okupazio-laukian eta sonarrean oinarrituz Mu et al. [236]-ek metodo praktiko bat aurkezten dute. Algoritmoak, Kalman iragazki bat erabiltzen du ibilgailuaren mugimenduaren egoerak estimatzeko. Lehendabizi, SLAM metodoak, nabigazio sentzoreetatik datozen datuak bateratzen ditu mugimenduaren egoerak aurreratzeko. Ondoren, partikula multzoa optimizatu dituen algoritmo genetiko bat erabiltzen da, sonararen esplorazio datuak eta mapa laukiduna eratzeko. Horrela, iragarritako egoerak zuzendu daitezke mapa eguneratzeaz gain.

Modu labaingarriko kontrola eta SLAM klasikoak konbinatzen dute Ortiz et al. [237]-ek aurkezten duten ikerkuntzan. Konbinazio honek, aldibereko mapatze eta lokalizazioak dituen ziurgabetasunak gaitz ditzake. Gainera, metodo honek ere ibilbideak planifikatzeko aukera ematen du. Ildo beretik doa aurkezten dute beste lan bat, zeinetan SLAM-ean oinarritutako ibilbideak planifikatzeko metodo bat proposatzen duten, algoritmo genetikoak erabiliz. Metodo honek, modu labaingarriko metodoaren sendotasunaren abantailak eta algoritmo genetikoaren konbergentzia-gaitasun ona ustiatzen ditu (ikus Ortiz et al. [238]).

Gaur egun, algoritmo genetikoak bilaketa problemak ebazteko oso erabiliak direla diote Chakraborty et al. [239]-ek, bideratze eta optimizazio arazo ugari aplikatuz. GA-ak era askotako emaitza ia optimoak eman ditzakete, eta hauek, denbora jakin batean lor daitezke. Beraiek, kontzeptu hauek aprobetxatuz, algoritmo genetiko bat proposatzen dute aldi berean ibilbide alternatibo ugari aurkitzeko, hainbat irizpideen arabera: distantzia biderik laburrena, bihurtze kopuru gutxien dituen, etab. Honek garrantzi handia dauka, aplikazioaren arabera behar ezberdinak existitzen bait dira. Ingurune estatiko batean ibilbide plangintza egiteko Lamini et al. [240]-ek gurutzatze eragile hobetu bat aurkezten dute. Horrela, algoritmoaren konbergentzia goiztiarra saihestea lortzen dute, eta gurasoak baino doitzeko balio handiagoa duten bide egingarriak lortzen dira. Gainera, distantzia, segurtasuna eta energia hartzen dute kontutan bideetako irizpide bezala, hori betetzeko gaitasun funtzio bat diseinatu. Zuzenean algoritmo genetiko osoaren hobekuntza proposatzen dute Han et al. [241]-ek ibilgailu autonomo anitzen ibilbideen planifikaziorako. Berrikuntza, hiru aldaketa gurutzatzeko eragile euristicokoak sartzean oinarritzen da, informazio gehiago lortuz eta ondorioz, emaitza optimizatuagoak.

Bakdi et al. [242]-ek aurkezten duten lanean bi Kinect kamera erabiltzen dira ibilgailu autonomoaren gainean, *offline* ibilbide kalkulua gauzatzeko. Robotak beraz, ikusmen sistemak emandako sakontasun informazioa erabiltzen du inguruko gunea zehaztasunez modelatzeko, irudiak prozesatzeko tekniken bidez. Algoritmo genetikoa, horrela, talkarik gabeko ibilbide egokiena aurkitzeko gai da, robotaren hasierako egoera bat, jatorria dena, eta azken konfigurazio bat lotuz, helmuga. Bai ingurune estatikoetan, bai dinamikoetan ibilbide optimo bat bilatzeko gai den teknika proposatzen dute Hu et al. [243]-ek. GA-a kasu honetan ezagutzen oinarritzen da, domeinuaren ezagutza sartuz eragile espezializatuetan, non batzuek, tokiko bilaketa teknikak ere konbinatzen dituzten. Ingurune estokastikoen kasuan, Elshamli et al. [244]-ek algoritmo genetiko planifikatzaile bat diseinatzen dute, luzera aldakorreko irudikapen batean oinarrituta, eta hartan zeinbait eragile ebolutibo aplikatzen dituzte. Doitze funtzio genetiko bat erabiltzen da ere, arazoaren helburu guztiak konbinatu ahal izateko.

Ibilbide kalkularen arazorako algoritmo genetikoaren erabilera eraginkorra dela demostratzen da Santiago et al. [245]-en ikerketan. Mapa bat kargatzen dute lehendabizi, ondoren algoritmo genetikoaren parametroak aukeratuz. Aldagai hauek azertu egiten dira, arazoarentzako zein balio-multzoko den gehien egokitzen dena zehazteko. Hasiera eta helmuga posizioak definitzen dira eta penalizazioak kontsideratzen dira ibilbideak oztopo batekin topo egiten duen kasuetarako. Lehenik eta behin, robot mugikorraren mugimenduaren gaineko lauki-sarearen eredurako ingurune globala eraikitzen da Liu et al. [246]-en kasuan. Ondoren, algoritmo genetiko klasikoa aplikatzen da ibilbideak planifikatzeko mapa laukizuzenean, eta optimizatu egiten da.

Ingurunean objektuak dauden kasuetarako, algoritmo grafikoetan oinarritutako soluzioek kalkulu-potentzia handia kontsumitzen dutela diote Kwaśniewski et al. [247]-ek. Gainera, sare neuronalak edota *Fuzzy* logikak eraginkortasun txikia dutela ingurune konplexutan ere aipatzen dute. Robotak eraikitako bi dimentsiotako mapan oinarritzen dira ibilbidearen ereduak kalkulatzeko. Honetaz gain, erabiltzen duten algoritmo genetikoaren eraginez sortzen den ibilbidea ezberdina izan daiteke mapa bererako, algoritmoaren ausazko izaeragatik. Sedighi et al. [248]-oztopo saiheste lokalean oinarritzen den ibilbide plangintza proposatzen dute, baliozko ibilbide bat aurkitzeaz gain, optimoa izan dadin bilatuz. Optimizazio hori bilatuz ere, oztopoen saiheste lokalerako Geisler et al. [249]-ek, GA-an oinarritutako *software* baten garapena deskribatzen dute. Lan honetan, kodifikazio teknika berri bat proposatzen da, algoritmo genetikoaren informazio edukia optimizatzeko garatua. Oztopoak saihesteko beste sistema bat aurkezten da Ghorbani et al. [250]-en lanean GA-ak erabiliz, baina kasu honetan ingurune global baterako. Konplexutasuna apaltzeko, ibilbidearen igarotze-puntuak bi dimentsioko kodifikazioa dimentsio bakarreko kodetzean bihurtzea proposatzen dute, eta talkak saihesteko eta distantzia laburrena aurkitzeko gaitasuna funtzio batean integratzen dute.

Robot mugikorra ingurune laukidun batean egonda, hasierako puntu baten eta amaierako puntu baten arteko ibilbide egokiena aurkitzen laguntzeko hasierako ikerketa aurkezten dute Al-Taharwa et al. [251]-ek. Hauen kasuan berrikuntza, algoritmo genetikoaren bidez, hasiera eta amaieraren artean egin beharreko urrats kopurua murrizteko ideia da, emaitzaren optimotasunari eutsiz. Tu et al. [252]-ek proposatzen duten algoritmoaren ezaugarri nagusia kromosomaren luzera aldakorra da. Sare-laukizuzenean banatzen dute ingurunea ere, eta helburuaren eta oztopoen kokapeak sartzen dira. Modu honetan, sarearen gelaxka bakoitza gene bat bezala hartzen da eta kromosoma bateko gene kopurua inguruneak baldintzatzen du. Mutazio eragile berri bat aurkezten dute Tuncer et al. [253]-ek, ingurune dinamikoaren ibilbide planifikaziorako aplikatzen dutena. GA sinpleen edo hobetutako mutazio eragile batzuen ausazko eragileak bide bideraezinak sortu ditzakeen bitartean, diseinatzen duten eragileak ez du halako egoerak eragiten eta konbergentzia goiztiarra saihesten du.

Nagib et al. [254]-en lanean nodo eta lotura bidez adierazita dagoen mapa bat erabiltzea proposatzen da. Helmugaren eta oztopoen kokapenak bi dimentsioko nabigazio ingurunean ematen dira, eta ibilgailua pasako den sareko nodo bakoitza, kode bitarraren bidez adierazten den gene batean bihurtuko da. Horrela, kromosoma bateko gene kopurua mapan dauden oztopoen menpe egongo da, beraz, luzera finkoko kromosoma bat erabiltzea gomendatzen dute. Eraitza bezala, distantzia laburrena duen ibilbidea lortzen da. Kodifikazio erreala erabiltzen dute Shi et al. [255]-en ibilbide plangintza dinamikoko eskeman. Horretaz gain, gaitasun funtzio bat eta eragile genetiko espezifikoak aplikatzen dituzte, hiru faktore kontutan izanda: talken saihestea, distantziarik motzena eta ibilbideko leuntasuna. Gainera, erabiltzen duten kodetzearen teknika algoritmo genetiko konbentzionalaren konplexutasun konputazionala murrizten du. Halaber, datu bidimentsionalak eta dimentsio bakarrekoak proiektatuz bilaketa azkartzen dute, ondorioz, bilaketa-espazioaren tamaina murriztuz. Sare neuronalekin batera ere erabili daitezke algoritmo genetikoak Du et al. [256]-ek demostratzen duten bezala. Robot mugikor baten lan eremuaren informazioarekin sare neuronalaren eredu eraikitzen da eta eredu horrekin, oztopoak ekidingo dituen ibilbide baten eta ereduaren irteeraren arteko erlazioak finkatzen dira. Ondoren, ibilbideko igarotze-puntuen bi dimentsioko kodifikazioa dimentsio bakarrekoan bihurtzen da eta talkak saihesteko ibilbidearen eta distantzia laburrenaren egokitasuna gaitasun funtzioan integratzen dira.

Algoritmo genetikoak MPC ez lineal bat bezala formulatzen dute Arrigoni et al. [257]-ek, denbora errealeko ibilbide planifikatzaile bat bezala aurkeztuz. Formulazio matematikoa deskribatzerekin, MPC ez linealaren problemaren zenbakizko ebazpena, proposatutako lanaren alderdi berritzailea demostratzen dute, algoritmo genetikoaren estrategia berritzaile bat lortuz emaitzatzat. Beste aldakuntza bat Tsai et al. [258]-ek diseinatzen dutena da. Lan honetan, ingurune egituratu batean ibilbidea lortzeko elitezko algoritmo genetiko paralelo bat erabiltzen da, bi elitezko GA izanda migrazio eragile batekin batera. Honek abantailak aurkezten ditu: populazioaren aniztasun hobearen mantentzea, konbergentzia goiztiarra inhibitzea eta ohiko GA-ekin alderatuz, paralelismoari eustea. Honetaz gain, planifikatzaileak sortutako hasierako ibilbide bideragarria leundu egiten da linea kubikoaren teknikarekin, talkarik gabeko ibilbide jarraitu optimoa eraikiz.

Jerarkietan oinarritutako algoritmo genetiko baten erabilera proposatzen da Wang et al. [259]-en kasuan, oztopo estatikoak dauden ingurune baterako. Hauen ideia nagusia, oztoporik gabeko kurbadura kotatuko biderik laburrenari buruzko Dubins-en teoremaren aplikazioa da eta hori erabiltzen dute planifikatzailea diseinatzeko. Modu honetan, Dubins-en teoremarekin problemaren eredu sinplifikatzea lortzen dute eta algoritmo genetikoak ibilbiderik hobeena bila dezake. Jerarkiak kromosomen egiturari aplikatzen dira ingurunean ibilbide posibleak adierazteko, eragile genetiko bereziak izanda modulu bakoitzerako non, bilaketa egiten den bitartean gauzatu ezin diren kromosomak "zigortzeko" estrategia bat barneratzen den. Panda et al. [260]-en ikerkuntzan portaera dinamikari oinarritzen dira eta, horrela, robot mugikorraren mugimenduaren plangintza dinamikoen arazoa ibilbide planifikazio eta kontrol sistema integratuaren kontrol arazo bihurtzen dute. Gainera, mugimenduaren plangintza dinamikoen problema optimizazio arazo batean bihurtu daitekeela ere diote, robotaren azelerazio tartean. Hau bilduz, lan honetan, ikuspegi desberdin bat eskaintzen da eta ibilbidea planifikatzearekin batera robotaren mugimenduak ere hartzen dira kontuan.

Robot mugikorraren mugimenduen plangintzaren arazoa planteatzen dute Kumar et al. [261]-ek. Erregresio analisiak eta algoritmo genetikoak erabili dituzte ibilgailuaren kontrolagailuak diseinatzeko. Erregresio kontroladorearen kasuan, oztopoetarainoko ezker-eskuin eta aurre distantziak kontuan hartuz garatzen da, AV-a erreferentziatzat hartuta eta biraketa angelu optimizatua kalkulatzeko du helburutzat. GA kontrolagailuaren kasuan, oztoporik hurbilenera dagoen distantziatik abiatuta garatzen da eta hurrengo AV-aren posiziorik onena kalkulatzeko du. Gainera, beste robot

batzuekin talkak ekiditeko, Petri sare bat ere implementatzen da. Yoshikawa et al. [262]-en kasuan, hiru fasetan banatzen dute beraien sistema. Alde batetik, algoritmo genetiko eta Dijkstra algoritmoa konbinatzen duen teknika hibrido bat aurkezten dute, kalitatezko bidea lortzeko. Bestetik, proposatutako nabigazio sistemak, azken helmugara iritsi baino lehen, zenbait igaro-puntu dituen ibilbidea gauzatzea ahalbidetzen du. Azkenik, eragile genetiko berriak sartzen dituzte gene “hilgarri” baten sortzea ekiditeko.

Mousavi et al. [263]-en hitzetan, AGV-en atazen esleipena, eragiketen itxaron-denborak kontuan izanda, AGV-en programazio prozesua definitzen du. Horretaz gain, ibilgailu autonomoen helburu anitzeko programazioa prozesu konplexu eta konbinatorioa da. Eredu matematiko bat diseinatzen dute beraz, algoritmo genetiko eta PSO algoritmoekin integratuz, ibilgailuen atazen programazioa optimizatzeko. Horrela, fabrikazio denborak eta erabili beharreko robot kopurua kontrolatzea lortzen dute nabigazioa egiterakoan. Jafar Jalali et al. [264]-ek, imitazio bidezko ikaskuntza robot autonomo adimendunak irakasteko eta kontrolatzeko paradigma eraginkorra dela diote. Arazoa, prozesatu gabeko pixeletatik abiatuta, politika integrala erakusteko adibide egokiak lortzea zaila dela da. Problema honi irtenbidea aurkitzeko, hauen ikerketan algoritmo genetikoaren erabilera aplikatzen da, sare neuronal sakon baten arkitektura optimoa lortzeko, hiperparametroetan oinarrituta. Metodoaren eraginkortasuna frogatzeko, ibilgailu autonomo baten nabigazioaren entrenamenduan erabiltzen dute.

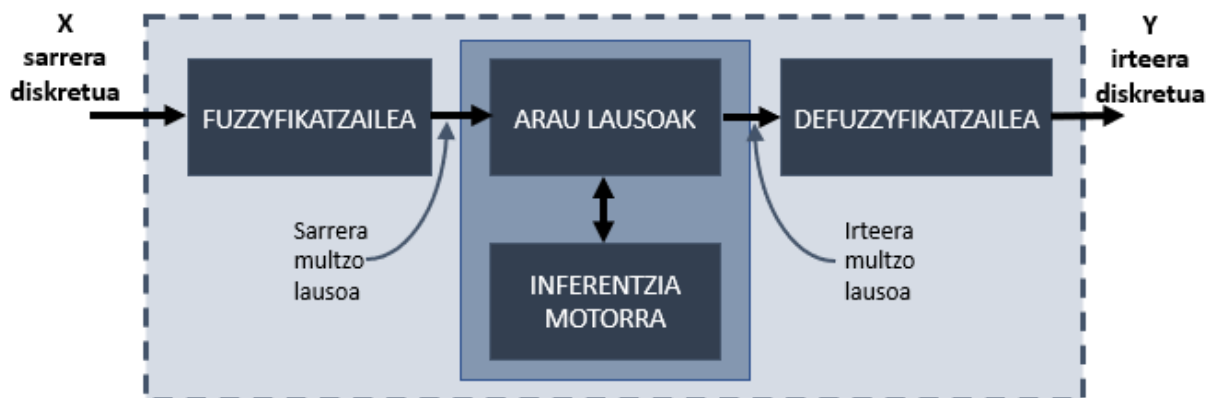
Eszenatokia kontuan izanda, Gyenes et al. [265]-ek algoritmo genetikoek oztopoak denbora errealean saihesteko zereginik izan dezaketen ikertzen dute, hau da, *online* egoeran. Hau aurrera eramateko, algoritmo familia bat lortzen dute, algoritmo genetikoak eta oztopoen abiadura eredu konbinatuz. Horrela demostratzen dute, kromosomen parametrizazio egokiarekin errendimendu ona lor daitekeela denbora errealeko objektu saihestean. Oztopoak, inguruko beste ibilgailu autonomoak ere izan daitezkeenez, Yan et al. [266]-ek elkarguneen kasua aztertzen dute. Hauek, ibilgailu autonomoak norbanako independentetzat hartzen dituzte, eta zirkulazioa kontrolaren helburua ibilgailuen pasabide eraginkorra erabakitzea da. Honetarako konbinazio asko existitzen dira eta sekuentzia eraginkor bat aurkitzea oso zaila bihurtzen da. Algoritmo genetiko baten aplikazioa da aurkezten duten irtenbidea. Sekuentzia honen optimizazio automatikoa egiten lan egin dute Cruz-Piris et al. [267]-ek ere. Arazoari heltzeko, hiru mekanismo proposatzen dituzte: bata, edozein elkargune mota modelizatzen duena da, beste, sarrera eta irteeren artean gatazka-puntu gutxien dituzten bideak kalkulatzeko duena eta azkena, ibilgailuen iriste-tasa optimizatzen duena. Guztia GA-baten bidez gauzatzen da, errendimendu maximoa lortzeko.

2.4. Fuzzy logikaren teknikaren erabilera

Fuzzy kontroladorearen logikaren erabilera oso zabalduta dago ibilgailu autonomoen esparruan. Fuzzy edo zehaztugabearen kontzeptuak “egia” edo “gezurra” izan ezin diren kontzeptuak landu ditzake, partzialtasuna erabiliz, gizakiarentzako ulergarriak diren kontzeptuekin. Honek, giza-espereintzia kontroladoreen diseinuan aplika daitekeela inplikutzen du. Logika lauso honen definizioa hurrengo hau da [268]:

“Izan bedi X multzo klasiko bat. X -ko A multzo zehaztugabea $f_A(x)$ partaidetza funtzioaren bidez karakterizatzen da. Funtzio horrek $x \in X$ puntu bakoitzari $[0, 1]$ tarteko zenbaki erreal bat erlazionatzen dio, non $f_A(x)$ -ren balioek A -ko x -ren partaidetza-maila adierazten duten. Horrela, zenbat eta hurbilago egon $f_A(x)$ 1 baliotik, handiagoa izango da A -rekiko x -ren partaidetza-maila.”

Kontrolatzaile hau diseinu oso erraza dauka, **Irudia 20**-an ikus daitekeen bezala. Gainera, diseinu simple hauek sistema konplexuetarako ere baliagarriak dira, diseinatzailearen ezagutzaren eta esperieintziaren araberakoa baita. Modu horretan, kontrolatu beharreko sistemaren eredu matematikoaren independentzia osoa dauka. Kontrolatzailearen diseinuaren sinpletasuna garrantzi handia du gainera etorkizuneko sare adimenduarentzat, oso konplexua izango dena, beraz, forma egokia hautatzeko prozesua ez da lan erraza eta esperieintziak bakarrik lagun diezaiokie diseinatzaileari [269].



Irudia 20: Fuzzy kontroladorearen osagai garrantzitsuenak

Funtsean, kontrolatzaile lausoa lau osagai nagusi ditu: fuzzifikazio interfazea, ezagutza oinarria, inferentzia mekanismoa eta defuzzifikazio interfazea. Diseinatzaileak funtzio askoren artean aukeratu dezake parte hartzeko, esaterako, forma trianguluarra, trapezoidala, Gausiarra, sigmoidala, etb. Lehenengo biak dira ohikoenak, aurkezten duten linealtasunagatik eta sinpletasunagatik. Beraz, kontrolatzaile hauen erabilerekin, adimen artifizialaren teknikak eta kontrol ingeniariakoak konbinatzen dira; adimenez auteman, arrazoitu eta planifikatu, ikasi eta jardun dezaketzen sistema autonomoak diseinatuz.

Logika lausoa nabigazio autonomoaren eremuan izan ditzakeen erabilera anitzak dira Saffiotti et al. [270]-ek dioten bezala, baina lau eztabaidagai dira nagusi. Alde batetik, portaerak sortzen dituzten modulu sendoak diseinatu behar direla argi dago eta, bestetik, modulu horietako batzuen jardura nola koordinatu jakin behar da. Gainera, sentsoreen datuak nola erabili kudeatu behar da eta azkenik, goi-mailako arrazoibidea eta behe-mailako exekuzioa nola integratu jakin behar da.

Mapak egiteko eta robot mugikorrek lokalizatzeko arazoak aztertzen dituzte Gasós et al. [271]-ek, robota mugitzen den bitartean sentsoare ultrasonikoek lortzen duten informazioaz baliatuz. Datu horiek integratzen dituzte inguruko mapa eraikitzeke, ondoren robotaren posizioa lortzeko erabiliko dena, kalkulu sistemak desplazamendu luzeetan metatuko dituen akatsak zuzenduz. Neurtzeko baldintzak ezezagunak direnez eta, normalen, objektuen mugak berreraikitzeke behaketa kopuru txikiak egongo direnez, ziurgabetasuna sortzen da ibilgailuaren posizioa definitzerakoan. Ziurgabetasun hori errepresentatzeko erabiltzen dituzte multzo lausoak, jabetza graduek definituko bait dute muga bat inguruko beste batzuen antzerakoa den.

Lokalizazioan, Alakhras et al. [272]-ek logika zehaztugabearen oinarritutako lokalizazio sistemek ikuspegi erakargarri eta egingarria eskaintzen dutela diote, ziurgabetasuna tratatzeko duen esparru matematiko eraginkorrari esker. Horretaz baliatzen dira Demirli et al. [273], mapa global batean robota non dagoen identifikatzea helburutzat hartuz, sonar sentsoarean oinarritutako informazioa soilik erabiliz. Multzo lausoak aplikatzen dituzte datu horiek modelatzeko eta triangulazio lausoa erabiltzen dute robotaren kokapena eta orientazioa detektatzeko. Caroprese et al. [274]-ek ere eredu bat proposatzen dute, zeinetan sentsoare bakoitzerako arau lausoen multzo bat ondorioztatzen da. Horrela, sentsoareen irakurketen hurbilketa zehatzagoa lortzen da behar izan ezker, adibidez, sentsoareek funtzionamendu okerra dutenean.

Chung et al. [275]-ek proposatutako lanean, robota lokalizatzeko metodoak sentsoare moduluaren irteera balioak erabiltzen ditu. Modulu horrek, girokopio baten eta magnetometro baten datuak ematen ditu, horiek ibilgailuaren uneko errotazio norabidearen angelua zuzentzeko erabili ahal izanez. Angelua zuzentzeko metodoa hiru ataletan banatu daiteke, lehenengoa sentsoareetatik lortutako balio angeluarrak kalkulatzeko dituen izanda. Bigarren zatian, sentsoare moduluaren eta robotaren angeluaren arteko errorearen ezaugarriak aztertzen dira. Azkenik, errorearen ezaugarriak erabiltzen dira logika lausoa diseinatzeko eta Kalman iragazki bat gehitzen da lokalizazio zehatza lortzeko. Bestalde, aldaketa dinamikoak daudenean, orain arteko irtenbideek ezin dutela lokalizazio zehatz bat eman diote Lee et al. [276]-ek, adibidez, gurpiletan labainketak daudenean. Akats horiek konpentsatzeko logika lausoaren bidez lagundutako elkarrekintza eredu bat proposatzen dute. Kalman iragazkiaz ere baliatzen dira, hauetako bi diseinatuz, labainketa duen eta labainketa gabeko dinamika kontsideratzeko. Ondoren, inferentzia lausoko sistemaren eredu bat eraikitzen dute lerradura zenbatesteko, inferentzia lausoko sistema neuronal egokitzaila erabiliz.

SLAM *software*-aren inplementazio bat aurkezten dute Jajulwar et al. [277]-ek, barne inguruneetako AV-en nabigazio-sistema autonomorako egokitzapen segimendu lausoaren kontrolatzailean oinarrituta. Sistema erabilgarria da robota helmugara iristeko bidearen tartean dauden oztopoak saihesteko, objektu horiek detektatzeko diseinatzen bait da *Fuzzy* kontrolatzailea. Gainera, helmugarainoko biderik laburrena lortzen da ere. Fu et al. [278]-en lanean, sistema konplexuago bat aurkezten da, zeinetan kontrolatzaile logiko lauso edo zehaztugabe (FLC-*Fuzzy Logic Controller*) bat garatzen den, entropia gurutzatuaren optimizazioan oinarrituta. Horrela, ibilgailu autonomoak GPS gabeko inguruneetan talkak saihesteko gaitasunak lortzen ditu, ikusmeneko SLAM inertzial monokularrean oinarrituta. FLC horren eginkizuna beraz, robotaren norabidea kontrolatzea da oztopoak saihesteko, informazio zehatza eta denbora errealekoa erabiliz.

Kalman iragazkiaren erabilera aplikatu daiteke logika lausoarekin batera, adibidez, Pathiranage et al. [279]-ek egiten duten bezala, Kalman iragazki lauso bat aplikatuz prozesu eta neurketa eredu pseudolinealak erabiliz aldibereko mapatze eta lokalizatorako. Prozesu ez-linealaren eredu eta behaketa eredu, modelo pseudolineal gisa formulatzen dira, eta eredu konposatu baten bidez berridazten dira. Azken honetan, eredu lokalak linealak dira *Takagi-Sugeno* eredu lausoaren arabera, eta Kalman iragazkian oinarrituta tokiko zenbatespenak zehaztea lortzen dute. Modu

honetan, tokiko zenbatespen hauen konbinazio linealak sistema osoaren zenbatespen orokorra ematen du. Watanabe et al. [280]-ek ere, *Takagi-Sungeno* eredu lausoa erabiltzen dute Kalman iragazkiarekin batera. Kasu honetan, ibilgailuaren eredu, modelo lineal ugartan deskonposatzen dute, eta hauetako bakoitza iragazkitik pasatzen dute estimazio lokalak lortzeko.

Ibilbide plangintzarako ere erabilia izan da logika lausoa Khanmohammadi et al. [281]-ek egiten duten erara adibidez, kontrol zehaztugabeko teknikak aplikatuz ingurune ezezagun batean hainbat AGV-en ibilbide plangintza aurrera eramateko. Horretarako, hibridoak erabiltzen dituzte helmuga jakin batzuetara iristeko. Ingurune ezezagunekin jarraituz, Fakoor et al. [282]-ek informazioa lortzea dute helburutzat, hori bait da ingurune horietako arazo nagusia ibilbideak aurreratzeko orduan. Markov-en erabaki prozesu zehaztugabe bat proposatzen dute prozesu hori egiteko, eraginkortasuna lortuz. Soltani et al. [283]-en kasuan industria inguruneetan ibilbideen planifikazioa aztertze laguntza esparru bat aurkezten da, garraioaren, segurtasunaren eta ikusmenaren ebaluaziotan oinarrituta. Helburu anitzeko optimizazio ikuspegi honetan, datu lausoetan oinarritzen dira langileek eta ibilgailuek zirkulatzeko erabiliko dituzten bideei buruzko erabaki estrategiko informatuagoak hartzeko. Desplazamendu distantziak eta ibilbide operatiboak ere kontsideratzen dira, ziurgabetasun maila handia duten ibilbideen egoerak aztertze aukera emanez planifikatzaileei.

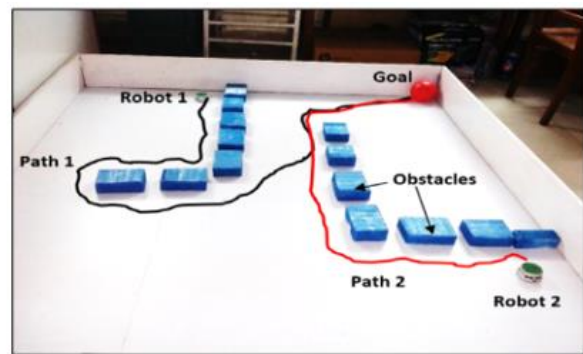
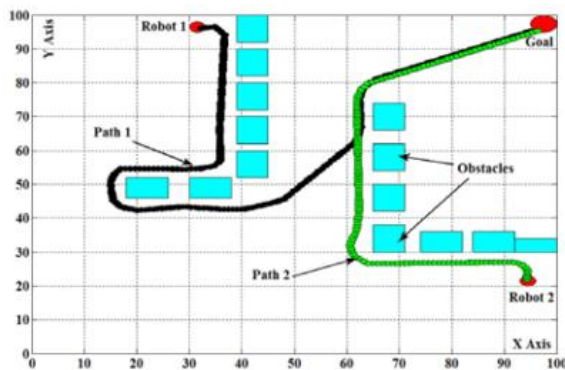
Wang et al. [284]-ek ere arrisku minimoaren ideiarekin, tokiko ibilbide plangintza egiteko eta minimo lokaletik ihes egiteko metodo bat aurkezten dute. Ikuspegi honek, portaera anitzak koordinatzeko estrategia bat hartzen du, bertan ibilbideak bila daitezen gainera arrisku gutxien duen norabidea aurkituz, guztia *Fuzzy* logikaz baliatuz. Sare neuronalen ildotik, sare neuronal lausoaren oinarritako algoritmoak konbergentzia motela eta kalkulu ugari dituela jakina da (ikus Liu et al. [285]). Irtenbidea sare neuronal lausoko algoritmo bat diseinatzea da baina partikula-multzo optimizazioan entrenatuta. Gainera, partikula-multzo tradizionalaren optimizazioa eginez, erraztasun handia dago minimo lokaletan erortzeko eta efizientzia baxua dauka. Arazo horiek gainditzeko beraz, eguneratze arau berriak diseinatzea beharrezkoa da sareko inertzia pisu eta ikaskuntza faktoreentzako.

Sonar sentsorearen ereduaz, Sun et al. [286]-ek kontrol lausoko algoritmo optimizatu bat proposatzen dute, ibilbideak planifikatzeko erabili daitekeena ingurune konplexuetan. Lehendabizi, ibilbidea ibilgailuan kokatutako sentsoreez baliatuz aurreikusten da. Ondoren, sonar horiek jasotako informazioa erabiliz, abiadura eta azelerazioa sistema lausoaren bidez lortzen da, automatikoki oztopo dinamikoak ekiditeko aukera emanez ibilgailu autonomoari. Oztopoak modu eraginkorrean saihesteko eta helmuga azkar heltzeko, objektu ugari dituen ingurune batean, Liu et al. [287]-ek ikerlan bat aurkezten dute. Honetan, sentsore ultrasonikoen informazioan oinarritutako AMR baten ibilbide plangintza aztertzen dute, algoritmo genetiko bat kontrol lausoarekin konbinatuz. Lehendabizi, eredu dinamiko eta ekuazio zinetikoak eraikitzen dituzte, ondoren, oztopo kopuruaren arabera, portaera eta saiheste arauak irudikatze. Aurreko honetan oinarrituz, *Fuzzy* kontrola diseinatzen dute ibilgailuaren norabidea zehazteko zinetika kontsideratuz. Hau, sarrera aldagaiak, irteera aldagaiak, ondasun funtzioak eta datu multzo lauso bat definituz lortzen da.

A* algoritmoaren erabilera ere oso ohikoa da ibilgailu autonomoen esparruan. Kala et al. [288]-ek inferentzia lausoarekin konbinaketa bat egitea proposatzen dute, roboten ibilbide planifikazioa ebazteko. A* algoritmoak goi-mailako plangintza egiteko ahalmena du xehetasun gutxiko mapa bat hartuz, beraz probabilitateetan oinarritutako mapa batean erabiltzen dute. Behe-mailako plangintza aldiz inferentzia lausoko sistemak (*FIS-Fuzzy Inference System*) egiten du, xehetasunak dituen grafiko baten bidez, non zehatz-mehatz ezagutzen diren oztopoek agertzeak. Kontrol lausoak murrizketa ez-holonomikoekin bat datozen ibilbideak sortzen ditu, A* algoritmoaren emaitzak gida gisa erabiliz. Sc Sarah Uttendorf et al. [289]-ek konbinazio berbera aztertzen dute, industriaren

errealitatean, benetan AGV-etan aplikagarriak diren ibilbideak kalkulatzeko algoritmo hobetu bat diseinatzeko.

Oztopoak saihesteko teknikei dagokienez lan asko argitaratu dira eta horietako bat Li et al. [290]-ena da. Honetan, logika lausoko sistema bat diseinatzen da eta oztopoak ekiditeko algoritmo bat proposatzen da, ingurune ezezagunetan ibilbideak planifikatu behar dituen robot batentzako. Sentsore ultrasonikoez baliatzen dira oztopoen distantzia eta posizioa zehazteko eta horren arabera arau lausoak finkatzen dituzte. *Fuzzy* logikaren probabilitatean (*PFL-Probabilistic Fuzzy Logic*) oinarrituz Patle et al. [291]-ek ibilbideak aurreikusten dituen algoritmo bat proposatzen dute, logika lausoaren errendimendua hobetzeko dualtasun-teknika gisa. Logika zehaztugabeak ibilbidean dauden oztopoen graduazioarekin eta probabilitatearekin elkarrengaitan du, robotaren mugimendua erabakiz. Graduazio zehaztugabeak probabilitate erabakiekin duen lotura robot mugikorra helmugarantz helarazteko oinarritzko funtzioa da. Gainera, bigarren-mailako funtzio bat gehitzen dute, ibilbidea planifikatzeko probabilitate banaketaren funtzioaren arabera. Modu honetan, ikus daitezkeen emaitzak lortzen dituzte bi AV-ek izango zituzten bideak kalkulatzeko orduan, ingurune berean, irudian ikusten den erara.



Irudia 21: PFL metodoak sortzen duen ibilbide planifikatza ([291]-ko irudia)

Planifikatutako ibilbidea trazatu nahi baldin bada, Ishikawa et al. [292]-ek funtzio bat aurkezten dute AMR-a bidearekiko zer distantziatarra dagoen kalkulatzeko, ibilgailuaren eta ibilbidearen arteko angeluaren aldea ere detektatuz. Gainera, beste funtzio bat proposatzen dute ibilgailu autonomoak objektu estatiko eta dinamikoak ekidin ditzan, haren aurretik irekita dagoen eremuaren distantzia aurreikusiz. Kontrol lausoa kasu honetan, ibilbidea iragartzeko eta oztopoak saihesteko arau egokiak hautatzeko erabiltzen da, sentsoreen informaziotik abiatuz. Barrualdeko ingurunetan ibilbidea jarrai dezan aplikatzen dute logika zehaztugabea Omrane et al. [293]-ek. Hauek bi funtzio proposatu beharrean, kontrolatzaile lauso bakarra aurkezten dute bai nabigatzeko, bai oztopoak ekiteko. Erabilitako robotak korrante zuzeneko motor bat erabiltzen du eta bederatzi sentsore infragorriez hornituta dago oztopoetara dagoen distantzia neurtzeko. Gainera, bi kodifikatzaile optiko ditu posizio erreala eta abiadura jakiteko.

AV-aren aurrealdean dagoen kamera batean soilik oinarritzen dira Rubio et al. [294] ibilbide jarraipenerako ikuspegi lauso bat egiteko, robot ez-holonomiko batez baliatuz. Hiru pausutan banatzen dute metodologia. Lehenengoan, irudia lortu eta prozesatzen da egin nahi den ibilbidea detektatzeko eta isolatzeko, erabilgarria ez den informazioa ezabatuz. Bigarren etapan, trazatu beharreko ibilbidearen zatien orientazioa kalkulatzeko, azkenik, robot mugikorraren norabidea kontrolatuko duen sistema lausoa diseinatzeko. Interpolazio lausoan oinarritutako kontrolatzaile baten arau-oinarria ez du nahitaez osoa izateko beharrik Kovács et al. [295]-ek diotenaren arabera, hau da, arau zehaztugabe esanguratsuenak kontsideratu daitezke behaketa batzuetarako, ondorioz

ez izateko arriskurik gabe. Ideia horrekin, arrazoibide lausoko hurbilketa metodo bat aurkezten dute beraien lanean, K-H interpolazioan oinarrituta. Metodo hau, logika lausoaren zuzeneko kontroleko aplikaziotan erabili daitekeela diote duen sinpletasunaren ondorioz. Proposatzen duten ideia eraginkortasuna frogatzeko, gidatze automatikoko ibilgailu baten norabide kontrolean aplikatzen dute, ibilbidearen jarraipena gauzatuz eta talkak saihestuz.

Pradhan et al. [296]-ek zenbait robot mugikorretarako nabigazio teknikak ikertzen dituzte, erabat ezezaguna den ingurune baterako logika lausoa erabiliz. *Fuzzy* logikako kontroladoreak garatu eta erabiltzen dituzte, partaidetza-funtzio ezberdinez baliatzen direnak roboten nabigazioa gauzatzeko. Lehenik, kontrolatzaile lauso bat erabili dute, lau sarrera-kide mota, bi irteera-kide mota eta hiru parametro dituena. Ondoren, bi kontrolatzaile zehaztugabe garatzen dituzte sarrera eta irteera-kide berdinekin baina bost parametroekin. Guzti, hau aurrera eramateko sentsore ultrasonikoak erabili dituzte oztopoen distantziak neurtzeko eta sentsore infragorri bat helburuaren norabidea detektatzeko. Beste robot mugikor batzuk dituen ingurune baterako Parhi et al. [297]-ek ere logika lausoan oinarritutako nabigazio sistema bat proposatzen dute. Kasu honetan, roboten kontrolatzaileetan integratutako arau lausoen haien arteko talkak ekidin ditzakete, baina Petri sare bat ere barneratuta daukate segurtasun gehiago eman dezan.

Petri sareak gai dira ia edozein sistema ulergarri modelatzeko. Gainera, sistema hori deskribatu edo exekutatzeko idealizatutako ekintza-sekuentzia hobeto ulertzeko bidea eman dezakete, dituen gaitasun analitiko ahaltsuei ezker. Kim et al. [298]-ek aldiz, benetako munduaren baldintzak idealak eta sendotasunik gabekoak izatea arraro dela diote, beraz, beraien esanetan, Petri sareen modelatze eta simulaziorako oinarritzko propietateak eskasak gerta daitezke zenbait aplikaziotarako. Horretarako, logika lausoa robotaren nabigazio autonomorako algoritmoa hobeto modelatzeko erabiltzen dute. Beraz, zenbait algoritmo zehaztugabe proposatzen dituzte plataforma robotiko mugikor batean ezartzeko, zeregin jakin baten beharrezko ekintzen sekuentzia optimizatzeko. Gainera, biltegi batean dauden robot mugikor mota ezberdinak kontrolatzeko ere logika zehaztugabea lantzen dute Faisal et al. [299]-ek. Haririk gabeko nabigazio teknika bat ikertzen dute, eredu lausok erabiliz ingurune dinamiko ezezagun batean egongo diren AV-entzako, eraginkortasuna erakutsiz metodoan.

Jaradat et al. [300]-ek ere, robot autonomo baten mugimendu plangintza aurkezten dute ingurune dinamikoetan, helmuga, oztopo dinamikoak eta estatikoak bateratzen dituena. Ezberdintasuna, datu lausoetan oinarritutako eremu potentzialaren bidez egiten dutela da. *Mamdani* eta *Takagi-Sugeno* eredu zehaztugabe bi erabiltzen dituzte robot mugikorrean eragingo duten erakarpen eta aldarapen indar totalak garatzeko. Indar erakargarri eta aldaratzaileak kalkulatzeko, lau sarrera erabiltzen dira, robotaren helmugarekiko eta oztopoekiko posizio eta abiadura erlatiboak errepresentatzen dituztenak x eta y ardatzetan. Nukleo bezala *sing4echip* bat erabiltzen da Dianying et al. [301]-en lanean, non FLC bat garatzen den. Hori egiteko, AGV-aren kontrol aritmetika aztertzen dute kontrol zehaztugabeko teknologia auto-doigarriaren bidez.

Beste alde batetik eta tekniken konbinaketak eginez, logika lausoko ikuspegia duen kontrolatzaile bat aurkezten dute Kelkar et al. [302]-ek, norabidea eta abiadura kontrolatzeko, baina ultrasoinuak detektatzen dituen ikuspegi neuro-lauso bat gehitzen dute. Modu hoetan, sistema modular baten abantailak eramangarritasunarekin eta garraiaagarritasunarekin lotuta daudela esan dezakete, edozein ibilgailu, autonomo bihurtu daitekeela gutxienezko aldaketekin ziurtatuz. *Singleton 1* motako logika lausoko sistemaren kontrolatzaile bat eta *Fuzzy-WDO (Wind Driven Optimization)* bat aurkezten dute Pandey et al. [303]-ek nabigazioaren arazoa ebazteko. WDO algoritmoa kontrolatzaile lausoaren sarrera eta irteeraren partaidetza-funtzioen parametroak optimizatzeko eta sintonizatzeo erabiltzen da, oztopo estatiko eta dinamikoak saihesteko gaitasuna lortuz.

PID kontrolatzaile batentzako sistema baten errendimendu estatiko eta dinamikoaren artean oreka bat lortzea oso zaila denez, Li et al. [304]-ek PID kontrolatzaile lauso bat proposatzen dute, ibilgailu autonomoen kontrol sistemak lehunago eta egonkorragoak izan daitezzen. Lehenik eta behin, kontrol lausoaren printzipioan oinarrituta, PID kontrolatzaile lausoa diseinatzen dute, hizkuntza zehaztugabea, memoria funtzioa eta arau lausoak barneratuz. Ondoren, irteerako balio zehaztugabeak ohiko PID kontrol sisteman sartzen dira, fuzifikazioaren, kontrol lausoaren eta erabaki zehaztugabeen bidez. Azkenik, AGV-en eredia ezarriz, Kalman iragazkia erabiltzen dute hurrengo unean ibilgailuak izango duen posizioa aurreikusteko, zarataren eragina murriztuz. Kodagoda et al. [305]-en kasuan PD eta PI kontrolatzaile lausoen diseinuan lan egiten dute, ibilgailu autonomo baten norabide eta abiadura kontrolatzeko. Kontrol-legeak sintetizatzeke logika lausoa erabiltzeak, besteak beste, kontrol-heuristikoa sartzea errazten du eta, aldi berean, egonkortasuna bermatzen du. Gainera, abiaduraren kontrola eta norabidearena banatzen da horrela eta balaztatze-kontrol bat erraz sartzeko aukera ematen du.

Sare neuronalak erabiliz, Shi et al. [306]-ek sentsoare anitzen fusioaren bidez ikuspegi berri bat aurkezten dute, sare neuronal lauso bat erabiliz AV-ak oztopoak saihestu ditzan. Proposatutako eredia, aurre-kondizio sare batez eta ondorioen sare batez osatuta dago. Lehenengo sarean, arau bakoitza arau-lausoaren aurre-kondizioekin bat egiten du. Bigarren sareak aldiz, arau-lausoaren ondorioak sortzen ditu. Emaizta osoa beraz, batura haztatua da, eta pisuak arau bakoitzaren aplikagarritasuna errepresentatzen du. Modu horretan, proposatutako ereduak, informazioa logika lausoaren bidez erabiltzeko gaitasuna izateaz gain, sare neuronalen bidez ikasteko ahalmena ere badu.

Maaref et al. [307]-ek ingurune ezezagunetarako nabigazioa planteatzen dute, oinarrizko portaeren konbinazioan oinarrituta eta, portaera horietako gehienak inferentzia-sistema zehaztugabeen bidez lortuz. Proposatutako nabigatzaileak gainera, oztopoak saihesteko bi portaera konbinatzen ditu, bata oztopo ganbilentzat eta bestea ahurrentzat. Ingurune ezezagun batean ibilbideak bilatzeko dauden nabigazio sistemek oztopo-eremu konbexuetan aurkitu ohi dutela ibilbiderik laburrena diote Lee et al. [308]-ek, baina hauek huts egiten dute oztopoak ahurrak direnean. Orduan, nabigazio sistema lauso bat aurkezten dute oztopo ahurrak dituen eremuetatik ihes egin dezakeena. Sistemak algoritmo tangente bat konbinatzen du gainera, ibilbideak planifikatzeko, kontrol arau lausoeekin.

Ibilgailu autonomo batek oztopoak saihestu ditzan, kontrolatzaile lauso baten diseinua aurkezten du Lilly [309]-k ere. Kontrolatzailea beraz, objektuak ekiditeko gai da, arau lauso negatiboak ohiko positiboekin erabiliz. Arau zehaztugabe negatiboak exekutatu ordez saihestu behar diren ekintzak preskribatzen ditu. Horrela, arau positiboek oztopoak ez dauden lekutik bideratzen dute ibilgailua. *E-puck* modulurako oztopoak saihesteko ikuspegi bat aurkezten da Mohammad et al. [310]-en lanean ere, FLC-a erabiliz. Honetan, zortzi sentsoare infragorrien sarrera eta motorren abiaduraren irteera arau lausoak eraikitzeke erabiltzen dira. Rao et al. [311]-en kasuan, oinarrizko portaerak dituen AMR batentzako sistema bat sortzen dute, oztopoak saihestu eta helburuak bilatzen dituen. Ibilgailuak *Fuzzy* logika, sare neuronalak eta neuro-fuzzy inferentzia sistema egokitzaile baten bidez nabigatzen du, horrela, oztopoen ekiditeko eginkizunak errendimendua hobetzen du une bakoitzeko. Kontrolatzailearen sarrerak AMR-an kokatuta dauden sentsoaren seinaleak dira eta irteera seinaleak ibilgailuaren gurpilen abiadura angeluarra erregulatzen du.

3. Ikusmen Artifiziala Gidatze Autonomoan

Gaur egun, robotika mugikorraren eremuak bilakaera handia izan duela eztabaidatu ezin den gai bai da, eta gero eta maizago agertzen dira hauek modu autonomoan lan egiteko algoritmo proposamenak. Gainera, ikusmen sentsoreen erabilerak zeregin garrantzitsua izan du ibilgailu autonomoen zereginetan, ordenagailu bidezko *hardware* eta ikusmen algoritmoetan egindako aurrerapenak direla eta. Azpimarratzekoa da ere AI tresnen erabilera irudietan oinarritutako roboten hainbat arazo konpontzeko, bai informazio iturri bakar gisa, bai beste sentsore batzuekin konbinatuta, hala nola laser edo GPS sentsoreekin. Honen fusioa ikusmen artifiziala du emaitzat.

Ikusmen artifiziala erabili nahi baldin bada, kontuan izatekoa da irudiak datu oso dimentsionalak direla eta arrazoi ezberdinengatik aldaketak jasan dezaketela; robotaren mugimenduak eragindako aldakuntzez gain, argiztapenek, erreflexuek edo ingurunea osatzen duten objektu batzuen kokapena eta formek sortutakoak adibidez. Ondorioz, datu horiekin lan egiten duten sistemek ingurunetik informazio esanguratsua eta aldezina ateratzeko helburua izan behar dute. Aipatzekoa da gainera, bibliografian zehar YOLO algoritmoa dela erabiliena ikusmenarekin erlazioa duten aplikazioak aurrera eramateko, baina CNN sareekin ere oso ohikoa da lan egitea.

3.1. Datu-baseak eta erreferentzia-puntuak

Ibilgailu autonomoetan bisioaren aplikazioak erabili ahal izateko datuak behar dira, hau da, irudiak. Azken finean, ikusmen artifiziala, mundu errealeko irudiak eskuratu, prozesatu, aztertu eta ulertu egiten dituen teknika bat da. Informazio horretatik, inguruko ezaugarriak lortu ahal dira, gidaketa autonomoaren zereginetan beharrezkoak direnak. Bibliografian zehar ikusi den bezala, aldibereko mapatze eta lokalizazioan da bisioa gehien erabiltzen den esparrua, baina inguruko errekonozimendu hau nabigazioan ere aplikagarria da. Ondorioz, estrategia automatikoak garatu behar dira irudi ezberdinetan “patroiak” detektatzeko. Hau egiteko, ikaskuntza bat behar da, datu-multzo batetik etorria.

Datu-multzoek funtsezko zeregina izan dute ikerketa eremu honetan, ikuspegi ebaluazio kuantitatiboa ahalbidetzen bait dute, eta haien gaitasun eta mugei buruzko funtsezko informazioa ematen dutelako [312]. Zenbait datu-baseek gainera, erreferentzia-puntu zorrotzak sartu dituzte mapen berreraikitze lanetan, mugimenduaren estimazioan eta errekonozimenduan adibidez, laborategiko inguruneen eta mundu errealeko egoera zailen arteko ezberdintasunak murrizten lagundu dutenak.

Duela urte batzuk, ehunka adibide zituzten datu-multzoak nahikoak zirela uste zen, arazo askori irtenbidea ematen baitzioten. Hauekin baina, aurrerapen handia egon da, ehunka eta milaka adibide etiketatuta zituzten datu-baseen etorrerarekin. Honek, ordenagailu bidezko ikusmenaren arlo askotan gaitasun handia eman du, sare neuronal sakonak modu gainbegiratuan entrenatzeko aukerarekin esaterako. Hau erakunde askoren esker izan da, datu hauek eremu ezberdinetan sortzeko lan egin bait dute, prozesua ahal den heinean automatizatuz. Ondorioz, ibilgailu autonomoetan oso baliagarriak diren datuak daude eta **Taula 4**-ak garrantzitsuenak jasotzen ditu [313].

Taula 4: Datu-base erabilienetakoak gidaketa autonomoan

Datu-multzoa	Sentsorea	Irudi kopurua	Datuak	Ingurunea	Garapena
A2D2 [314]	6 kamera (erdikoa+ertzetakoak) eta 5 LiDAR	+40.000	Segmentazio semantikoa (38 kategoria), laino-puntu segmentazioa eta 3D kutxa mugatzaileak	Kanpoaldea	Audi (Alemania)
ApolloScape [315]	2 laser eskaner, 6 kamera (eraztunean) eta IMU/GNSS sistema	+100.000	Segmentazio semantikorako 3D puntuak (inguruneetako 25 kategoria)	Kanpoaldea	Baidu Ikerketa, Teknologia Ingeniaritzako eta Ikaskuntza Sakonaren Aplikazioko Laborategi Nazionala (Txina)
Argoverse [316]	7 kamera (eraztunean), 2 aurreko kamera estereo eta 2 LiDAR	+30.000	3D-ko jarraipen oharrak (15 kategoria) eta ibilgailuen ibilbideak iragarpenak egiteko datuak	Kanpoaldea	Argo AI, Carnegie Mellon Unibertsitatea eta Georgiako Institutu Teknologikoarekin kolaborazioan (Pensilvania)
CityScapes [317]	2 kamera estereo eta GPS sistema	+25.000	Segmentazio semantikoa (30 kategoria)	Kanpoaldea	Daimler AG R&D, Max Planck Informatikarako Institutua eta TU Darmstadt ikus-inferentzia taldea (Alemania)
Google-Landmarks [318]	Kamerak (domeinu publikoko irudiak)	+5M	Erreferentzia puntuen errekonozimendua	Kanpoaldea	Google Ikerketa (Ameriketako Estatu Batuak)
KITTI [318]	Laser eskaner birakorra, 2 kamera estereo, 2 kamera txuri/beltz eta GPS/IMU sistema	+100.000	3D kutxa mugatzaileak eta hauen sakontasuna (11 kategoria), segmentazio semantikoa eta jarraipenak egiteko datuak	Kanpoaldea	Teknologiako Karlsruhe Institutua eta Ameriketako Teknologiako Toyota Institutua (Ameriketako Estatu Batuak)
nuScenes [319]	6 kamera, LiDAR birakorra, 5 hein luzeko radar, IMU/GPS sistema	+1.000	3D kutxa mugatzaileak datuekin, segmentazio semantikoa (32 kategoria) eta laino-puntuak	Kanpoaldea	nuTomy konpainia (Ameriketako Estatu Batuak)
Oxford Radar RobotCar [320]	Radara, 2 LiDAR birakor, kamera estereo, 3 kamera, 2 2D LiDAR eta GPS sistema	+5.000	Lurreko odometria, abiadurarekin	Kanpoaldea	Oxford Unibertsitatea (Erresuma Batua)
PandaSet [321]	5 angelu zabaleko kamera, foku luzeko kamera, LiDAR birakorra eta aurreko LiDAR-a	+48.000	3D kutxa mugatzaileak (28 kategoria) datuekin	Kanpoaldea	Hesay Technology (Ameriketako Estatu Batuak)

Datu-base hauek barneratuta duten informazioarekin lan ezberdinak egin daitezke aurreko bibliografian antzeman daiteken bezala, erreferentzia-puntuez baliatuz batez ere. Gehien ikertu den zereginetako bat *mapen berreraikitzea estereoskopikoa* da, korrespondentzia estereoskopiko izenpean ere ezaguna dena baina askotan datu-multzo propioak sortzea beharrezkoa da. Scharstein et al. [322]-ek diotenaren arabera, nahiz eta korrespondentziarako algoritmo asko garatu izan diren, oraindik errendimendu eskasa dute. Horri aurre egiteko, bi fotogramako estereoskopia trinkoaren metodoen taxonomia aurkezten dute, estereoko algoritmo indibidualek daukaten osagai ezberdinak eta diseinu erabakiak ebaluatuz. Gainera, hainbat fotogrametako datu-multzo berriak sortu dituzte datu estereoskopikoekin. Häne et al. [323]-en lanean, barrualdeko inguruneak berreraikitze metodo bat aurkezten da irudi estereoskopiko bikoteetatik abiatuta, roboten nabigazioan pentsatuz. Robot batek soilik kamera estereoskopikotik jasotako seinale bisualak erabiliz nabigazioa gauzatu dezan, irudi bikoteen sakontasun informazioa ateratzea beharrezkoa dela diote, hau errepresentatuz. Beraz, sakontasun mapa gordinak lortzen dituzte eta bi mailako altuera-mapatan berreraikitzen dira, lurzoruaren altuera eta sabaiarena definituz.

Errendimendua alde batera utziz, Scharstein et al. [324], informazio global zehatza lortzea jartzen dute helburutzat aurkeztutako lanean. Konplexutasun altuko irudi estereoskopiko bikoteak lortzeko metodo bat deskribatzen dute, korrespondentzia informazio zehatza dutenak pixeletan. Argi egituratua erabiltzen dute, kalibrazio beharrik ez duena eta mapak sortzeko balio duten datu-multzo berriak aurkezten dituzte. Argi egituratua lortzeko sistema bat aurkezten dute Scharstein et al. [325]-ek beste lanetako batean, barrualdeko ingurune estatikoetako datu-multzo zehatzak sortzeko. Sistemak azpi-pixelen bi dimentsiotako korrespondentzia eraginkorra egiteko teknikak barneratzen ditu, kameraren auto-kalibrazioa eta lentearen distorsioaren modelaketaz gain. Horrela, 33 datu-multzo berrien ekarpena ere egiten dute, 6 megapixelek osatutako irudi estereoskopikoez beteak. Yao et al. [326]-ek ikaskuntza sakona bista anitzeko estereoskopian (*MVS-Multi View Stereoscopic*) errendimendu handia erakutsi duela azpimarratzen dute. Hala ere, egungo MVS ikuspegiaren muga nagusietako bat eskalagarritasuna da, hau da, memoriak kontsumitzen duen kostu-bolumenaren erregularizatzeak, ikasitako MVS-a bereizmen handiko eszenei aplikatzea zaildu egiten du. RNN baten oinarritutako bista estereo anitzeko esparru eskalagarri bat aurkezten dute, bi dimentsiotako mapen kostua sekuentzialki erregulatzeko sakontasunaren norabidean.

Datu-multzoak ere *optical flow* bezala ezagutzen den teknikarako ere sortu dira, hau da, objektuek jarraipena egiteko, ibilgailu autonomoen aplikaziotan oso erabilia dena. Aplikazio honi dagokionez, Leal-Taixé et al. [327]-ek, erreferentzia-puntu kuantitatiboen estandarizazioan lan gutxi egin dela aipatzen dute, nahiz eta oso ohikoa den metodo bat izan. Honek, aplikazio forma inkoherenteak sortzen ditu, adibidez, eskura dauden datuen azpi multzo ezberdinak erabiliz, ereduak entrenatzeko modu desegokiak aplikatuz edota ebaluazio gidoi ezberdinak jarraituz. Aurkezten duten lanean, objektuek jarraipena egiteko datu-multzo propio bat sortzeko deskribapena ematen dute, aurrez aipatutako akatsak saihestuz. Baker et al. [328]-ek erreferentzia-puntu multzo berri bat proposatzen dute ere, fluxu optikoko algoritmoen ebaluazio metodoez gain. Lan hori aurrera eramateko, lau datu moten ekarpena egiten dute fluxu optikoko algoritmoen alderdi ezberdinak frogatzeko. Alde batetik, mugimendu ez-zurrunezko sekuentziak, non fluxu erreala ezkutuko egitura fluoreszente baten jarraipenaren bidez zehazten den. Bestetik, sekuentzia sintetiko errealistak. Hirugarrenik, fotograma-abiadura handiko bideoak, interpolazio-errorea aztertze erabiliak. Azkenik, eszena estatikoen sekuentzia estereoskopiko aldatuak. Honekin batera, errore angeluarraz gain, fluxuaren azken puntuko errore absolutua eta fotogramen interpolazioaren errorearen neurriak ere kalkulatu dituzte.

Fluxu optikoa miatzeko lanetan, Janai et al. [329] pixelen arakatzea egiten dute abiadura handiko bideo-kamerarekin grabatutako laginketa trinko eta tenporaleko bolumenen bidez. Ereduak mugimendu txikien linealtasuna ustiatzen du eta oklusioak arrazoitzen ditu fotograma ugaritatik abiatuta. Modu horretan, laborategi barne eta kanpoko erreferentzia eremu zehatzak ezar daitezke, eta irudi ugaritzea ere egin daiteke mugimendu errealistaren desfokuratzearekin. Horrela, fluxu optikora aplika daitekeen datu-multzo bat argitaratzen dute, teknika honekin lortua. MOT (*Multiple Object Tracking*) kontzeptua ere sartu daiteke atal honetan, helburutzat bideoetako objektuen lauki mugatzaileak eta identitateak zenbatestea duena. Orokorrean, identitateak atalase batetik gorako puntuazioak dituzten laukiak detektatuz eta lotuz lortzen dira. Hau ez da aproposena Zhang et al. [330]-entzako eta elkartze metodo erraz, eraginkor eta generiko bat aurkezten dute, ia detekzio lauki guztiak lotzen dituen, soilik puntuazio altukoak beharrean. Puntuazio txikiko detekzio laukietan, “*tracklets*”-en antzekotasunak erabiltzen dituzte benetako objektuak berreskuratzeko eta hondoko detekzioak iragazteko. Horrela, aztarnari sinple bat diseinatzeko gai dira, 9 aztarnariz osatua eta *ByteTrack* izena duena.

Milan et al. [331]-ek, MOT-eko erreferentzia-puntuen bertsio berri bat aurkezten dute. Honetan MOT16-ko bideo guztiak arreta handiz markatu dira protokolo koherente bati jarraituz. Gainera, etiketatutako laukien areagotze esanguratsu bat eskaintzeaz gain, objektuen klaseak ugari ematen ditu ere, estandarizazioari bultzatuz eginez. Dendofer et al. [332]-ek diotenaren arabera, estandarizatutako erreferentzia-puntuak ezinbestekoak dira ordenagailu bidezko bisioko aplikazio gehienetan. Hau horrela izanda, MOT15, MOT16 eta MOT17 asko lagundu dute ikerkuntza komunitatean, datu-multzo garbi bat eta hainbat objektuen jarraipen-metodoak ebaluatzeko esparru bat zehatza sartuz. Horrekin aurrera pausu bat emateko, MOT20-ko erreferentzia-puntuak aurkezten dituzte, zortzi sekuentzia berriz osatua, objektu asko dituen ingurune batekoak.

Sare bakar batean MOT-a objektuak detektatzeko eta re-ID (pertsonek identifikazioa) ikasteko multiatza gisa formulatzea erakargarria da Zhang et al. [333]-entzat, bi lanak batera optimizatzeko aukera ematen duelako, konputazio-eraginkortasuna handitzeaz gain. Auteman dutena da, bi atazek elkarren artean lehiatzeko joera dutela da, beraz, kontu handiz ekin behar zaie. Aurreko lanek orokorrean pertsonen identifikazioa bigarren mailako ataza bat bezala tratatzen dute, baina honek zehaztasunaren galera eragiten du, sare partzial bat emaitzat emanez. Aurkezten duten lanean, *FairMOT* deitutako ikuspuntu bat proposatzen dute arazo hauei aurre egiteko, ainguratu gabeko objektuak detektatzeko CenterNet arkitekturan oinarrituta. *CenterTrack* izenpean Zhou et al. [334]-ek aldibereko objektu jarraitzea eta detekzioa egiten duen algoritmo bat aurkezten dute, sinple eta zehatza dena. Aurreko fotogramako irudi eta detekzio pare bati detekzio-eredu bat aplikatzen diote eta informazio minimo honekin, objektuak lokalizatzeko gai dira, aurreko fotogramarekin erlazionatzeaz gain, **Irudia 22**-an ikus daitekeen bezala. Meinhardt et al. [335]-en kasuan, objektuen jarraipena fotograma multzoen iragarpen problema gisa formulatzen dute, *TrackFormer* kontzeptua sartuz. Ideia honek, muturretik muturrerako entrenagarria den MOT ikuspegi bat eskaintzen du, *Autonecoder Transformer* arkitekturan oinarrituta. Ereduak beraz, fotogramen arteko datu-asoziazioa lortzen du bideo sekuentzia baten bidez, pista-iragarpenen multzo baten bilakaera egiez.



Irudia 22: Ibilgailuen jarraipena fotogramen artean ([334]-ko irudia)

Objektuen detekzioaz gain, hauen errekonozimendua egiteko edo segmentazio semantikoko informazioa duten datu-baseak ere ohikoak dira. Eskala handiko datu-multzo hauek eskuragarri izateak inpaktu handia izan du ikaskuntza sakonaren arrakastan eta objektuen sailkapen, detekzio eta segmentazio semantikoko lanetan. PASCAL *Visual Object Classes* (VOC) erronka, ikusizko objektuen kategoriak ezagutu eta antzemateko erreferentzia lana da. Ikusmen eta ikaskuntza automatikoaren ikerlariei, irudien eta idazpenen datu-multzo estandarra eta ebaluazio-prozedura estandarrak zehazten dizkie (ikus Everingham et al. [336]). Urtero, irudien klasifikazio eta detekziorako metodoen errebisioa egiten dute eta hauen ezberdintasunak azpimarratzen dituzte ikerketa lanei bultzada emanaz. Baina ez da objektuen klasifikazioa egiteko erabili daitekeen datu-base bakarra. Lin et al. [337]-ek, datu-multzo bat aurkezten dute ere objektuen errekonozimenduaren teknikaren egoeran aurrera egiteko, objektuak klasifikatzeko gaia eszenak ulertzeko testuinguru zabalenean kokatuz. 91 motatako objektu ezberdinak dituzten irudiak dira, erraz errekonozitzen direnak. Guztira, 2,5 milioi etiketa aurki daitezke 328 mila irudietan banatuta eta hauetan, kategorien, lokalizazioaren eta segmentazio semantikokoaren informazioa dago barneratuta.

Srinivas et al. [338]-ek BoTNet deituriko arkitektura bat aurkezten dute, kontzeptualki erraza dena, baina oso eraginkorra. Arkitektura honek, ordenagailu bidezko hainbat ikusmen lanen auto-arreta barneratzen du, hala nola, irudien sailkapena, objektuen detekzioa eta instantzien segmentazioa. ResNet bateko botila-lepoko azken hiru blokeetan konboluzio espazialak auto-arreta globalagatik ordezkatzeko dituztenez, ikuspegi honetan hobekuntza detektatzen da instantzien segmentazioan eta objektuen detekzioan. Gainera, latentzia murriztea ere lortzen da parametroak gutxitzen direlako. Yu et al. [339]-ek, segmentazio semantikoa informazio espazial aberatsa eta eremu hartzaile handia behar duela diote. Hala ere, oraingo ikuspegiak denbora errealean inferentzia-abiadura lortzeko espazio-bereizmena arriskuan jartzen dutela ere aipatzen dute, errendimendua kaltetuz. *BiSeNet* izendatu dute diseinatutako segmentazio sare bilaterala eta hau sortzeko, lehendabizi, oinkada txiki bat duen ibilbide espaziala diseinatu dute non, informazio espaziala gordetzen den eta bereizmen handiko ezaugarriak sortzen diren. Bigarrenik, laginketa-estrategia azkarra duen laster-bide bat erabiltzen da eremu hartzaile nahikoa lortzeko eta horrela, abiadura eta segmentazio errendimenduaren arteko oreka lortzen da.

Konboluzioa alde batera utziz ere, Strudel et al. [340]-ek *Segmenter* izena duen eredu transformatzaile bat aurkezten dute segmentazio semantikoa egiteko, lehenengo geruzan eta sare osoan testuinguru globala modelatu dezakeena. Bisio transformatzailean (*ViT-Vision Transformer*) oinarritzen dira eta segmentazio semantikora hedatu dute, irudi-adabakiei dagozkien irteera-inkrustazioen bidez. Horrela, inkrustazio horietatik abiatuz, klase-etiketak lortzen dituzte deskodetzaile lineal puntual baten bidez edo maskara-deskodetzaile transformatzaile baten bidez, bigarren honetan errendimendu handiagoa nabaritzen da.

3.2. Patroi fiduziarioak







Patroi fiduziario bat bisio sistema baten ikusmen eremuan dagoen objektu edo forma ezaguterraz bat bezala defini daiteke, irudi askotan agertzen dena eta erreferentzia-puntu edo neurri bezala aprobe txatu daitekeena. Orokorrean, hau inguruko elementuekin soilik aurrera eramatea lan oso korapilatsu bat izan daiteke beraz, markatzaileak erabili ohi dira.







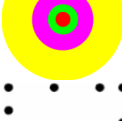








3.2.1. MARKATZAILE FIDUZIARIOAK

Markatzaile fiduziarioak, esan bezala, erreferentzia-puntu bat edo irudi bateko neurri bat emateko erabiltzen diren objektuak dira. Aplikazio desberdinetarako baliagarriak dira baina, adibidez, markatzaileak erreferentzia-puntu gisa soilik erabiltzen direnean, eredu sinpleak dira. Beste kasuetan aldiz, markatzaile fiduziarioek identifikatzaile edo mezu kodifikatu espezifiko bat barneratuta izan dezakete, informazio gehigarria emanez. Hauetako ohikoenak QR kodeak edota barra kodeak dira, eguneroko bizitzan aurki ditzakegunak [341].

Gaur egun robotikako aplikaziotan erabiltzen diren markatzaileak errealitate areagoturako (AR-*Augmented Reality*) diseinatuak izan ziren lehendabizi. Hauek erreferentzia puntu artifizialak dira, neurri eta forma esanguratsu eta jakinak dituztenak, patroi zehatz batekin. Ohiko patroiak karratuz eta borobilez osatuta daude, orokorrean forma geometrikoak jarraituz. Horrela, bit anizten kodifikazioa egiteko erabili daitezke eta identifikazio lanetan martxan jarri. Hurrengo **Taula 5**-ak markatzaile hauek jasotzen ditu, hauen zehaztasunekin.

Taula 5: Markatzaile fiduziarioen zerrenda ([341]-ko taulatik moldatua)

Izena	Ezaugarriak (Forma+Kolora)	Kodifikazioa	Markatzailea	Oharrak
AprilTag [342], [343]	Karratua+ Monokromatikoa	Topologikoa		Irudiaren segmentazio hobea, sendotasuna handituz
ARTag [344], [345]	Karratua+ Monokromatikoa	Topologikoa		Erabilera hedatuena duen multzoa
ARToolKit [346]	Karratua+ Monokromatikoa	Topologikoa		Lehenetariko markatzaile fiduziarioa
ArUco [347]	Karratua+ Monokromatikoa	Topologikoa		Liburutegi konfiguragarrien erabilera ahalbidetzen du
BinaryID [348]	Karratua+ Monokromatikoa	Topologikoa		Kodetzeko erabiltzen diren patroi bitarrak, gainera errotaziopean sendoak dira
BullsEye [349]	Zirkularra+ Monokromatikoa	Topologikoa		Emaitza sendo eta zehatzak oklusioen pean

CALTag [350]	Karratua+ Monokromatikoa	Topologikoa		Kameren kalibrazio aplikazioetarako diseinatua
CCC [351]	Zirkularra+ Monokromatikoa	Ez dauka		Markatzaile fiduziaro sinpleena
CCTag [352]	Zirkularra+ Monokromatikoa	Topologikoa		Oso sendoa oklusio eta mugimendu lausoen aurrean
ChromaTag [353]	Zirkularra+ Kolare Anitzak	Topologikoa		Detekzio ratio azkarra du baina ez da sendoa
FourierTag [354], [355]	Zirkularra+ Gris Eskala	Frekuentzia espektroa		Zehaztasunaren degradazio leunak izateko diseinatua
InterSense [356]	Zirkularra+ Monokromatikoa	Topologikoa		Sendoa argiztapen baldintzen aurrean
Multi-ring [357]	Zirkularra+ Kolare Anitzak	Ez dauka		Erresistentzia argiztapen baldintzen aurrean
Pi-Tag [358]	Karratua+ Monokromatikoa	Topologikoa		Azkarra eta nahiko sendoa oklusioen aurrean
RuneTag [359], [360]	Zirkularra+ Monokromatikoa	Topologikoa		Oso sendoa oklusioen aurrean baina detekzio abiadura motela
SIFT [361]	Karratua+ Gris Eskala	Gradientea		SIFT (Scale Invariant Feature Transform) ezaugarriak ezena bati gehitzeko diseinatuak
STag [362]	Karratua+ Monokromatikoa	Topologikoa		Elipse doikuntza erabiltzen du egonkortasuna hobetzeko
Topotag [363]	Karratua+ Monokromatikoa	Topologikoa		Pertsonalizagarriak
TRIP [364]	Zirkularra+ Monokromatikoa	Topologikoa		Kodetzeko zenbaki hirutarrak erabiltzen ditu bitarren ordez
VisualCode [365]	Karratua+ Monokromatikoa	Topologikoa		Gailu mugikorretan baliabide mugatuekin lan egiteko diseinatua
WhyCode [366]	Zirkularra+ Monokromatikoa	Topologikoa		Kodetzeko aukera ematen duen patroi zirkular sinplea

Robotika mugikorren markatzaile hauen erabilerarik hedatuena mapatze eta lokalizazioarekin du erlazioa. GNSS sistemek, VIO (*Visual Inertial Odometry*) sistemek eta aldibereko mapatze eta lokalizazioak ere arazo hau konpondu dezaketela jakina da, baina askotan hauek ez daude eskuragarri edota ez dute behar besteko zehaztasuna ematen argiztapen baldintzak edo ezaugarri bisual nahasiak direla eta. Markatzaile fiduziariorak beraz, baldintza hauek bete ezin diren lekuetan posizioaren estimazioaren kalitatea hobetzeko erabiltzen dira. Horretaz gain, markatzaile hauek zehaztasuna eta sendotasuna handitu dezakete lokalizazio sistema batean, detektatzeko ezaugarri erraz bat delako [367]. Akatsak ere era sinplean detektatu daitezke, irtenbide erraz, merke, eraginkor eta fidagarri bat izateaz gain.

Jada jakina denez, lokalizazioa funtsezko garrantzia duen robotikaren alderdi bat da ibilgailu autonomoen erabileran. Odometria gainera, AV batek egiten duen distantzia lortzeko modu arrunta da baina honek berezko akatsak ditu, hala nola, gurpilak lurrian dituen irristatzeak eragindako akatsak. Jayatilleke et al. [368]-ek erreferentzia-puntutan oinarritutako lokalizazioa GPS eta odometriarentzako osagarri oso ona dela diote eta ideia hori sustatzen dute beraien lanean. Ibilgailu autonomoa ROS sistemarekin batera erabiltzen dute parametroak lortzeko eta algoritmo matematiko eta kamera batekin batera, posizio erlatiboa lortzeko gai dira. 100-150 mm-tako erroreak dituzte beste lokalizazio metodo batzuekin konparatuz hobekuntza bat suposatzen duena, beraz markatzaileak era eraginkorra dira GPS-a erabili ezin den inguruneetarako. GPS seinalearen galeren arriskua ere aztertzen da Venegas et al. [369]-en lanean, arriskua areagotuz oztopoak dauden inguruneetan. Oztopo horietan ARTag-ak kokatzen dituzte hauen kokapena zehazteko eta Makov prozesuaren bidez nabigazio arazoa planteatzen dute. Horrela politika bat definitzeko gai dira, markatzaileak ibilgailutik hurbil daudenean mugimendu aginduak sortzeko. Politika hori *online* kalkulatu eta eguneratzen da, ingurune desordenatuetan mugitzeko aukeraz.

Barrualdeko inguruneetako hormetan kokatzen dituzte erreferentzia-puntuak Kalaitzakis et al. [370]-ek, ibilgailua puntu horietara joanez auto-lokalizazioa egiteko. Irudi digitalen korrelazioaren (DIC-*Digital Image Correlation*) teknikaz baliatzen dira, metodo zehatz bat izatea demostratu duelako neurketak egiteko. Gaurko DIC aplikaziotan, kamera estereoskopikoa erabiltzen dute oso finkoak direlako eta ARTag -a sendotasunez detektatu dezakeelako. Cantieri et al. [371]-en kasuan, markatzailea lurrian dago eta berriz ere ibilgailu autonomoaren posizionamendu zehatza bilatzen dute. Kokapen globala denbora errealeko zinematikarekin egitearen planteamendua alde batera uzten dute, satellite seinalearen hartze oso ona behar izaten delako, askotan zaila dena. Aurkeztutako lanean beraz, posizio datuen sistema osagarri bat proposatzen dute, markatzaile fiduziariorak oinarrituta posizionamendu sistemaren fidagarritasuna handiagotzeko.

Klein et al. [372]-en lanean adibidez, markatzaile fiduziarior asko lurzorutik sakabanatuta aplikatzen dira eta ikaskuntza automatikoaren ikuspegian arreta asko jartzen dute. Ideia, ibilgailu autonomoak duen kamera batek pose erlatiboa lortzea da, ArUco-ei dagokienez. Ondoren, robot mugikorraren posearen estimazioa planteatzen dute ML-aren bidez. Lortzen duten errorea eskala milimetrikoa da eta gainera, ez da markatzaile fiduziariorak posizio zehatza jakin behar. Markatzaile askorekin ere Zhenglong et al. [373]-ek posearen arazoa aztertzen dute, beherantz orientatutako kamera monokular batekin eta AprilTag-ekin. Irudi asko ateratzen dira eta bisio algoritmo batekin AprilTag-en arteko posizio erlatiboa lortzen da, ondoren Kalman iragazki baten bidez ibilgailuaren posea sendotasunez kalkulatu.

GPS gabeko inguruneetan Tsoukalas et al. [374]-ek ere jardun dira ikerketa egiten. Kasu honetan markatzaile fiduziariorak ibilgailuaren gainean doaz kokatuta eta ikusizko jarraipena egiteko mekanismo bat erabiltzen dute, obturazio kamera global batekin. Horrela ibilgailuaren pose erlatiboa

lortzeko gai dira. Ibilgailuaren gainean markatzailea kokatzearena Pickem et al. [375]-en lanean ere ikus daiteke, baina hauetan posea ibilgailu ugariren segurtasun sistema osatzeko aztertzen da.

Aldibereko mapatze eta lokalizazioa ere lor daiteke markatzaile fiduziarioen bidez. VIO metodoarekin egiten dute lokalizazioa Bertoni et al. [376]-ek, markatzaile fiduziaroetan oinarrituz. Metodo honek barruko inguruneetako nabigazioa ahalbidetzen du, irudien bidez posearen estimazio jarraia planteatzen bait da. Markatzaileekin mapa bat diseinatzen dute, emaitza sendo eta heterogeneoak sortuz. Gainera, leku estuetan kokatutako erreferentzia-puntuak informazio galera oso txikia daukate eta erabilitako lau neurritako markatzaileak koherentzia ematen diote ibilgailuak hauetara duen distantziari. Errorea 0,11 m baino gutxiagoan mantentzen da, emaitza onak lortuz. Muñoz-Salinas et al. [377]-ek diotenaren arabera, markatzaile fiduziaroen erabilera markatzaile bakar batera mugatzen dira maiz, edo, gehienez ere, alde aurretik pose ezaguna duten hauetako multzo txiki batera. Markatzaile fiduziario multzo handi batetik abiatutako SLAM-a oraindik ez da oso tratatua dagoen arazo bat. Hau da zehazki beraien lanean ikertzen dena, hala nola, markatzaile asko erabiliz aldi bereko mapatze eta lokalizazioa ebazteko metodo bat. Lehendabizi, markatzaileen pose erlatiboen multzo bat sortzen da, binaka elkartzen direnak, eta horietatik hasierako poseen grafiko bat lortzen da. Horrela, eta ziklikoki erroreak zuzenduz, SLAM-a lortzeko metodo eraginkor bat lortzen da.

Aurreko lan honetako emaitza onak ikusiz, Muñoz-Salinas et al. [378]-ek beste bigarren lan bat aurkezten dute. Muga batzuk antzematen dira lehenengo lanean: alde batetik, paperean inprimatutako marka multzoak askatasunez daude banatuta inguruneetik eta marken posizioa kalkulatzeko, gutxienez bi patroik agertu behar dira irudi bakoitzean. Bestetik, metodoa ez da inkrementala beraz mapa handitu behar badin bada, hasieratik egin behar da prozesu osoa. Azkenik, metodoa ezin da erabili baliabide informatiko mugatuak dituen denbora errealeko sistemetan, beraz oso zaila da ibilgailu autonomoetan aplikatzea. Arazo hauei aurre egiteko gauzatzen da bigarren ikerketa hau hain zuzen ere.

Kamera bakar batekin denbora errealean SLAM-a egiten da Lim et al. [379]-en lanean. Kalman iragazki hedatuaz baliatzen dira kasu honetan robotaren posea eta erreferentzia puntuen posizioa estimatzeko. Ikuspegiaren nukleoa, ingurunean dauden markatzaile fiduziaroekin *online* mapa bat sortzea da, probabilitate esparru baten barruan. Guztia bilduz, TagSLAM du izena AprilTag-ekin aldibereko mapatze eta lokalizazioa egiteko interfaze sendo eta malguak. Honek abstrakzio erraz batzuk aprobeztatzen ditu, hau da, gorputzak, etiketak eta kamerak, grafo faktorialak optimizatzeko. Horrela, interfaze berak, erabateko SLAM-a, gainjarri gabeko bistak dituzten kameraren kalibrazio estrintsekoa, zoruaren ikusizko lokalizazioa, odometriaren begiztaren itxiera eta pose zenbatespena egiteko aukera ematen du (ikusitoki Pfrommer et al. [380]).

4. Beste Teknika Motak Gidatze Autonomoan

Orain arte ikusitako ibilgailu autonomoen sistemek ikaskuntza automatikoaren oinarriean daude garatuak baina hauek ez dira bakarrak. Bibliografian zehar beste algoritmo mota batzuk aurki daitezke emaitza oso probetxugarriak dituztenak eta askotan, era errazago batean eraman dezakete emaitza on batera. Teknika hauetan ere lokalizazioa, mapatzea, ibilbide plangintza eta nabigazio metodoak aurki daitezke, eta ibilgailu autonomoetan hauen aplikazioa.

Antzeman ahal izan den bezala, robot mugikorrenzako ezinbesteko sistemetako bat **posizioa** determinatzen duena da. Arazo horri aurre egiteko modurik tradizionalenetako bat gurpilen odometria erabiltzea da, Kilic et al. [381]-ek aztertu duten bezala. Nabigazio sistema inertzial batekin konbinatzen dute non, neurriak *encoder* baten bidez hartzen diren. Hala ere, teknika honen erabilerarekin kontuan izan behar da gurpiletan gertatu ohi den lerradura, kalkulatzeko datu gehigarri bat izanez eta lokalizazioan zehaztasuna galduz. Honi aurre egiteko, Chen [382]-ek Kalman iragazkiaren erabilera proposatzen du, parametro horien estimazioa egiteko baliabide oso aproposa bait da.

Beste alde batetik, Kalman iragazkia oso baliagarria da parekatze teknikekin batera erabiltzeko. Cho et al. [383]-ek egindako azterketaren arabera, bi parekatze metodo erabili daitezke iragazki horrekin konbinatuta. Alde batetik, metodo geometrikoa eta bestetik, ICP (*Iterative Closest Point*) algoritmoaren puntutik linearako parekatzean oinarritutako metodoa. Kasu honetan, metodo geometrikoa posea aurrerako aplikatzen da eta ICP-a berriz, zenbatetsitako posizioa zuzentzeko. Honekin, AV-a dagoen ingurunearen aurretiko informazioaren beharra dago. Era berean, metodo geometrikoen erabilera hutsa egin daiteke, Shamsfakhr et al. [384]-ek demostratzen duten erara hauek garatutako algoritmoan. Bertan, laser erreal baten datuen eta simulatutako laser baten datuen segmentazioa egiten da eta hauetan oinarritutako patroia erregistro bat burutzen da. Bietatik lortutako puntu kritikoak hartuz eta hauen arteko ezberdintasunetan erreparatuz, algoritmo sendo eta konputazionalki eraginkor bat lortzen dute, denbora errealean posea zehazten duena.

Partikula iragazki bat integratuz, Tao et al. [385]-ek robot mugikorren lokalizazio globalerako metodo bat proposatzen dute, irrati frekuentziaren identifikazioan (*RFID-Radio Frequency Identification*) oinarrituta. Señale hauen bi informazio mota konbinatzen dituzte, hala nola, fasearen arteko ezberdintasuna eta irakurgarritasuna. Horrela, zentimetrotako zehaztasunarekin kalkula daiteke ibilgailu autonomo baten posizioa. Beste alde batetik, eremu-lerro magnetikoak tartean dauden AV-en aplikazioetan, eremu RFID lokalizazio metodo aktiboa era berean erabili daiteke. Suspentsio erroreen posizio metodoan oinarrituz, prezisio handiagoa lortzen da eta fluktuazio tasa-txikiagoa (ikus Lu et al. [386]).

Hurbileneke puntu iteratiboa metodo errepikari bat izaten jarraitzen duela ikus daiteke bibliografian zehar, lokalizazioaren arazoa ebazteko orduan. Gainera, LiDAR sentsoreek kanpo-azpiegituraren beharrik ez dutela jakinda, ICP ponderatu eta 2D-ko puntu geruza batekin batera, norabide bateko batez-besteko errorea kalkula daiteke lokalizazioa definituz (ikus Naus et al. [387]). Wang et al. [388]-en proposamenean, ICP iterazioak murrizten dituen arkitektura bat aurkezten da LiDAR sentsorearen batera, konputazio kostearen esfortzua gutxituz ere. Barrualdeko lokalizatze metodo honetan, puntu-lainoa lehengo prozesatzen da izkinak eta lerroen ezaugarriak lortzeko puntuen erregistroa aplikatu aurretik, denbora errealean erabili daitekeen metodo bat lortuz.

Era berean, iterazioak murriztu ordez, 3D puntu-multzo gehiago sortzeko aukera ikertzen da Senin et al. [389]-en lanean, dentsitate edo disposizio espazial ezberdinak dituzten multzoak adibidez. Ingurune geometriari oinarritzen dira eta sendotasuna eta errendimendua hobetzea lortzen dute. LiDAR-arekin batera beste motatako sentsoere batzuk erabiltzen badira, hala nola, INS eta GPS-a, Gao et al. [390]-ek barrualde eta kanpoaldeko lokalizazioa egin daitekeela azaltzen dute. Beraz, biak konbinatzen dituen algoritmo hibrido baten erabilerarekin ICP puntuen korrespondentzia aurki daiteke ibilgailu autonomoetan aplikatzeko aukeraz. Hala ere, Kim et al. [391]-ek adierazten dute ICP-aren erabilera hutsarekin ez dela altuera ezberdinak dituzten inguruak ezberdintzen. Honi aurre egiteko, ibilgailuen baldintza pean, hurbileneko puntu iteratiboaren algoritmoak korrespondentzia sendoak aurkitzeko metodo bat proposatze dute, KITTI datu-multzoaren informazioaz baliatuz.

Kanpoaldeko inguruneetan erabiltzeko GPS-a da sentsoere egokiena baina hau ez da industriari kasua. Antena honen aplikazioa ekidin nahi izan ezker, trilaterazioaren bidez egin daiteke AGV-aren lokalizazioaren hurbilketa, ibilgailuarekiko distantzia estimatzen duten seinale batzuen bidez. Sadeghi Bigham et al. [392]-ek hau planteatzen dute azkartasuna eta efizientzia altua behar duten algoritmoetarako, poligono ortogonalean oinarrituz. Horrela, $n/2$ erreferentzia-puntuak nahikoak dira posea lortzeko eta n -gono ortogonal baten trilaterazioa ere gartzeko aukera dago. Gainera, erreferentzia-puntuen erabilera, jakinaenez, bisioarekin erabili daiteke Yap et al. [393]-ek egiten duten bezala. Hauek aukeratutako geometria zirkularra da eta AGV-aren eta erreferentzia-puntuen distantzia zirkuluen intersekzioaren kontzeptuan oinarritzen dute.

Bayesiar iragazki batean oinarritutako algoritmo bat erabiltzeak ere emaitza onak eman ditu robot mugikorren lokalizazioan. Horretaz gain, inguruneari buruzko alde aurretiko informazioaren beharra izatea saihesten du, Gentner et al. [394]-ek aipatzen duten bezala. Iragazkien erabilerarekin jarraituz, partikula iragazkia da, halaber, lokalizazioaren arazoa konpontzen saiatzeko aplikatu den beste metodo bat. Iragazki honek gainera, ingurune bateko egoera zenbatesteko aukera ematen du. Kasu honetan, zehaztasuna partikula-kopuruarekin erlazio zuzena dauka, baina kontuan hartu behar da partikula kopurua handitzeak kostu konputazionala zigortzen duela (ikus Yang et al. [395]).

Aldibereko mapatze eta lokalizazioan ere ICP-a erabilia izan da. Clemens et al. [396]-ek algoritmo hau deribatzen dute ibilgailu autonomoetan erabili daitezkeen sentsoere guztiak kontsideratuz. Ausazko puntuen mapak eraikitzen dituzte eta SLAM-a egiteko algoritmo bat garatzea lortzen dute, erailtzen diren sentsoerekiko independentea dena. Modu honetan, sentsoeren zarata ziurgabetasuna neurtzeko era batean bihurtzen da. Arrazoi honegatik, askotan, SLAM-ean sortutako gelaxken optimizazio metodo bat beharrezkoa suertatzen da. Optimizazio honen ondorioz, Zhang et al. [397]-ek, itxura antzekoa duten guneak ezberdintzeko zailtasun bat sortzen dela demostratzen dute. Horregatik, erreferentzia modu globalean hartzea proposatzen dute. Hauen lanean, partikula iragazkiaren eta PSO algoritmoaren ikuspegi hibrido bat aurkezten da posizionamendu lanetan zehaztasun handia lortuz.

Ibilbidea kalkulatzeko lanetan PSO (*Particle Swarm Optimization*) algoritmoa da konbentzionalki erabili den teknika. PSO-aren inertzia pisuak bariazio lineal batekin hobetuz gero, algoritmoa minimo lokal batean ez erortzea lortzen da, eta, hala, konbergentzia abiadura handiagoa lortzen da Fei et al. [398]-ek frogatzen duten bezala. Modu honetan, ibilbide optimo baten kalkulua ahalbidetzen da edozein inguruneetan, A* algoritmoak lortzen dituen bide-luzera hobeagoak eskuratuz 2 dimentsiotako mapetan. Besteak beste, Liu et al. [399]-ek, A* algoritmoa mapak modelatzeko metodo gisa erabili daitekeela azaltzen dute, ondoren ibilbidea estimatzeko. Kasu askotan, helmugarainoko distantzia optimoa bakarrik kontuan izatea ez da nahikoa beraz, oso baliagarria izaten da puntu horretaraino denbora laburrena ere ebaluatzea. Hau kontsideratuz eta ibilgailuaren

atribuetan oinarrituz Cheg et al. [400]-ek A* algoritmoak bi ibilbideak kalkulatu ditzakeela demostratzen dute.

Beste alde batetik, Silva Costa et al. [401]-ek A* algoritmoa eta RRT (*Rapidly-Exploring Random Tree*) algoritmoaren konparaketa bat aurkezten dute, ingurune dinamikoen aplikazioen kasuetarako. Sortzen duten ibilbidearen luzera, konputazio kostea edota ibilbidearen segurtasuna dira kontuan izan dituzten aldagaietako batzuk. A*-ak errendimendu global altuagoa duela ondorioztatu dute, baina segurtasunaren aldetik exkaxa dela ere. RRT algoritmoaren beraren modifikazio bat aurkezten da Wang et al. [402]-en lanean. Ibilbide plangintzarako diagrama berri bat proposatzen dute, zeinetan bidea bat-batean lortzeko gai diren. Wei et al. [403]-ek beste diseinu bat argitaratzen dute, Smoothly RRT izenpean. Honetan, optimizazio estrategia ibilbidearen kurba maximoan oinarritzen da, esplorazio abiadura handiagoa lortuz.

Era berean, *Wavefront* algoritmoa ibilbidea kalkulatzeko beste metodo errekursogile bat da, fronte hurbileneko puntuak lortzeko erabilia. Hala, AGV-aren mugimenduen eskakizunetan oinarritzat hartuta, puntu hoberena aukeratu daiteke. Beraz, *Wavefront* algoritmoak ibilbide osagarriak ere bila ditzake beharrezkoa izan ezker, Tang et al. [404]-ek adierazten duten bezala. *Wavefront* algoritmo orokorraren erabilera eztabaidatua izan den gai bat da. Helburu-puntu multzo anitzak duten kasuetan, maila anitzeko laukien kostuen eta oztopoen inguruko hedatze geometrikoen arteko konbinazioa egin daiteke. Informazio guzti honekin, ibilbidea optimizatzeko aukera dago, bide seguru eta lehunak lortuz (ikusi Wu et al. [405]).

Nabigazioaren kasuan, bibliografian landuago dagoen teknika DWA-rena (*Dynamic Window Approach*) da. Irakaskuntza-ikaskuntza teknikan oinarritutako metodo honen hibridazioa egiterakoan, posible da robot mugikorrek emandako puntu objektibo batera heltzea, Kashyap et al [406]-ek demostratzen duten bezala. Horretaz gain, geldialdirik egin gabe oztopoak gainditzeko gaitasuna ematen dio ibilgailuari. Bestalde, mugimenduak denbora errealean planifikatzeko metodo batekin ere egin daiteke konbinazioa, AGV-aren nabigazio abiadura handituz (ikusi Brock et al. [407]). Konbinazioekin jarraituz, Dijkstra algoritmoa eta DWA batzen badira, nahi den posizioa SLAM sistema batek emandako informazioaren bidez lortzeko aukera dago Liu et al. [408]-ek frogatzen duten erara. Honetaz guztiaz gain, oso garrantzitsua da ibilgailuen ezaugarri dinamikoko kontuan izatea nabigazioaren kontrola egiten denean, ibilgailu autonomoak berak abiadurari eta azelerazioari ezartzen dizkion murrizketen ondorioz Fox et al. [409]-ek azpimarratzen duten bezala.

Wang et al. [410]-ek egindako azterketan ikus daitekeenez, nabigazioaren problema bi geruzatan banatu daiteke. Alde batetik, bidearen jarraipen egokia gauzatzea eragiten duten erabakien hartzea eta bestetik, ibilbidearen jarraipenaren exekuzio propioa. Bigarren kapa honetan, posible da indar birtualeko eremuaren (VFF-*Virtual Force Field*) algoritmoa erabiltzea, objektuen detekzioa ahalbidetzen duena. Era berean, indar potentzialaren eremu (PFF-*Potential Force Field*) algoritmoarekin konbinazio bat egin daiteke, nabigatzeko bitarteko bideragarri bat eraikiz (ikusi Burgos et al. [411]). Ibilbideen jarraipena egiteko aplikatutako beste algoritmoetako bat eremu bektorialaren histograma (VFH-*Vector Field Histogram*) da, Borenstein et al. [412]-ek frogatzen duten bezala. Hauek histogramaren laukiak erabiltzen dituzte eredu gisa inguruneko mapa bat sortzeko, modu honetan, AGV-aren kontrol aginduak lortzeko aukera izanez. Gainera, partikula iragazki bat VFH-rekin konbinatzen bada oztopoak ekiditeko sistema bat lortzen da (ikusi Wang [413]). Azken algoritmo honen aldaketa bat existitzen da ibiltzeko eremu histograma (TFH-*Traversability Field Histogram*) izena duena, ibilgailuaren uneko abiaduraren independentea dena Ye [414]-k antzeman zuen bezala. Horrela, urrutiko objektuek ibilbidearen jarraipen optimoari kalte egitea ekidin daiteke.

Ikuspegi ezberdin batetik, nabigazioaren arazori irtenbidea emateko alboko kontrola erabiltzea beste metodo bat da, ibilbidearen punturik hurbilena identifikatuz AGV-a erreferentziaz hartuz. Hau egin dezakeen algoritmoetako bat Stanley metodoa da, zeinetan robot mugikorraren etorkizuneko egoerak modu diskretuan aurretateko eredu bat eraikitzen den. Abdelmoniem et al. [415]-en ikerketan, aipatutako algoritmoaren eta LiDAR sentsorearen konbinazio bat erabiltzen da. Horrela, ibilbideen planifikatzaile lokal bat sortzea lortzen dute, talkarik gabeko nabigazioa osatu dezakeena.

ATALBURUA 3

BEHARREN HAUSNARKETA

*Gutxi inporta du zein heltzen den lehenengo ideia batera,
esanguratsua dena ideia hori noraino heldu daitekeen da.*

SOPHIE GERMAIN

1. Ibilgailu Autonomoetako Erronkak

Ibilgailu autonomoi buruzko bibliografiaren analisia egin ostean, zenbait alderditan gabeziak antzematen jarraitzen da. Nahiz eta, areagotzen doan diru eta denbora inbertsio handia egin izana ibilgailu autonomoen teknologiei dagokienez, badaude zenbait atal oraindik zailtasunak aurkitzen dituztenak duten gaitasun guztia demostratzeko. Orokorrean, antzeman daiteke ibilgailu autonomoen sistemek konplexutasun altua dutela. Gainera, arazo askori aurre egiteko erabaki desberdinak hartzeko aukera dago. Honek, problema baten emaitza, beste baten jatorrian bilakatzea suposatzen du. AV baten zenbait ataletan hobekuntza nabarmenak detektatu arren, oro har, ibilgailuaren sistemaren errendimendu globalaren mende daudela ere antzeman da.

AV-bat funtzionamenduan jartzerako orduan badaude zenbait erabaki giza pentsamenduaren antzera hartu behar direnak, batez ere segurtasunaren alorrean. Beraz, ibilgailu autonomoetarako diseinatzen den mugimendu planifikatzailea eraginkorra izan behar. Zoritxarrez, eraginkorrak diren kontrolatzaileak energia kontsumo altu batekin lotura estua daukate. Gainera, ikusienez, kontrolatzaile sinpleek sendotasun baxuagoa daukate, baina energia gehiago aurrezten dute. Hala ere, azken hauek, mugimenduaren planifikazio ikuspegi oso zehatz baten beharra dute. Ondorioz, metrika kontraesankor hauen arteko oreka bat bilatzen duen sistema adimentsuen garapena behar da, irtenbiderik optimoena bilatuko duena.

Industria 4.0 etorrerarekin eta enpresen digitalizazioaren areagotzearekin, ibilgailu autonomoen nabigazio inguruneen konplexutasuna ere igotzen da. Inguruneak geroz eta dinamikoagoak bihurtzen ari direla argi dago eta hau, lanetan erreflexatzen da, gune mota hauek tratatzen duten azken urteotako ikerketen bolumena ikusiz. Gainera, iluminazio eta kolore aldaketak ere ohikoagoak dira eta hauek tratatzeko bisioa aplikatzen ari da nagusiki. Honek, zenbait elementuen detekzioaren zehaztasuna kaltetzen du, posizioan edota SLAM-ean adibidez. Ibilbideen jarraipena zailtzen da ere eta ondorioz, AV-ekin erlazioa duten atazen exekuzioa korapilatsua bilakatzen da.

Inguruneen eta sistema propioen konplexutasunaren igoerak, lan egiteko behar diren datu-bolumenaren handitzea dakar. Gainera, antzeman ahal izanenez, gero eta eskakizun handiagoa dago datu horiekin denbora errealean lan egiteko. Jakina da AV-ek detekzio gailu ezberdinak erabiltzen dituztela ingurua ezagutzeko. Hauek, datuak etengabe jasotzen dituzte, datu-bolumen handiak sortuz. Detekzio gailuen hobetzeak ere jasotako datuen kalitatea ere hobetzen du, hauen pisua handituz. Beraz, datu kopuru handi eta pisutsu horiek denbora errealean prozesatzeak erronka bat suposatzen du, zehaztasuna eta kostu konputazionala kontuan hartu behar bait dira. Esateko dago, sentsoreen hobetzeak energia kontsumo altuagoa ere inplikatu duela.

Datu-multzoetan jatorri eta ezaugarri ezberdinak dituzten datuak biltzen dira. Honek *big data* kontzeptuari pausu ematen diona da. Aurrez egindako ikerketetan nabaritu da ibilgailu autonomo batentzako ezinezkoa dela jasotako datu guztiak prozesatzea. Honek, datuei lehentasuna emateko mekanismo baten beharra suposatzen du. Gainera, ikusi da baita ere, nabigaziorako behar diren datuen eduki garrantzitsuak iragazi behar direla, ondoren prozesatu ahal izateko eta beharrezkoak ez diren datuak baztertzeko. Ataza hau egiten duten sistemek beraz, ingurune desberdinetan jasotako datuen aniztasuna mantendu behar dute, AV-ak malgutasuna mantentzeko.

Detekzio gailuetara itzuliz, aplikazio errealetan orokorrean sentsoreen konbinaketa egiteko joera nabaritu da, oso zaila bait da gailu bakar batekin espero den errendimendu eta zehaztasuna izatea. Ondorioz, eta aipatu den bezala, dispositiboek jasotako datuak prozesatzeko algoritmo ugari daude, hauen arteko erabakiak hartzen laguntzen dutenak. Antzeman daitekeena ere, ibilgailu autonomoetan bi sentsore mota erabiltzen direla nagusiki da. Alde batetik “ingurunearen pertzepzio” multzoan sartzen diren sentsoreak daude, inguruko objektuak detektatzeko helburua dutenak. Beste aldetik, “lokalizazio” multzoko sentsoreak, izenak esaten duen bezala AV-aren posizioa determinatzeko xedea dutenak. Hauek jasotako datuak erabiltzen dituzten algoritmoak ere bi multzotan bildu daitezke. Bata, ikaskuntza automatikoko metodoak eta bestea, iragazkiak erabiltzen dituzten sistemak. Bai sentsoreen fusioa bai algoritmoena proposatzen dituzten eredu ugari daude, batez ere zehaztasunean bideratuta. Honek, aplikazioaren bideragarritasuna alde batera uzten du, gehienetan simulazio lanetan geratzen bait dira. AV-ekin lan egiten duten ikerlariak denbora errealeko komunikazio eta exekuzioan jartzen dute arreta batez ere. Baita ere ikaskuntzan oinarritutako kontrol sistemetan. Hautematen da hala ere, lan falta dagoela sentsore eta algoritmoen fusio arin, sendo eta eraginkor baten bilaketan.

Ohartarazi ahal izan den bezala, malgutasunarekin batera, ibilgailu autonomoek izan behar duten bezte ezaugarrietako bi, sendotasuna eta moldagarritasuna dira. Ikerketa asko aurki daitezke helburu hauen atzetik, eta orokorrean, jasotako datuek duten garrantzia hauek determinatzeko. Argi dago, ingurune jakin batean datuak jasotzea eta prozesatzea ez duela zailtasun handirik. Ondorioz, asko sinplifikatu da datu-base publikoen sortzea. Datu horiek baina, ingurune espezifiko batekoak dira eta adimen artifizialaren ohiko teknikak ingurune jakin horietako datuekin entrenatzeko joera dago. Honek, gehiegizko orokortze bat suposatzen du, datuak hainbat aplikaziotarako baliagarriak ez direlako eta hau ez delako aztertzen. Beraz diseinatzen den sistema, nahiz eta bere inguruneari dagozkion datuekin ez entrenatu, bere kokapenera egokitze bezain sendoa izan behar du.

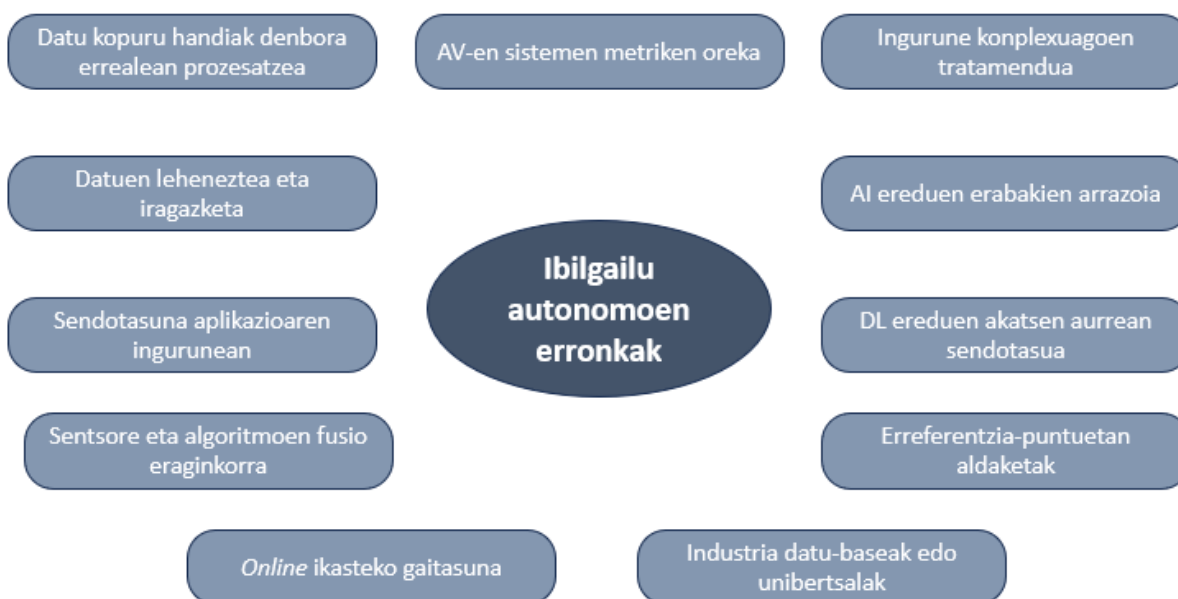
Inguruneen ildo beretik, AV-en erronka handienetarikoa bat hainbat ingurune eredu adaptagarri batekin maneiatu ahal izatea da. Hau lortzerakoan, industria batetik besterako aldaketa ez zen arazo bat suposatuko robot mugikorrarentzat eta guztiz malgua izango litzateke. Honetan ere, argiztapenak eragin handia izan ohi du. Ikerlari batzuek *online* ikaskuntza teknikak erabiltzen hasi dira ataza hau aurrera eramateko nahian. Modu honetan, eredu sakon bat iteratiboki doitzeko ideiarekin lan egiten dute, non parametroak eguneratzen doaz ingurune berriari egokituz.

Orain arte aurki daitezkeen ikerketa gehienetan gainera, detekzio erduek, objektu aldaezinak erabiltzen dituzte. Industriako inguruneetan, “eszenatokia” aldatzeko posibilitateaz gain, erreferentzia-puntuak kausa anitzengatik aldaketak ere jasan dezakete. Gainera, AV-ak ateratako irudiak berriz dimentsionatzen dira sare neuronalean sartu aurretik eta horrek, zehaztasun galera suposatzen du askotan. Prezisio galera hori ere robot mugikorren abiaduraren igoerarekin dator, beraz markatzaile guztien detekzio egokia ataza konplexu batean bilakatzen da

Beste azterketa ugari *deep learning* eredu ahultasunak aurkitzea dute helburutzat. Ahultasun hauek, naiz eta kodigoan antzeman ezin diren, erabakia okerrak hartzera eramaten dituzte, hala nola, klasifikazio desegoki bat egitera. Akats hauek erronka handi bat suposatzen dute AV-entzako eta, naiz eta lanen gehikuntza bat egon hauek detektatzeko estrategien inguruan oraindik oso ahul dauden teknikak dira, bai diseinu bai eraginkortasun aldetik sendo bezala kontsideratzeko.

Badaude ere zenbait ikerketa adimen artifizialeko ereduak hartutako erabakiak azaltzeko premia azpimarratzen dutenak. Adimen artifizialaren azalgarriaren (XAI-*Explainable Artificial Intelligence*) [416] teknikekin lan eginda, ereduaren barnekotasuna eta hauek ikasitakoa aztertu daiteke, horrela hauen ezagutza handituz. Tresna honetaz baliatuz, okerreko erabakien miaketa egiteko aukeraren beharra dago, ereduaren funtzionamendua perfektionatzeko. Beste alde batetik, entrenamendu datu oso desorekatu edo urriek ikaskuntza sakoneko erduei eragiten ez dietela bermatzeko beharra dago. Horrek, errendimendu orokorraren hobekuntza finkatuko du, ibilgailuaren funtzionamendu fidagarriago bat lortuz. Ereduak ikasi dutena ulertzeak hauen errendimenduan sortzen diren mugak ulertzeko tresna bezala erabili daiteke. Egon daitezkeen akats fokua aurkitu daitezke, hauek sortzeko ereduak hartu zituen erabakiak aztertzeko gaitasuna emanez. Ereduen azalpenean aurrerapenak egiten ez badira, ibilgailu autonomoen funtzionaltasunak modelatzeko ahalmena geldituta geratzen da.

Datu-baseetan atentzioa jarriz, orokorrean ibilgailu autonomoentzako aurki daitezkeen kanpoaldeko inguruneetakoak dira, bibliografian zehar industrien barneko datu-multzoak topatzeko zailtasunak egonda. Ez bakarrik ingurune industrialetako datuak, baizik eta barne-inguruneetako informazioa falta da lan egin ahal izateko. Hauek sortzeko erronka, zer mapatu behar eta ahal den jakitean dago, inguru pribatuak bait dira eta informazio konfidentzialtzat har daitezkeelako informazio hau. Orokorrean, erreferentzia unibertsala duten datu-baseen behar bat dago ibilgailu eta ingurune guztiei aplikagarria dena. Gainera honek, ereduak ebaluatzeko erraztasuna suposatuko zuen, ikerketa guztien arteko konparaketak egiteko gaitasuna emanez.



Irudia 23: Ibilgailu autonomoen inguruan gaur egun dauden erronkak

Irudia 23-an, aipatutako kontzeptu guztiak biltzen dira era eskematikoan, ikuspegi orokorra izan ahal izateko. Guzti hau ikusi ostean ondorioztatu daitezkeena da, AV-ak oraindik teknologia hasiberritzat hartu daitezkeela da eta ikerketa ildo ugari jarraitu daitezkeela hauen probetxu guztia lortzeko.

1.1. Lan honetan jorratutako gabeziak

Behin erronka guztiak aztertuta, hutsune gehiago antzematen dira zenbait ataletan, eta horiek izango dira lan honetan aztergai izango direnak. Lehenik eta behin, *Reinforcement Learning*-arekin alderatuta adibidez, *Deep Learning*-etik dauden lan kopurua da arreta deitzen duen puntu bat. Bigarren teknika hau oso baliagarria dela demostratu da eta hori, aplikatu den esparru guztietan du erreflexua. Hau, ikaskuntza sakona dagoeneko ustiatuta dagoenaren adierazlea da. Beraz, lan hau errefortzu bidezko ikaskuntzaren ikerlanean jartzen du arreta. Eskura dauden lan zientifikoan azterketa zehatzago bat eginez, ez da uste errefortzu bidezko ikaskuntza teknika desegokia denik; aitzitik, gutxi ustiatuta dagoenez, zaila da emaitza optimoak beste tekniken erraztasun berarekin lortzea.

Gehiago sakonduz, hiru atal dira Reinforcement Learning-aren barruan arrazoiketa teoriko landuagoak behar dituztenak eta ondorioz, programazioari dagokionez, optimizatzeko zailak direnak ulermen faltagatik. Alde batetik, *reward function* edo sari funtzio bezala ezaguna dena dago. Aplikazioaren baitan dago, beraz kasuan-kasuko funtzio bat da. Arazoa da ez dagoela hau era egokian planteatzeko irizpiderik beraz, ezin da jakin zehaztasunez aproposa den ala ez. Honen zailtasuna batez ere kontuan izan daitezkeen aldagaien kopuruan dago. Adibidez, ibilgailu autonomoen kasuan, funtzio honek barneratzen dituen parametro ohikoenak ibilgailuaren posea, abiadura eta oztupoekiko distantzia dira. Hiru hauek kontsideratuz soilik, sari funtzioa ez da batere optimoa. Esaterako, simulazio denbora, helburua lortzeko behar izan duen denbora edota iterazio kopuruak ere garrantzitsuak dira agente bat doitzeko orduan, baina aplikazio batean parte hartuko duen guztia integratu nahi izan ezker, infinitu aldagai izan beharko lirateke kontuan. Esan bezala, ez dago "patroirik" *reward*-a era egokian definitzen dela ziurtatzen duena. Hau ezinezkoa kontsideratzen da, aipatu bezala aplikazioaren baitakoa bait da eta infinitu parametro barneratuko zituelako. Ondorioz, errefortzu bidezko ikaskuntzaren optimizazioan hutsune bat suposatzen du, emaitzak kaltetuz.

Beste bi atalak, antzezle eta kritikoarekin dute zerikusia, *Reinforcement Learning*-ak ez duelako beste biderik optimizazio bat lortzeko. Hauen konplexutasunagatik, orokorrean, sare neuronalak erabiltzen dira bai kritikoaren funtzioa egiteko, bai antzezlearena. Nahiz eta sari funtzio simple bat izan, hauen lana da agentea doitzea era egokienean. Kritikoak saria zenbatetsi behar du eta horren arabera antzezleak akzio aproposena kalkulatu behar du. Hau horrela izanda, sariaren egokitasun-maila garrantzia galtzen du bi sare neuronal hauen aurrean. Gehiago espezifikatuz, lan honetan, antzezlean jartzen da garrantzi gehiena, azken finean kritikoak ebaluazio bat egin behar duelako eta ez da kontsideratzen sare hori oso konplexua izan behar denik emaitza onak lortzeko. Gainera, erraztasun handiarekin ordezkatu daiteke funtzio batengatik. Ondorioz, kontsideratzen da *Reinforcement Learning*-aren garrantzi handiena antzezlearen sarearen definizioan dagoela. Hau optimizatzeko modu bat diseinatzeak, agenteen hobekuntzan eragin handia izango duela hausnartzen da, emaitza egokiak lortuz. Beraz, lan honen pisu handiena zeregin honetan egongo da. Kasu honetan, errefortzu bidezko ikasketa ibilbidearen plangintza aurrera eramateko eta nabigazio kontsignak lortzeko aplikatuko da. Helburu hau betetzeko era analitikoan eta eboluzio bidezko irtenbideetan jarriko da arreta.

Ibilgailu autonomoen sistema osatzeko SLAM sistema baten beharra ere dago. Bibliografian zehar antzeman daiteke orokorrean sentsoreen konbinaketa bidez egiten dela. Honek erraztasuna ematen du neurketetan eta ingurunearen miaketan, baina AV-aren sistema ere garestitzen du. Alabaina, alderaketa bat eginez, gutxi dira sistema monokular soilak planteatzen dutenak. Egia da kamera bakar bat sentsore bezala izateak gabezia asko mahaigaineratzen dituela, esaterako, distantzien neurketa, baina AprilTag-en erabilerak hauek gainditu ditzake. Etiketa hauek merkeak dira

eta ez dute instalazio konplexurik. Gainera zehaztasun handia eman dezakete irakurketetan [417], mapaketa eta lokalizazioa egin ahal izanez. *Reinforcement Learning*-algoritmoan sare neuronalen erabilerak kostu konputazionala igoko duenez, SLAM-a modu analitikoan planteatzen da, ondorengo lanetan guztia barneratu nahi baldin bada arazo bat ez suposatzeko.

Kamera bakar baten erabilerarekin distantzia kalkulatzeko arazo bat suposatzen duela ikusi da ere, goiko parrafoan aipatu den bezala. Lan honetan, teknika guztiei heltzeko, *Deep Learning*-aren bidez ataza hau egingo duen sare neuronal bat ere planteatzen da. Honen helburua sistemari segurtasun atal bat gehitzea da. Horrela, ibilgailu eta oztopoen arteko neurriak detektatzeko aukera dago geldialdi bat egiteko beharra egon ezker. Aukera dago atal hau *Transfer Learning*-aren bidez osorik egiteko baina sare neuronal berri bat egitea erabaki da, sinpletasunagatik egokiagoa izateko, lehen esan bezala, ondorengo lanetan guztia bildu nahi baldin bada. Hemen jokia batez ere datuekin dago, industria esparruetan ezin izango direlako, kasu askotan, irudi asko lortu entrenamendu egoki bat egiteko beraz, datu gehitze teknikak aplikatu beharko dira. Nahiz eta distantziak kalkulatu dituen sarearen diseinua egin, *Transfer Learning*-a bai aplikatuko da oztopoen detekzioan. Biak batuz beraz bilatzen hari den segurtasun sistema lortzeko aukera dago.

ATALBURUA 4

GARATUTAKO ALGORITMOAK

*Ez utzi inoiz besteen irudimen mugatuak
zu mugatzen.*

MAE JEMISON

1. Erabilitako Ibilgailu Autonomoa

Vitoria-Gasteizko ingeniari-eta eskola AMR eredu bat aurkitu daiteke, ikerketak egiteko erabili ahal dena. Beraz, eredu honetarako garatzen da tesia hain zuzen ere, probak plataforma honen gainean egin daitezkeelako eta, hala, funtzionamendua balioztatu hurrengo lanetan. Ikusi den bezala, ikertzaileek lan asko dute oraindik ibilgailu autonomoei dagozkien sistemak perfektionatzeko eta hau, bai sektore pribatuari bai publikoari dagokio. Arazoekin aurrera egiteko ideiarekin nahi izan zuen unibertsitateak bere AMR propioa izatea eta Argolabe Ingeniería S.L.-ren esker lortu zen plataforma hau 2018-an, hauek izan bait ziren diseinu mekanikoan parte hartu zutenak (ikusi **Irudia 24**).



Irudia 24: Unibertsitatean eskuragarri dagoen ibilgailu autonomoa 2

Plataforma hau ikasleen hainbat lan burutzeko prozesuan oso baliagarria izan da, bai gradu-mailan, bai master-mailan eta, jakina, doktorego-mailan. Modu honetan, ibilgailu autonomo industrialen ezagutza unibertsitate arlora eraman da, honen garapenerako ataza ezberdinak landuz. Robot mugikorren ezaugarriak ikasteko aukera emateaz gain, industriara hurbildu ditu ikasleak, barneko nabigazioa landuz, programazioa egiteko baliabide izanez. Gainera arazo ezberdinak planteatzeko aukera ematen du eta honek, ideia ezberdinei pausu eman die, sentzore ezberdinen erabilerarekin adibidez.

1.1.AMR-aren konfigurazioa

Unibertsitateko robot mugikorrak ez du konposizio oso konplexua eta hori, abantaila bat da, moldaketa ezberdinei pausu ematen bait dio. Ibilgailu autonomo honek duen barne-konfigurazioa ondorengo irudian ikus daiteke eta, aintzat hartu ahal den bezala, elementu gutxi batzuek osatzen dute. Alde batetik 2 bateria aurki daitezke, bakoitza 12 V-takoa FIAMM FG25505 eredukoak. Hauek 35 Ah eskentzen dute eta gainera, gainontzeko osagaiak elikatzeke, seriean konektatuta daude 24 V-ko tentsioa lortuz. Bateria hauek birkargatzeko beharra dute, korrante elektrikoari konektatuz eta horretarako Victron Smart IP65 kargatzailea erabiltzen da. Hiru elementu hauek beraz, elikaduraren arazoari irtenbidea ematen diete.

Kontrolaren ataza gauzatzeko, elementu garrantzitsuena Beckhoff IPC-a (*Industrial Personal Computer*) da, hau da, ordenagailua. Hauek Windows sistema operatiboarekin funtzionatzeaz gain, denbora errealean lan egiteko aukera ematen dute. Ordenagailu hau C6925-0030 ereduko da eta

prozesadore bezala Intel Atom E3815 bat dauka. Ezaugarri bezala aipatu daiteke, 1,64 GHz-ko frekuentziarekin lan egiten duela eta nukleo bakarreko CPU bat dela, 512 Kb-eko cache memoriarekin. Modu horretan, EtherCat komunikazioaren bidez sarrera edo irteera industrialen informazioa jaso/bidali dezake berehala. Konexio hauek gauzatzeko, beste txartel bat darama, EtherCat EK1100 eredua hain zuzen ere, irteera eta sarrera digital eta analogikoak dituena. Kontaktore bat ere dauka instalatuta, segurtasun-botoia maneiatzeko.



Irudia 25: Unibertsitateko AMR-aren barne konfigurazioa

Kanpoaldeari dagokionez, hori kolorekoa da, industriaren barruan funtzionamenduan dagoenean ondo desberdintzeko. Plataforma mugikor hau, hiru gurpilez ekipaturik dago; hauetako bi motor bati helduta eta hirugarrena libreki bira eta mugitu daitekeena. Beraz, ulertu daiteke bi gurpil elektriko dituela, 24 V-ko tentsioa jasan dezaketenak eta agindu bidez mugitu daitezkeenak. Gainontzeko kanpo elementuak, kamerak edo LiDAR sentsoreak adibidez, aplikazioaren baitakoak dira eta beste kanpo ordenagailu baten konektatuko lirateke, programazioa egiteko eta exekutatzeko erabiltzen den ordenagailua hain zuzen ere.

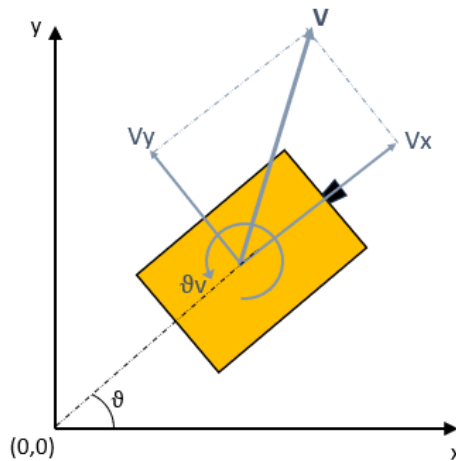
1.2.AMR-aren programazioa

Lan honetarako programazioa MATLAB-en bidez egin da baina ez da aukera bakarra, Python bezalako hizkuntzak ere erabiltzeko aukera dagoelako. Hala ere, AMR propioari aginduak ematen dizkiona Beckhoff-eko automata da beraz, TwinCat plataforma erabili beharra dago. Horrela, MATLAB eta TwinCat programen arteko komunikazioa egin daiteke eta bigarren honek, ibilgailuak mugitzeko behar duen informazioa transmitituko du motorrera. Informazio hau programazioaren baitakoa da baina MATLAB-ek bidali dezakeen balio sinpleenak eta minimoak abiadura lineala eta biraketa angelua dira, bi kontsigna horiekin AMR-a martxan jartzeko gai bait. Hau, ikasleek diseinatutako TwinCat-erako programaren eskakizuna da.

Kontuan izan behar da, Windows barnean denbora errealean funtzionatu nahi baldin bada, liburutegi bereziak behar direla eta ondorioz, TwinCat *software*-aren bitartez programatu behar direla kontsigna horien bidalketa gurpilen motorrari. MATLAB eta TwinCat arteko komunikazioa oso azkarra da ere baina azken finean, TwinCat-a da zuzenean Beckhoff-ek eskaintzen duen programazio

ingurunea beraz ordena hori jarraitu behar du informazioaren bidalketak. Beste alde batetik, programaren diseinua egiterakoan, prozesadorearen %80-a TwinCat-eko denbora errealeko programazioari utzi behar zaiola kontsideratu behar da, esan bezala CPU bakarreko IPC-a delako.

MATLAB-eko programazioan arreta jarriz, AMR-aren dinamika kontuan izateko beharra dago, aipatutako kontsignak lortzeko. Ibilgailu autonomo honek mugimendu bai translazionalak bai errotazionalak egin ditzake. Errotazio hori bi gurpiletako bati abiadura handiagoa emanaz lortzen da. Modu horretan, bi abiadura lineal dituela kontsidera daiteke, V_x [m/s] eta V_y [m/s] eta hauek sortzen duten bektorea V [m/s] abiadura definituko dute. Abiadura honek, ϑ_v [rad] balioaren eragin handia izango du, AV-ak hartuko den norabidea zehaztuz. Azken bi balio hauek izango dira beraz, TwinCat-ak jaso beharko dituenak.



Irudia 26: Ibilgailu autonomoan eragina duten abiadura ezberdinak

Irudia 26-an agertzen diren akzioak bilduz, guztiak kontsideratzen dituen \vec{u} bektorea defini daiteke, eta akzio-bektore hau edozein aldiune t -an aplikagarria izango da.

$$\vec{u}(t) = \begin{bmatrix} V_x(t) \\ V_y(t) \\ \vartheta_v(t) \end{bmatrix} \quad (1)$$

Behin ibilgailuaren gain eragingo duten parametroak ezagunak direla, dinamika definitu daiteke etorkizuneko aldiunetan, AV-ak hartuko dituen posizioak identifikatzeko. Hurrengo ekuazioak planteatzen dira, non x [m] eta y [m] ibilgailuaren posizioak diren planoan. Gainera, Δt [s] parametroa ere barneratzen da, hau integrazio urratsa izanez.

$$x(t + 1) = x(t) + u_1 \Delta t \quad (2)$$

$$y(t + 1) = y(t) + u_2 \Delta t \quad (3)$$

Aipatu den bezala, kontsigna bezala V [m/s] abiadura eta ϑ_v [rad] angelua behar dira beraz aurreko ekuazioak hauen baitan jartzeko aukera dago, hurrengo erlazioetan erakusten den erara:

$$x(t + 1) = x(t) + V(t) \cos(\vartheta_v) \Delta t \quad (4)$$

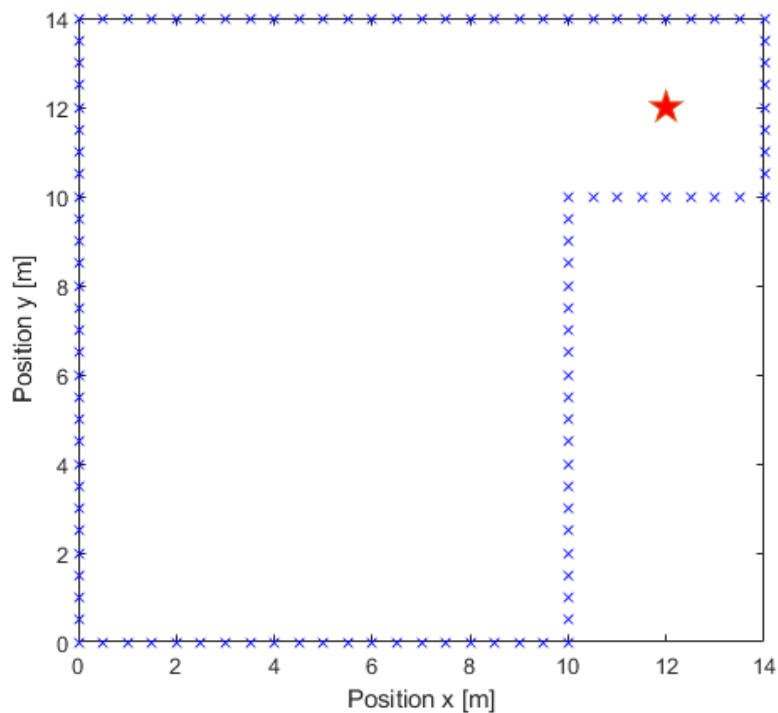
$$y(t + 1) = y(t) + V(t) \sin(\vartheta_v) \Delta t \quad (5)$$

Ibilgailuak erabiliko duen kontrol araua definiturik geratzen da ekuazio hauekin. Horrela, gainontzeko algoritmoei pausu eman ahal zaie, lortu behar diren balioak jakinak direlako.

2. Errefortzu Bidezko Ikaskuntzaren Optimizazioa

Nabigazio algoritmo bat diseinatzerako orduan, *Reinforcemet Learnig*-aren teknikan oinarritzea ideia ona da, ustiatuago dauden metodoetatik aldentzeko nahian. Aurreko atalean aipatu den bezala, zaila da espero diren emaitzak lortzea teknikari zehaztasun asko falta zaizkiolako. Aktorearen sare neuronalean garrantzi gehiena jartzea erabaki da, ondorioz, hau optimizatzeko lanetan esfortzua jarritz. Ataza hau betetzeko zenbait bide erabili daitezke baina, lan honetan, eredu analitikoa eta eboluzio bidezko optimizazioak planteatzen dira.

Helburua sinplifikatzeko lanetan, **Ekuzio (4)** eta **Ekuzio (5)**-etako dinamika erabiliko da zinetan V [m/s] abiadura balio finko bat izango duen, $V = 1$ [m/s] izanik. Obserbazioak, hau da, miatuko diren parametroak sei izango dira x [m], y [m], $dist_{Helmuga}$ [m] eta $dist_{Ozt.Gertu}$ [m], azken bi hauek x eta y aldagaietan banatuta harturik. Gainera mapa simple bat definituko da, hurrengo **Irudia 27**-an agertzen dena. Aipatzekoa da integrazio urratsa $dt = 0,1$ [s]-takoa izango dela.



Irudia 27: Nabigazio eremuaren 2D mapa

Nabigazio helburua ere sinplea da, ibilgailu autonomoa, hau da, agentea, maparen zehar mugitu behar da helmugara helduz, jakina denez, ingurua finkatzen duten oztopoekin talka egin gabe. Ulertzen da beraz, ingurunea mugatzen dute hormak traba bezala kontsideratzen dira eta “x” urdinekin errepresentatuta daude. Optimizazio baten bilaketan ikerketa egiten denez, ingurua sinplifikatzeko ez dira oztopoak sakabanatzen. Helmuga [12,12] posizioan dago eta izar gorri batek adierazten du.

Reward (r) edo sari funtzioak, bi aldagai kontsideratuko ditu. Alde batetik helmugara heltzeko distantzia eta $a_1 = 20$ balioko garrantzia emango zaio. Beste aldetik, oztopoekiko duen distantzia, zehatzago izateko, oztopo hurbilenari, $a_2 = 100$ balioaz kontsideratuz. Horrela hurrengo ekuazioak defini daitezke:

$$dist_{Helmuga}(t) = norm([x_{helmuga}(t), y_{helmuga}(t)] - [x(t), y(t)]) \quad (6)$$

$$dist_{Ozt.Gertu.}(t) = norm([x_{Ozt.Gertu.}(t), y_{Ozt.Gertu.}(t)] - [x(t), y(t)]) \quad (7)$$

$$r(t) = a_1 \left(\frac{1}{dist_{Helmuga}(t)} \right)^2 + a_2 (dist_{Ozt.Gertu.}(t))^2 \quad (8)$$

Entrenamenduak gainera amaitze baldintza batzuk izango dituzte, garaia amaituz hauetako bat betetzen baldin bada. Alde batetik eta argiena, limiteetatik agentea irtetzerakoan beste iterazio batera pausu emango da. Bestetik, oztopo batetik 0,3 [m] baino gertuago dagoen kasuetan ere eten egingo da. Azkenik, helmugara heltzerakoan, helburua lortu duenez ez da iterazioan entrenatzen jarraituko.

Beste alderdi garrantzitsu bat entrenamenduen hasiera puntua arbitrarioa izatea da eta noski, aurreko baldintzekin bateragarriak izateko, hormen edo helmugaren gainean ez hastea. Parametro orokor hauek definituta geratzen direnez, optimizazioaren planteamendura pausu ematen da.

2.1.Optimizazio analitikoa

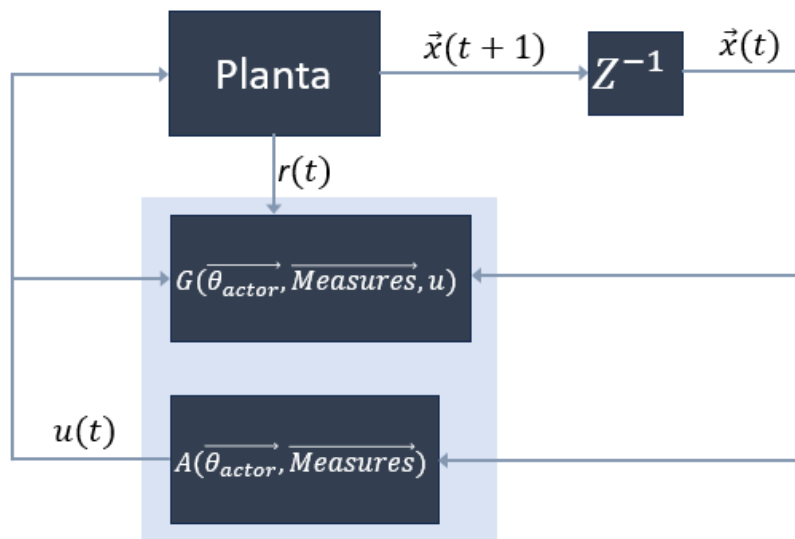
Errefortzu bidezko ikaskuntzaren planteamendua definitzeko **Irudia 28**-a aurkezten da. Bertan planta ibilgailu autonomoa da, aurrez definitutako ingurunean kokatuta. Formulazioa sinplifikatzeko AV-aren posizioa bektore batean bildu da, beraz $\overrightarrow{pos}(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix}$ bi posizio espazialak elkartzen ditu. $\vec{u}(t)$ -ren kasuan, akzioa errepresentatzen jarraitzen du baina oraingoan balio bakarra bilduko du, hau da, $u(t) = \vartheta_v(t)$. Esan bezala, abiadura konstante kontsideratzen da beraz beste bi akzioak kontuan izateak zentzua galtzen du. Horrela akzioa lortuko da, ibilgailuak hurrengo posizioa lortzeko behar duen informazioa izanda. Antzezlearen kasuan, hala nola, A -ren kasuan, obserbazioen bektorea izan beharko da kontuan eta antzezlearen sare neuronalaren pisu sinaptikoak, $\overrightarrow{\theta}_{actor}$. Bektore hori hurrengo erara definitzen da, non $x_h = x_{helmuga}$ eta $y_h = y_{helmuga}$ balioak diren eta $x_o = x_{Ozt.Gertu.}$ eta $y_o = y_{Ozt.Gertu.}$ diren, notazioa sinplifikatzeko.

$$\overrightarrow{Measures}(t) = \begin{bmatrix} x(t) \\ y(t) \\ (x_h - x)(t) \\ (y_h - y)(t) \\ (x_o - x)(t) \\ (y_o - y)(t) \end{bmatrix} \quad (9)$$

Hurrengo ekuazioan dinamika definitzen da, beste era batera esanda, planta kontrolatzen da, akzioaren datua kontsideratuz.

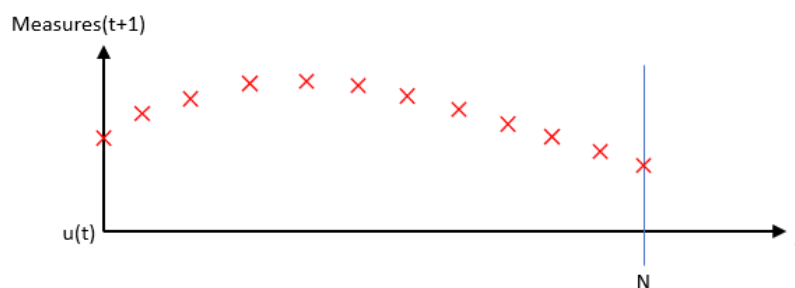
$$\overrightarrow{Measures}(t + 1) = f_{step}(\overrightarrow{Measures}(t), u(t)) \quad (10)$$

Gainera, akzioa kritikorengatik (G) ere ebaluatua izango da, posizio bektorea eta antzezlearen sare neuronalaren pisu sinaptikoekin batera. Horretarako, *reward*-aren balioa jasotzea ezinbestekoa du.



Irudia 28: Errefortzu bidezko ikaskuntzaren eskema

Sistemaren funtzionamendua zehaztuta geratzen da beraz eskema honekin. Kritikoan erreparatuz, ibilgailua hurrengo aldiunean hartuko duen posizioa ebaluatzen duela jakina da. Etorkizuneko posizio horiek hurrengo irudikapenean bezala errepresentatu daitezke denboran zehar, N iterazio arte, horizontea. Gainera, posizio horiek gainontzeko obserbaziotan eragin zuzena dutela antzeman daiteke.



Irudia 29: AV-aren hurrengo posizioen errepresentazioa denboran zehar

Ondorengo zehaztapenak egin daitezke **Irudia 29**-an kontu eginik eta **Ekuzio (10)**-aren definizioa jakinda. Azken finean, hurrengo egoera, momentuan duen egoeraren eta horretan akzioak duen eraginaren ondorioz lortuko da.

$$\overrightarrow{Meas.}(0), \overbrace{f_{step}(\overrightarrow{Meas.}(0), u(0))}^{\overrightarrow{Meas.}(1)}, \overbrace{f_{step}(\overrightarrow{Meas.}(1), u(1))}^{\overrightarrow{Meas.}(2)}, \dots, \overbrace{f_{step}(\overrightarrow{Meas.}(N-1), u(N-1))}^{\overrightarrow{Meas.}(N)} \quad (11)$$

Bestalde, akzioa antzezlearen ondorioz lortuko da, hau da, $\overrightarrow{\theta}_{actor}$ pisu sinaptikoez hartuko duten balioen ondorioz. Hau horrela definituz, kritikoa kontzeptu guzti hauen menpeko funtzio bat bezala finkatu daiteke.

$$G = F(\overrightarrow{Meas.}(0), \overrightarrow{Meas.}(1), \overrightarrow{Meas.}(2), \dots, \overrightarrow{Meas.}(N)) = F(\overrightarrow{Meas.}(t), \overrightarrow{\theta}_{actor}) \quad (12)$$

Honetaz gain, jakina da kritikoa ere *reward* funtzioaren baitakoa dela, bere ebaluaketa sari horren balioaren arabera izango bait da. Kritikoa hurrengo erlazioaren bidez guztiz definituta geratzen da eta antzezlearen sarearen menpekota izango da, **Irudia 28**-an seinatu den bezala. Menpekotasun hau ibilgailuaren aldiuneko posizioaren baitan egongo da aurrez ikusi denez. Gainera jakin beharra dago agente batek lortzen duen saria N aldiunetan lortutako *reward* osoa izango dela. Entrenamendurako interesa hurrengo aldiunean jasoko duen sarian dago, horrela akzioa barnean kontsideratuta egonik:

$$G = \sum_{t=0}^{t=N} r(\overrightarrow{Measures}(t), u(t)) = \sum_{t=0}^{t=N} r(\overrightarrow{Measures}(t+1)) \quad (13)$$

Akzioaren balioa lortzeko, sare neuronal baten erabilera planteatzen da, antzezlearen sare neuronal hain zuzen ere. Sare neuronalaren sarrerak kasu honetan, ibilgailu autonomoaren obserbazioak eta $\overrightarrow{\theta}_{actor}$ izango dira, hurrengo erlazioan erakusten den bezala.

$$u(t) = f_{NN}(\overrightarrow{\theta}_{actor}, \overrightarrow{Measures}(t)) \quad (14)$$

Sare neuronal honen egitura oso simplea da, diseinuan baino gehiago doikuntzan dagoelako interesa. Hasteko ezaugarri sarrera bat dago, obserbazioak jasotzeko, hau da, $\overrightarrow{Measures}(t)$. Ondoren 50 irteera dituen *Fully Connected* (FC) eta *ReLU* geruza bi bikote daude. Horrela nodo guztiak konektatzen dira gradienteari beherakorrari abiadura emanez bat egiteko. Hirugarren *Fully Connected* geruza bat dago, irteera bakar batekin, ondoren *Tahn* aktibazio funtzio geruzak -1 eta 1 artean balioa finkatu dezan eta azken *scalingLayer*-ak $\pi/2$ baliora mugatzeko akzioaren eragina. Horrela, ibilgailuak izango duen biraketa mugatzen da $[-\pi/2, \pi/2]$ tartean.



Irudia 30: Antzezlearen sare neuronal

Hurrengo taulak, antzezlearen sare neuronalen ezaugarriak biltzen ditu.

Taula 6: Antzezlearen sare neuronalaren ezaugarriak

Geruzaren Izena	Mota	Aktibazioak	Propietate Ikasgarriak
observation	Feature Input	6 (C) × 1(B)	-
ActorFC1	Fully Connected	50 (C) × 1(B)	Weights 50 × 2 Bias 50 × 1
ActorRelu1	ReLU	50 (C) × 1(B)	-
ActorFC2	Fully Connected	50 (C) × 1(B)	Weights 50 × 50 Bias 50 × 1
ActorRelu2	ReLU	50 (C) × 1(B)	-
ActorFC3	Fully Connected	1 (C) × 1(B)	Weights 1 × 2 Bias 1 × 1
ActorTanh	Tanh	1 (C) × 1(B)	-
ActorScaling	ScalingLayer	1 (C) × 1(B)	-

Errefortzu bidezko sistema definituta egongo litzateke baina planteatutako sare neuronalaren entrenamendurako jarraitu behar diren pausuak finkatu behar dira. Optimizazio honetan sare neuronalaren pisu sinaptikoak aktualizatzen dira garaiko. Horretarako, k parametroa eta t parametroaren arteko desberdintasuna aipatu beharra dago. Lehenengoak entrenamendu garaia egiten dio erreferentzia eta bigarrenak, simulazioan 0 aldiunea eta N horizontearen artean dauden denbora aldaketei, hurrenez hurren. Entrenatzerakoan eta, ondorioz, sarearen pisuak eguneratzerakoan k -ak markatuko du prozesua. Horrela izanda, $\overrightarrow{\theta}_{actor}$ ondorengo erlazioaren bidez definitzen da:

$$\overrightarrow{\theta}_{actor}(k+1) = \overrightarrow{\theta}_{actor}(k) + \alpha \frac{\partial}{\partial \overrightarrow{\theta}_{actor}(k)} G \quad (15)$$

Aurreko ekuazio honetan, entrenamendu garaian aurrera egiteko, aurreko garaiko pisu sinaptikoei beste termino bat gehitzen zaie. Bigarren hau gradiente bat interpretatzen du, hau da, G eskalarra da eta pisu sinaptiko pillo baten menpe dago. α koefizienteak biderkatuko du, hobekuntza ratioa dena eta adaptatiboa dena. Hurrengo ekuazioak gradiente honen xehatzea adierazten du eta **Ekuzio (13)**-a berreskuratuz planteatzen da.

$$\frac{\partial}{\partial \overrightarrow{\theta}_{actor}(k)} G = \sum_{t=0}^{t=N} \left(\frac{\partial}{\partial \overrightarrow{\theta}_{actor}(k)} r(\overrightarrow{Measures}(t+1)) \right) \quad (16)$$

Batukariak sari metatzeari egiten dio erreferentzia. Gainera ekuazio hau banatu egin daiteke termino ezberdinetan. Alde batetik, sariaren balioa dago ibilgailuak duen obserbazioekiko. Bestetik, obserbazioak posizioarekiko eta azkenik, posizioa pisu sinaptikoeikiko. Banaketa hau, hiru termino hauek eskuragarriagoak direlako ekartzen da.

$$\frac{\partial}{\partial \overrightarrow{\theta}_{actor}(k)} G = \sum_{t=0}^{t=N} \left(\frac{\partial}{\partial \overrightarrow{Meas.}(t+1)} r \right) \left(\frac{\partial}{\partial \overrightarrow{pos}(t+1)} \overrightarrow{Meas.}(t+1) \right) \left(\frac{\partial}{\partial \overrightarrow{\theta}_{actor}(k)} \overrightarrow{pos}(t+1) \right) \quad (17)$$

Lehengo atalaren balioa era sinplean lortzen da, sari funtzioaren deribatuak planteatuz obserbazioko termino bakoitzarekiko. Bigarren atala aldiz matrize bat izango da, posizioak obserbaziotan eragina izango duten aldagaiekin. Ondorengo ekuazioez egongo dira definituta beraz bi termino hauek:

$$\left[\frac{\partial}{\partial x(t+1)} r \quad \frac{\partial}{\partial y(t+1)} r \quad \frac{\partial}{\partial (x_h - x)(t+1)} r \quad \frac{\partial}{\partial (y_h - y)(t+1)} r \quad \frac{\partial}{\partial (x_o - x)(t+1)} r \quad \frac{\partial}{\partial (y_o - x)(t+1)} r \right] \left(\frac{\partial}{\partial \overrightarrow{Meas.}(t+1)} r \right) = \quad (18)$$

$$\left(\frac{\partial}{\partial \overrightarrow{pos}(t+1)} \overrightarrow{Meas.}(t+1) \right) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \quad (19)$$

Reward funtzioa gogoratzuz (ikusi **Ekuazio (8)**), interesa dagoen terminoen baitan idatz daiteke, deribatuak egiteko. Ondorioz, hurrengo eran planteatu daiteke ekuazioa:

$$r(t+1) = a_1 \left(\frac{1}{((x_h - x)(t+1))^2 + ((y_h - y)(t+1))^2} \right) + a_2 \left(((x_o - x)(t+1))^2 + ((y_o - y)(t+1))^2 \right) \quad (20)$$

Deribatuak orain argiak dira eta modu sinplean lortu daitezke. Hurrengo erlazioak aurkezten dute hauen balioak.

$$\frac{\partial}{\partial x(t+1)} r = -2a_1 \left(\frac{(x_h - x)(t+1)}{\left(((x_h - x)(t+1))^2 + ((y_h - y)(t+1))^2 \right)^2} \right) - 2a_2((x_o - x)(t+1)) \quad (21)$$

$$\frac{\partial}{\partial y(t+1)} r = -2a_1 \left(\frac{(y_h - y)(t+1)}{\left(((x_h - x)(t+1))^2 + ((y_h - y)(t+1))^2 \right)^2} \right) - 2a_2((y_o - y)(t+1)) \quad (22)$$

$$\frac{\partial}{\partial (x_h - x)(t+1)} r = 2a_1 \left(\frac{(x_h - x)(t+1)}{\left(((x_h - x)(t+1))^2 + ((y_h - y)(t+1))^2 \right)^2} \right) \quad (23)$$

$$\frac{\partial}{\partial (y_h - y)(t+1)} r = 2a_1 \left(\frac{(y_h - y)(t+1)}{\left(((x_h - x)(t+1))^2 + ((y_h - y)(t+1))^2 \right)^2} \right) \quad (24)$$

$$\frac{\partial}{\partial (x_o - x)(t+1)} r = 2a_2((x_o - x)(t+1)) \quad (25)$$

$$\frac{\partial}{\partial (y_o - x)(t+1)} r = 2a_2((y_o - y)(t+1)) \quad (26)$$

Ekuzio (17)-ko azken terminoari banakatze bat egin behar zaio kalkulatzeko errazago edo zuzenagoak diren erak proposatuz. Azken termino hau bestalde, sentsibilitate ekuazio bezala ere izendatu daiteke eta ondorengo itxura hartzen du:

$$\begin{aligned} & \frac{\partial}{\partial \vec{\theta}_{actor}(k)} \vec{pos}(t+1) \\ = & \left(\frac{\partial}{\partial u(t)} f_{step} \right) \left(\frac{\partial}{\partial \vec{\theta}_{actor}(k)} u(t) \right) + \left(\frac{\partial}{\partial \vec{pos}(t)} f_{step} \right) \left(\frac{\partial}{\partial \vec{\theta}_{actor}(k)} \vec{pos}(t) \right) \end{aligned} \quad (27)$$

Sentsibilitate ekuazioak uneko urratseko akzioa hartu beharko du kontutan, posizioarekin batera eta dinamikarekiko sare neuronalaren pisuak kalkulatu, azken finean $\vec{pos}(t+1) = f_{step}(\vec{pos}(t), u(t))$ bait da. **Ekuzio (27)**-an ere antzeman daiteke sistema dinamiko bat balitz bezalako portaera izango duela.

Entrenamendu dinamiko bat egiteko eta ondorioz, sare neuronalaren pisuen eguneratzea gauzatzeko, sinplifikazio batzuk egin daitezke sentsibilitate ekuazioaren terminoetan. Alde batetik, sistemak metatze moduan funtzionatzen duenez f_{step} -ren deribatua egoerarekiko, kasu honetan, 1 balioa hartuko du beti, hau da, matrize batera berdintzeko aukera dago. Beste era batean esanda, ibilgailuaren dinamikan erreparatuz, $\vec{pos}(t+1)$ lortzeko $\vec{pos}(t)$ -ri beste termino bat gehitzen zaiolako gertatzen da (ikusi **Ekuzioa (2)** eta **Ekuzioa (3)**). Bateragarria izateko baina $\vec{Measures}$ -en tamainakoa izan beharko da, zutabe eta errenkadetan.

$$\left(\frac{\partial}{\partial \vec{pos}(t)} f_{step} \right) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (28)$$

Sinplifikazioekin jarraituz, jakina da ere AV-aren hurrengo egoera lortzeko akzioak eragina duela, pausuaren norabidea zehaztuz **Ekuzio (4)** eta **Ekuzio (5)**-an ikusi daitekeen erara. Beraz, **Ekuzio (9)**-a ren forma gogoratuz, hurrengo sinplifikatzea planteatu daiteke:

$$\left(\frac{\partial}{\partial u(t)} f_{step} \right) = \begin{bmatrix} -\sin(u(t)) \\ \cos(u(t)) \\ \sin(u(t)) \\ -\cos(u(t)) \\ \sin(u(t)) \\ -\cos(u(t)) \end{bmatrix} \quad (29)$$

Honetaz gain, sentsibilitate ekuazioaren emaitzak izango duen itxura ere aurreikusi daiteke. Matrize bat izango da S_{pos} . bezala izendatuko dena eta honen dimentsioak hurrengoak izango dira: alde batetik, errenkadak sare neuronalaren ikasgarriekin bat egingo dute ($nlearnables$), bestetik, zutabeak egoerekin bat etorriko dira ($nmeasures$).

$$\frac{\partial}{\partial \vec{\theta}_{actor}(k)} \vec{pos}(t) = S_{posnlearnables,nmeas.}(t) \quad (30)$$

Sare neuronalaren pisuak eguneratzeko beraz, hurrengo erlazioa erabiliko da. Gainera, gradienteak bakarrik eskalarrekiko kalkulatzen direla gogoratu, ez zen arazorik suertatuko.

$$\left(\frac{\partial}{\partial u(t)} f_{step}\right) \left[\frac{\partial}{\partial \theta_{actor(1)}} u(t) \dots \dots \frac{\partial}{\partial \theta_{actor(nlearnables)}} u(t) \right] + \left(\frac{\partial}{\partial \vec{p}_{os}(t)} f_{step}\right) S_{x_{nlearnables, nstates}}(t) \quad (31)$$

Aipatu behar da ekuazio hau zero baliora hasieratuta dagoela, hasierako baldintzak ikasgarriekiko independenteak direlako. Modu honetan sentsibilitatea lortzen da eta ondoren **Ekuazio (17)**-ean aplikatuko zen. Prozesua argitzeko asmoarekin, hurrengo taula aurkezten da, algoritmoaren fluxua errepresentatuz.

Taula 7: Errefortzuzko ikaskuntzaren planteamenduaren algoritmoaren fluxua

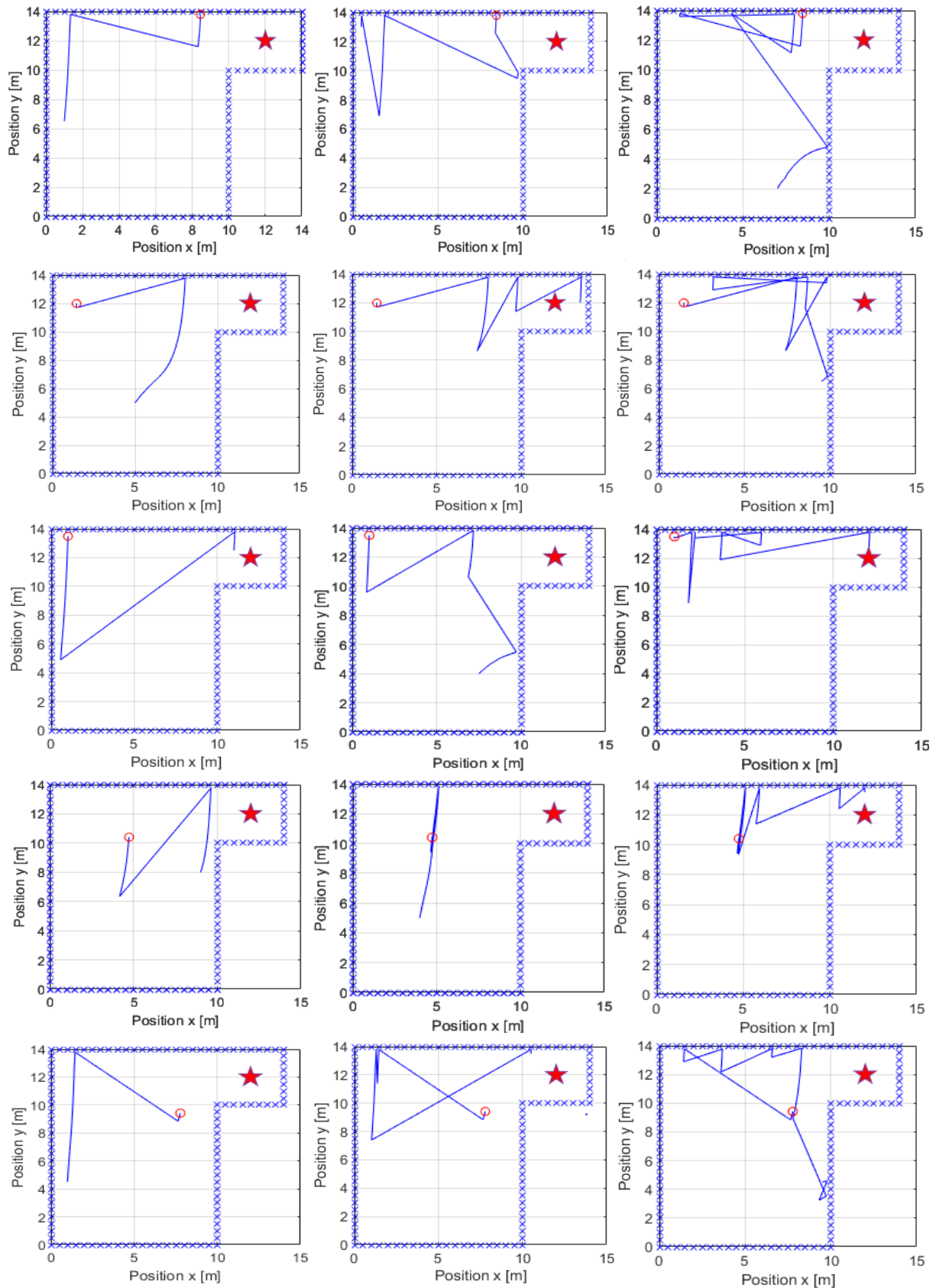
Errefortzuzko ikaskuntza optimizatua	
1:	Antzezlearen sare neuronalaren definizio eta sorkuntza
2:	Hasierako parametroen definizio eta hasieratzea
3:	S_{pos} hasieratu
4:	for i = 1 to k
5:	for mapako posizio guztiak
6:	Measures eta u hasieratu
7:	for j = 1 to t
8:	Sare neuronala ebaluatu
9:	Akzioaren aurreikuspena egin
10:	Ibilgailuaren hurrengo egoera kalkulatu
11:	reward-aren balioa kalkulatu
12:	Egoeraren deribatua sariarekiko kalkulatu
13:	Akzioaren deribatua pisuekiko kalkulatu
14:	Sare neuronalaren pisuak kalkulatu
15:	Sentsibilitate ekuazioa eguneratu
16:	Kritikoa metatu
17:	end
18:	Ikaskuntza tasa kalkulatu
19:	Sare neuronalaren ikasgarriak eguneratu
20:	end
21:	end
22:	Simulazioa egin

Prozesua aurrera eramateko, hurrengo balioak finkatu dira:

Taula 8: RL-analitikoaren sare neuronala entrenatzeko balioak

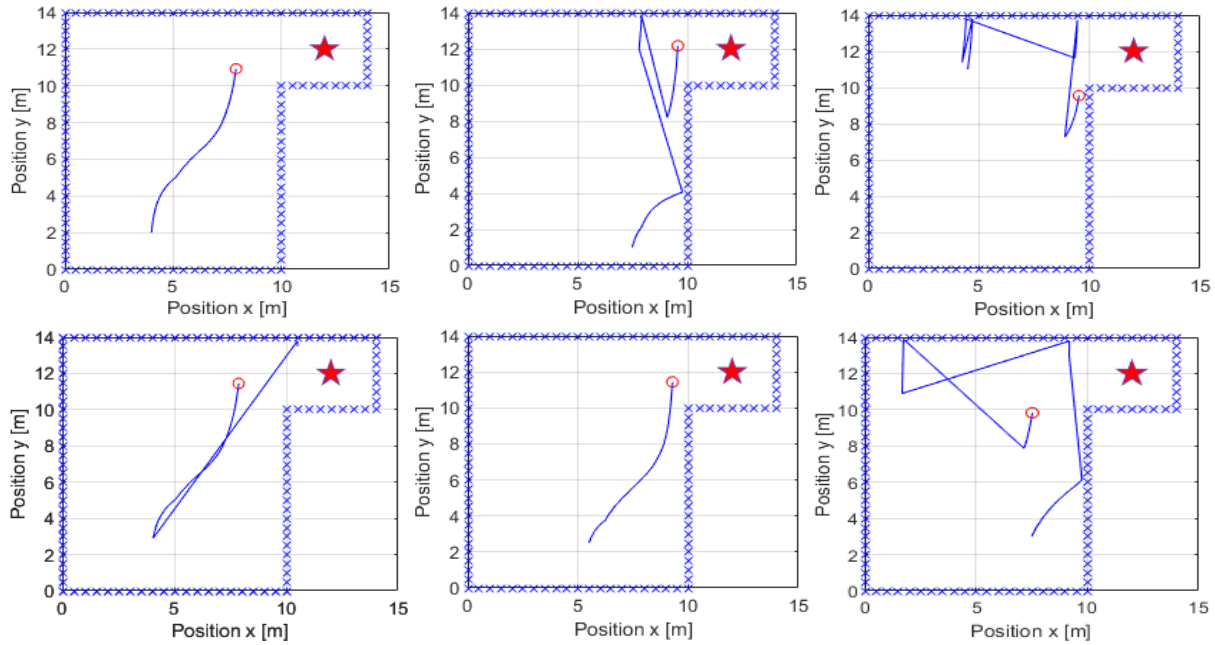
Parametroa	Balioa
Max Epochs	500000
Max Iterations	100
Velocity	1
Integration step	0,1
Initial Learning Ratio	$1 \cdot 10^{-6}$
Decay	0,75
Momentum	0,95

Parametro hauekin entrenatzen da sare neuronal, agente bat lortzeko eta alde aurretik simulazioan praktikan jartzeko. Entrenamenduak honako emaitzak ditu lehenengo garaietan, lerro urdina bidea izanda eta borobil gorria bidearen amaiera:



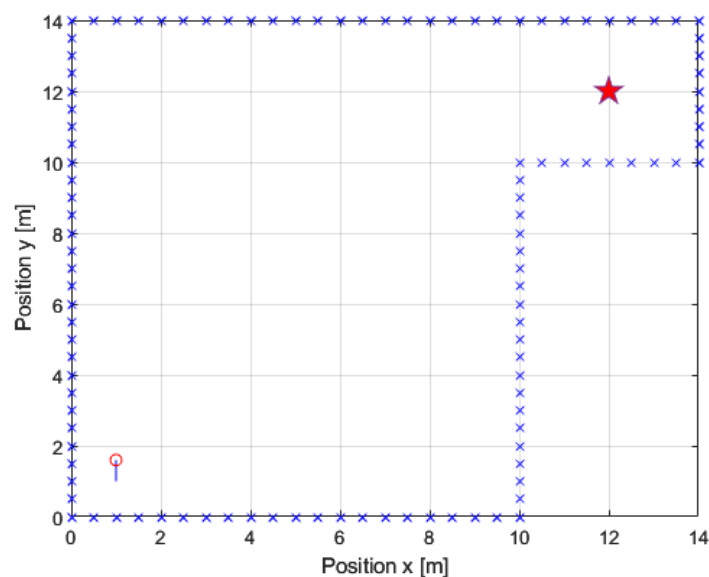
Irudia 31: Errefortzuko ikasketaren hasierako garaien simulazioa

Errenkadetan antzeman daitezenez agenteak maximo lokalak aurkitzen ditu baina garai batzuk bakarrik behar ditu bertatik irtetzeko. Maximo hauetako bat aurkitzerakoan gainera, ibilbide moldaketak egiten ditu bertara heltzeko, lortzen duen saria maximizatzeko. Gainera, errenkadetan behera egin ahala nabaria da pixkanaka helmugarekiko duen lerrokadura errorea zuzentzen doala. Hurrengo irudiak aurreratuago dauden garaietako simulazioak aurkezten ditu. Hauetan, aurreikusitako bideek kurbadurak sortzen hasi direla igartzen da. Horretaz gain lerrokadura errorea geroz eta gehiago zuzentzen da eta helmugara iristeko norabideari heldu nahi diola detektatzen da.



Irudia 32: Errefortzuzko ikaskuntzaren garai aurreratuago batzuen simulazioa

Nabaria denez prozesua ez da azkarra izango baina egin diren frogetan sortzen den agentea egonkorra dela baieztatu daiteke, ez delako arbitrarioa eta portaera konstante bat sumatzen delako. Entrenamendua luzea izango da lerrokadura errorea zuzentzen ikasi behar duelako, beste hitzetan, konbergentzia indartsua behar du. Hau garai gutxi entrenatu duten agenteetan nabaria da, ondorengo irudian ikusten den bezala. Honakoa 191-garitan entrenatu ostean lortutako emaitza da.



Irudia 33: Antzezle optimizatuaren emaitza 191 garaitan

Entrenamendu hau 100 simulazio pausu irudikatzen ditu. Irudian ez da nabaria baina datuetan antzeman daiteke ibilgailuak aurrera egiten duela eta urrats batzuen ostean biratu egiten duela. Entrenamendu falta duenez biraketa oso bat egiten du eta kontrako norabidera aurrera egiten du. Norabide egokia ez dela konturatzen da eta berriz ere biraketa egiten du baina akzioa doitzen ez dakenez berriz ere kontrako norabidera abiarazten da. Horrela, aurrera eta atzera geratzen da pauso guztiak bete arte. Igarri daiteke beraz entrenamendu falta eta denbora behar duela akzioaren balio egokia aurreikusi ahal izateko. Aipatzekoa da ere denbora terminoetan sareak hurrengo egoera bat lortzeko 0,00732 [s] inguru behar dituela eta 100 urratsetako bide bat 1,748 [s]-tan lortzen dela.

2.2. Eboluzio bidezko optimizazioa

Kalkulu analitikoa egin beharrean, beste optimizazio era bat planteatzen da antzezlearen sare neuronalaren topologia egokituz. Bibliografian antzeman izan denez, algoritmo ebolutiboak prozesuak optimizatzeko aukera ematen dute. Gainera, sistema jarriak ere optimizatzeko ahalmena dute. Kasu honetan baina, sarearen eboluzioa aztertzen da plantarena beharrean. Modu honetan, antzezlearen sare neuronal bat diseinatzen da, iteratiboki eboluzionatu dezan eta honekin, *Reinforcement Learning*-eko agentea sortzeko pausuak jarraitzen dira.

NEAT [418] izenarekin ezaguna den algoritmoa adibidez, algoritmo ebolutibo bat da sare neuronalak sortzen dituen. Honek, bi gene parekin eraikitzen du sare neuronala, nodoak eta hauen arteko konexioak izanik. Algoritmo honen funtzionamendua kontuan izanda, kasu honetarako planteamendu bat egiten da. Prozesua antzezlearen sare neuronal sinpleenarekin hasten da. Planteamendu honetan, eta aurreko optimizazioarekin bat egiteko, obserbazioak kontuan izango dituen ezaugarri geruza batek, 50 irteerako *Fully Connected* geruza batek, *ReLU* geruzak, irteera bat duen *Fully Connected* bat eta *tanh* eta *scalingLayer*-ak osatuko dute hasierako sarea. Hurrengo taulak biltzen ditu sare sinple honen ezaugarriak.

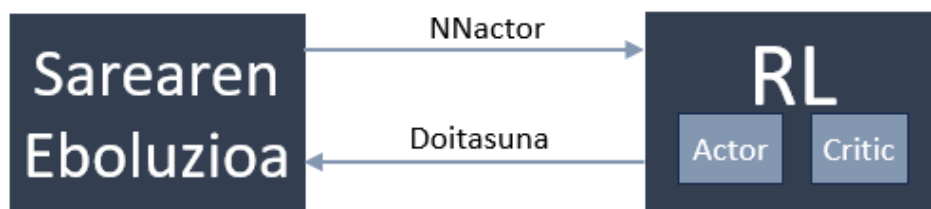
Taula 9: Antzezlearen hasierako sare neuronalaren ezaugarriak

Geruzaren Izena	Mota	Aktibazioak	Propietate Ikasgarriak
observation	Feature Input	6 (C) × 1(B)	-
ActorFC1	Fully Connected	50 (C) × 1(B)	Weights 50 × 2 Bias 50 × 1
ActorRelu1	ReLU	50 (C) × 1(B)	-
ActorFC3	Fully Connected	1 (C) × 1(B)	Weights 1 × 2 Bias 1 × 1
ActorTanh	Tanh	1 (C) × 1(B)	-
ActorScaling	ScalingLayer	1 (C) × 1(B)	-

Sare honekin eta kritikoarenarekin batera, errefortzu bidezko ikaskuntzaren agente bat sortzeko prozesua abiarazten da. Entrenamendua egiterakoan Q_0 balioan jarriko da atentzioa. Balio hau espero den epe luzeko metatze-sariaren zenbatespen kritikoa da, agentea eta hasierako baldintzak kontsideratuz. Kritikoaren inferentzia eginez kalkulatzen da garai bakoitzaren hasieran eta kritikoaren entrenamenduaren egokitasuna neurtzen duen balio bat bezala erabili daiteke. Kritiko perfektua erabiliko balitz, hau da, garaiko hasierako obserbazioarekin epe luzeko saria zehaztasunez aurreikusteko gai izango zena, Q_0 balioak garaiko sariarekin bat egin beharko zuen. Hau ez da

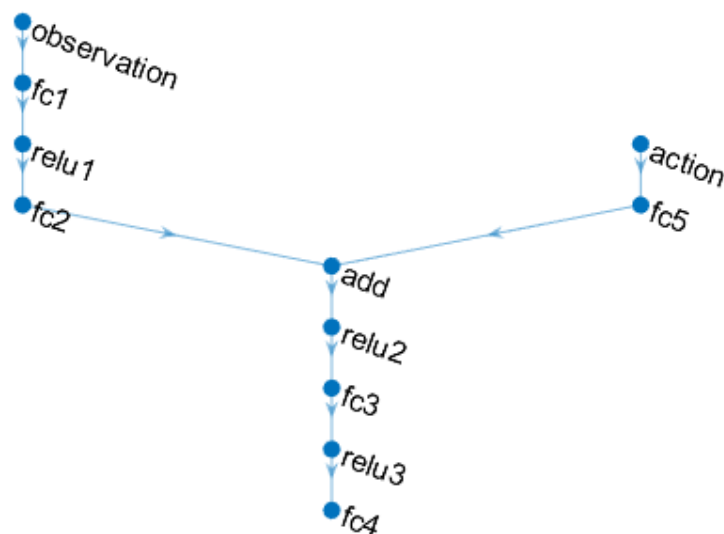
beharrezkoa izaten, antzezleak konbergitu egin dezakeelako lehenengo eta une horretan egokia da entrenamendua geratzea.

Kontzeptu hauekin finkatuko dira beraz entrenamendua gelditzeko baldintzak. Ondorioz, garai ezagun batean entrenamendu prozesua eten egiten da eta egokia den konprobatzen da. Garai kopuru horretan bilatzen den konbergentzia puntua lortu baldin bada, agentea egokitzen hartzen da eta prozesu osoa eten egiten da. Kontrakoa gertatzen bada, hau da, egokia ez dela kontsideratzen bada, antzezlearen sarea egokitu egiten da eboluzioaren ideiaz. Egokipen hori geruza pare bat sartuz egiten da iterazio bakoitzean, hau da, 50 irteera dituen *Fully Connected* eta *ReLU* geruza bikotea. Aztertu da orokorrean antzezlearen sare neuronalak bi geruza mota hauekin osatuta daudela nagusiki, beraz kriterio hori jarraitu da. Berriz ere agentea sortzeko prozesua abiarazten da sare berriarekin. Sistema hau errepikakorra izango da espero diren emaitzak lortu arte, hurrengo irudian hauteman daitekeen bezala.



Irudia 34: Errefortzu bidezko ikaskuntzaren sarearen eguneratzea

Era honetako optimizazioan berriz ere antzezlearen sare neuronalarekin egiten da lan, kritikoaren sarea eta sari funtzioa finko geratuz prozesu osoan. Antzezlearen sarean eboluzioa egongo da baina kritikoaren sarearen kasuan ez eta aurrez azaldu den bezala, bai AV-aren egoera bai akzioa izan behar ditu kontuan. Beraz 2 sarrera geruza aurkituko dira. Egoerari dagokion sarrera adarretik, *Fully Connected* geruza bat, *ReLU* geruza bat eta beste *Fully Connected* geruza daude jarraian, nodo guztiak konektatuz eta gradienteari beherakorrari abiadura emanez bat egiteko. Bestetik, akzioaren adarrean *Fully Connected* geruza bakarrak jarraitzen du, nodoak konektatzeko. Bi adarrak gehiketa geruza batean elkartzen dira. Ondoren, *ReLU* eta *Fully Connected*-ez osatutako bi bikote daude, sarearen topologia amaituz. Esan beharra dago, FC geruza guztien irteeraren tamaina 10-ekoa da, "fc4"-rena izan ezin, 1 balioa duena.



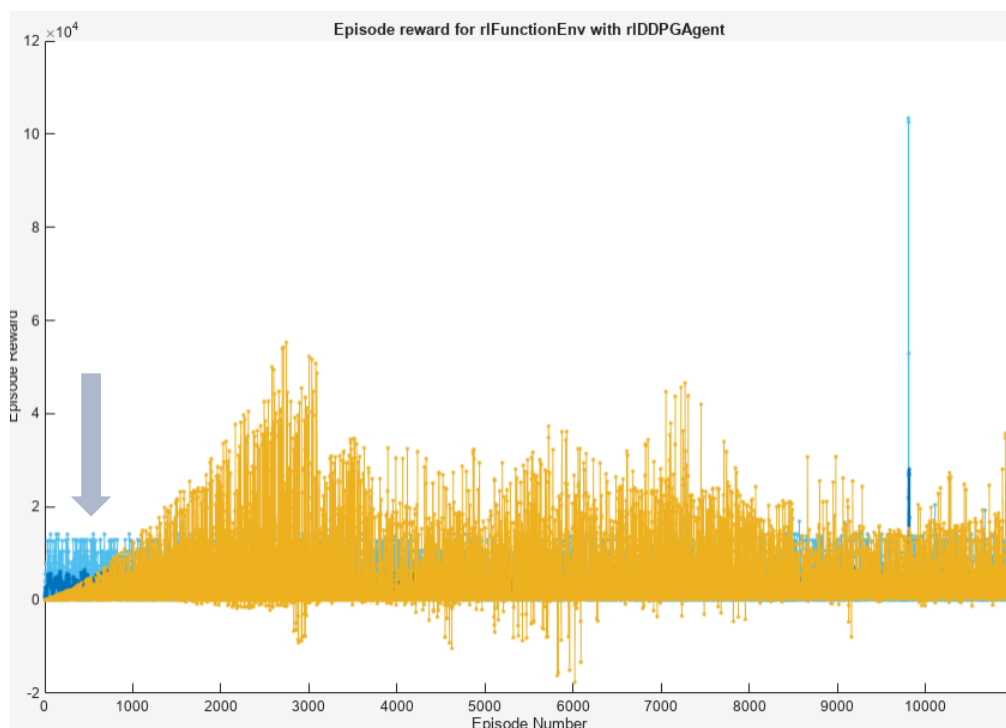
Irudia 35: Kritikoaren sare neuronal

Hurrengo taulak, kritikoaren sare neuronalaren ezaugarriak biltzen ditu.

Taula 10: Kritikoaren sare neuronalaren ezaugarriak

Geruzaren Izena	Mota	Aktibazioak	Propietate Ikasgarriak
observation	Feature Input	6 (C) × 1(B)	-
fc1	Fully Connected	10 (C) × 1(B)	Weights 10 × 2 Bias 10 × 1
relu1	ReLU	10 (C) × 1(B)	-
fc2	Fully Connected	10 (C) × 1(B)	Weights 10 × 10 Bias 10 × 1
action	Feature Input	1 (C) × 1(B)	-
fc5	Fully Connected	10 (C) × 1(B)	Weights 10 × 1 Bias 10 × 1
add	Addition	10 (C) × 1(B)	-
relu2	ReLU	10 (C) × 1(B)	-
fc3	Fully Connected	10 (C) × 1(B)	Weights 10 × 10 Bias 10 × 1
relu4	ReLU	10 (C) × 1(B)	-
fc4	Fully Connected	1 (C) × 1(B)	Weights 1 × 10 Bias 1 × 1

Aipatzekoa da bi sareek 1-eko ikaskuntza ratioa izango dutela, 1-eko *Gradient Threshold*-a, pisuen eguneraketarako, eta 0,01-eko *L2 Regularization Factor*-a, koefizienteen balioak galera funtzioari gehitzeko. Esan bezala, konbergentzia puntua finkatu behar da doitasuna neurtzeko. Sistema honen entrenamenduak hurrengo irudian aurkezten den erakoak dira. Bertan, urdin argiz garaiko sariaren balioa irudikatzen da. Urdin ilunez garaiko sarien batez bestekoa, 10 garai kontuan izanda eta azkenik, laranja, Q_0 balioa. Geziak entrenamendua eteteko bilatzen den puntuari egiten dio erreferentzia. DDPG prozedura jarraitu da entrenamendurako.



Irudia 36: Reinforcement Learning agentaren entrenamendu adibidea

Ondorioz, entrenamendu guztiak 1000 garaitan etengo dira, agenteari entrenamendu egokia izateko aukera emateko. Behin entrenamendua amaituta dagoela, batez besteko sariaren balioa Q_0 balioa baino txikiagoa den konprobatuko da, hori konbergentzia eman denaren seinale delako. Zehatzagoa izateko, azken 10 garaiko Q_0 -aren batez bestekoa egingo da, sariarenarekin bat etortzeko. Gainera agentea definitzeko parametroak ere finkatu behar dira, hurrengo taulan adierazita daudenak hain zuzen ere.

Taula 11: Agentea definituko duten parametroak

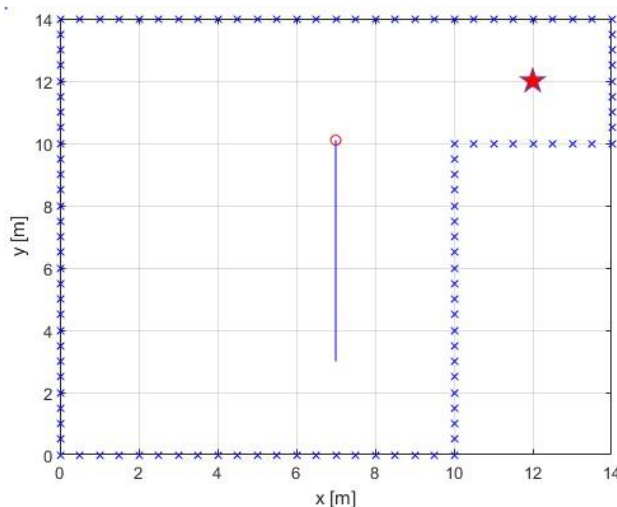
Parametroa	Balioa
Sample Time	0,1
Target Smooth Factor	$1 \cdot 10^{-3}$
Discount Factor	0,995
Mini Batch Size	128
Experience Buffer Length	$1 \cdot 10^6$
Noise Variance	0,1
Noise Variance Decay Rate	$1 \cdot 10^{-5}$

Parametro guztiak definituta daudelarik, entrenamendu prozesuarekin hasi daiteke. **Taula 12**-an algoritmo honen fluxua ikus daiteke, era ulergarriagoan adieraziz.

Taula 12: RL eta GA algoritmoaren fluxua

Errefortzuzko ikaskuntza optimizatua algoritmo genetikoaren bidez	
1:	Parametroak hasieratu
2:	Eboluzio kopurua definitu
3:	while ez dago doituta
4:	Antzezlearen sare neuronalak hasieratu
5:	for i=iterazio kopurua
6:	Kapak gehitu
7:	end
8:	<i>Reinforcement Learning</i> entrenamendua egin
9:	Obserbazioak eta akzioa definitu
10:	Planta sortu
11:	Kritikoa sortu
12:	Agentea sortu
13:	Entrenamendua abiarazi
14:	Doikuntza ebaluatu
15:	end
16:	Lortutako agentea simulatu

Planteamendu hau aurrera eramanda, honako hau izan dira lortutako emaitza:



Irudia 37: Sarearen eboluzioarekin lortutako agentearen emaitza

Ikustenenez agentea ez da gai behar duen akzioa ikasteko. Entrenamendua ikertuz, antzematen da sari funtzioaren balioa nahiko konstante mantentzen dela. Sarearen edozein topologiarekin gertatzen da hau beraz, arazoa sakonagoa dela determinatu daiteke. Sariaren balioaren konstante mantentze horrek ikasten ez duela adierazi dezake, bide egokiek saria handitzeko joera hartu beharko zuelako eta bide alternatiboak bilatzen ez dituenaren seinale izan daiteke. Aipatzekoa da ere entrenamendu mota honek ez duela agente egokiena aukeratzen dela ziurtatzen, baizik eta baldintzak bete diren uneko agentea. Honek entrenamendu arbitrario bat adierazten du, egonkortasuna galduz.

2.2.1. ALGORITMO GENETIKO BIDEZKO OPTIMIZAZIOA

Sare neuronalak eboluzionatzeak ez du DDPG sistemaren entrenamenduan eragin handirik, planteatua izan den moduan. Ondorioz, sare neuronalak alde batera utziz, *Reinforcement Learning* sistemaren sari metodologia jarraitzen duen planteamendu bat garatu da. Honetarako algoritmo genetikoaren erabilera barneratzen da, eboluzio kontzeptutik ez aldentzeko. Kasu honetan kritikoa berriz ere funtzio bat bezala finkatzen da. Helburua *reward*-aren balioa maximizatzea da kritikoko balio altuena lortzeko.

$$G = \sum_{t=0}^{t=N} r(\overrightarrow{pos}(t), u(t)) = \sum_{t=0}^{t=N} r(\overrightarrow{pos}(t+1)) \quad (32)$$

Gogoratu, $\overrightarrow{pos}(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix}$ zela definitu zen aurreko ataletan. Helmugak eta posizioarekiko oztoporik gertukoak ere x eta y balio propioak dituzte eta hauek $\overrightarrow{Helmuga}$ eta $\overrightarrow{Ozt.Gertu}$. bezala izendatzen dira. Datu hauekin helmuga eta oztopo gertukoarekiko posizio erlatiboak kalkulatu daitezke, ondorengo ekuazioetan agertzen den erara.

$$\overrightarrow{pos}_{erl.}(t) = \overrightarrow{Helmuga} - \overrightarrow{pos}(t) \quad (33)$$

$$\overrightarrow{pos}_{erl.}(t) = \overrightarrow{Ozt.Gertu} - \overrightarrow{pos}(t) \quad (34)$$

Datu hauek eskuratzuz, uneko kokapenak helmugarekiko duen lerrokadura errorearen balioa kalkulatzeko aukera dago. Helmugarekiko posizio erlatiboa zenbaki konplexu eran definituz, lerrokadura angelua kalkulatzeko da, AV-ak hartu beharko duen norabidea dena.

$$\vartheta_{lerrokadura} = x_{posherl.}(t) + y_{posherl.}(t)i \quad (35)$$

Lerrokadura angeluaren balioekin jarraituz, ondorengo M matrizea definitzen da. Hau erreferentzia sistema aldatzeko erabiltzen da, absolututik ibilgailukora. Modu honetan, **Ekuzio (37)** eta **Ekuzio (38)**-ez baliatuz AV-ak hurrengo unean lortu beharreko posizioa estimatzen da:

$$M = \begin{pmatrix} \cos \vartheta & \sin \vartheta \\ -\sin \vartheta & \cos \vartheta \end{pmatrix} \quad (36)$$

$$posherl.(t + 1) = Mposherl.(t) \quad (37)$$

$$posoerl.(t + 1) = Mposoerl.(t) \quad (38)$$

Informazio honekin $u(t)$ akzioaren balioen planteamendua kalkulatzeko da. Hurrengo unean hartu beharko zuen posizioa helburuarekiko zuzenduta badao ibilgailuak $[-\pi, \pi]$ balioen artean dagoen akzio bat hartuko du, helmugarekiko unean duen posizio erlatiboa kontsideratuz. Gainera, helmuga oztopo batekin lerrokatuta badago ez da beharrezkoa izango norabidea aldatzea, betiere oztoppa atzealdean geratzen bada.

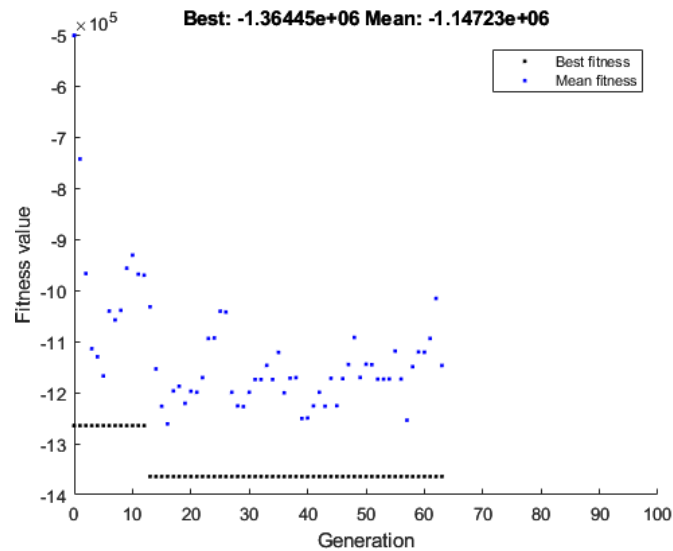
$$u(t) = atan \left(\frac{y_{posherl.}(t)}{x_{posherl.}(t)} \right) \quad (39)$$

Kontrako kasuan, hau da, desbideraketa baten beharra baldin badago, aurreko terminoari zuzenketa termino bat gehituko zaio, $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ balio artean dagoena. Kasu honetan estimatutako hurrengo posizioa hartuko da kontutan, bai helmugarekiko, bai oztopo gertuenarekiko. Zuzenketa honen inpaktua zein izan behar den aurreikusteko erabiltzen da algoritmo genetikoa. Ondorioz, K zuzenketa parametroaren bidez biderkatzen da bigarren termino hau, unitate gabekoa izanik.

$$u(t) = atan \left(\frac{y_{posherl.}(t)}{x_{posherl.}(t)} \right) + atan \left(K \left(\left(x_{posoerl.}(t + 1) - x_{posherl.}(t + 1) \right) \frac{x_{posoerl.}(t + 1)}{y_{posoerl.}(t + 1)} \right) \right) \quad (40)$$

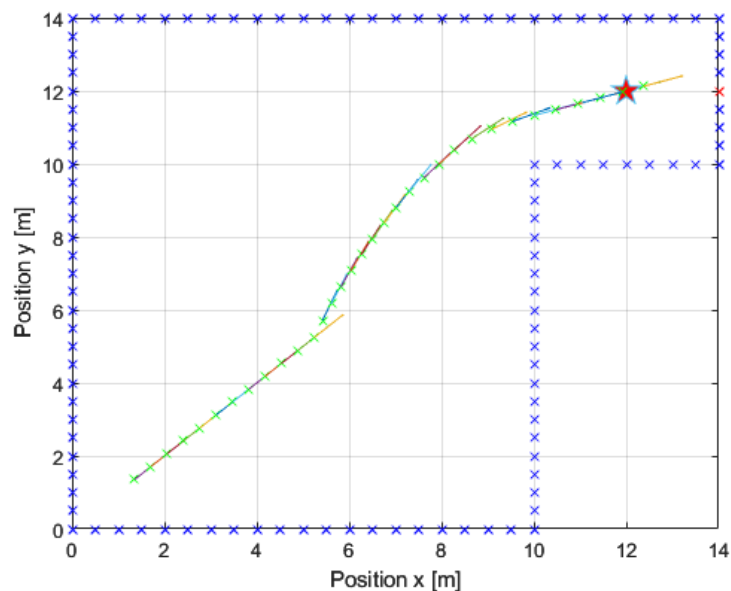
Ildo beretik K balioa mugatu egiten da, optimizazioa 0,01 eta 0,1 arteko balioa lortuz. Jarraituz, behin akzioa lortuta hurrengo urrats erreala kalkulatzeko da, aurreko ataletako **Ekuzio (4)** eta **Ekuzio (5)**-ean definitu zen bezala. **Ekuzio (8)**-arekin sariaren balioa kalkulatzeko aukera dago orain eta urrats guztietako sariaren balioa metatuz kritikoaren balioa lortzen da, saria maximizatzea lortuz.

Algoritmo genetikoak erakusten duen optimizazioa ondorengo irudian aurkezten da. Hemen, generazio bakoitzak hartzen duen doitzeko balioa adierazten da. Puntu beltzak garaiko doitzeko onena adierazten dute, urdinek batez bestekoa adierazten duten bitartean. Ikusten denez kromosomek lortzen duten zehaztasuna behera egiten du generazioak aurrera egin ahala, azkenean $-1,36445 \cdot 10^{-6}$ -ko doitzeko onena lortuz.



Irdia 38: Algoritmo genetikoaren bidezko K parametroaren doitzea

Bilakaera honek K -ren balio optimoa $K = 0,0774$ -an finkatzen du. Zenbaki honekin nabigazioa abiarazten da eta hurrengo irudian agertzen den bidea lortzen da [1,1] posiziotik abiaraziz.



Irdia 39: Algoritmo genetikoaren bidez lortutako nabigazioa

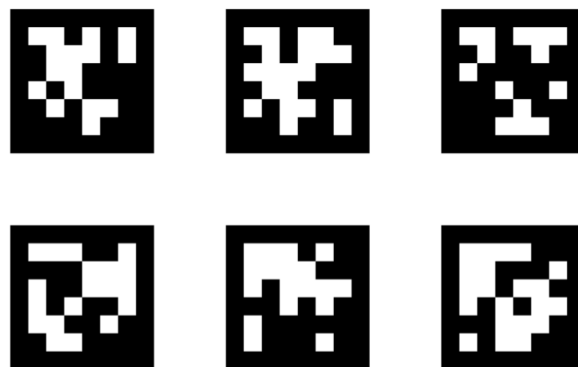
Antzeman daitekeenez bidea osorik kalkulatzeko gai da planteamendu honen bitartez. Kasu honetan, geziek AV-ak daraman norabidea adierazten dute aldiuneko, akzioaren eragina adieraziz. Horrela, talkarik gabeko nabigazioa egin daiteke ingurune batean, algoritmo genetikoek optimizatutako akzioaren eraginarekin.

3. SLAM Bisuala

Ibilgailu autonomoen funtzionamendu egokirako ezinbestekoa da kokatuta dagoen inguruko maparen informazioa eskuragarri izatea. Honetaz gain, nahitaezkoa da ere nabigazio inguru horretan AV-ari bere burua lokalizatzeko gaitasuna ematea. Ataza hauek, aurrez aipatu den bezala, SLAM teknikari dagozkio. Antzeman ahal izan den bezala, aurreko ataleko algoritmoa garatzeko mapa simple bat erabili da, baina irudikapen hau lan eremuaren baitakoa da. Ondorioz, inguruko mapa bat lortzeko sistema baten beharra dago, honetan lokalizatzeko metodo bat izateaz gain. Sare neuronalek jada pisua suposatzen dutenez kostu konputazionalerako, era analitikoan mapa lortzea egokiagoa dela ondorioztatu da.

Eginkizun honetarako AprilTag-etan oinarritzea erabaki da, informazioa lortzeko aurkezten duten era erraz eta azkarragatik. Inguruneke AprilTag bakoitzak (x, y) posizio propio bat izango du identifikazio zenbaki batez gain, hauen sailkapena egiteko aukera emanaz. Gainera, forma eta kolore detektagarriak dituzte, edozein ingurune konplexuan aurkitu ahal izanez. Forma karratua dutenez, erraz lortu daitezke hauen bertizeak ere lerroen gurutzaketetatik abiatuz.

Etiketa hauen miaketa egiteko kamera monokular baten erabilera soila planteatzen da robot mugikorrean instalatuta. Honetaz gain, inguruneke horma guztietan AprilTag-ak itsasteko beharra dago baina honek ez du kostu handirik suposatzen; Interneten erraz topatzen dira eta hauek inprimatzea besterik ez da behar. Oinarrian, etiketa hauen detekzioa kamera batek eskuratutako irudi batetik markatzailea erauztean datza, erpinen koordinatuak mugatuz eta hauen ertzak zehaztuz. AprilTag-aren posizio eta tamaina finkoa izango denez, aurreko informazio honekin kamerarekiko duten erlazioa kalkula daiteke. Dauden AprilTag familia guztietatik, MATLAB-ek prozesatzeko gaitasuna duen familietako bat aukeratu da, *Tag36h11* familia hain zuzen ere, hurrengo **Irudia 40**-an ikusi daitezkeen etiketez osatuta, beste askoren artean.



Irudia 40: *Tag36h11* familiako AprilTag ereduak

Kontuan izan beharrekoa da industria ingurune baten barnean, AV baten kokapen absolutu bat egiteko erreferentzia sistema absolutu bat aurkitu behar dela. AprilTag-ak zorizko distantzietan kokatuta daudenez ez dute erreferentzia hori finkatzen eta era erlatiboan soilik kontsideratu daitezke. Ondorioz, ibilgailuarekin detektatzen den lehenengo AprilTag-a erreferentzia absolutu bezala hartuko da eta miaketa prozesua aurrera doan heinean agertzen diren beste etiketak lehenengo honekiko erreferentziatuko dira. Beraz, AV-a miaketa mugimendu batean jarriko da inguruko mapa lortzeko, langileak gidatuta. Hau esanda, algoritmoa ondorengo **Taula 13**-an agertzen den fluxua jarraituko du.

Taula 13: Mapa sortzeko algoritmoaren fluxua

Inguruko maparen eraiketa algoritmoa	
1:	Kameraren kalibrazioa
2:	Kameraren parametro intrintseko eta estrintsekoak kargatu
3:	AprilTag absolutuak finkatu
4:	Markatzailea detektatu
5:	Berezko MATLAB funtzioaren bitartez etiketen informazioa lortu
6:	R biraketa matrizea XYZ Euler angeluetara pasatu
7:	Markatzailearen posizioa lortu kamerarekiko
8:	Kameraren posizioa erreferentzia sistema finkoan kalkulatu
9:	Ibilgailuaren posizio absolutua gorde
10:	for $i \leftarrow$ gainontzeko etiketa guztientzako (erlatiboak)
11:	Ibilgailua mugitu
12:	Markatzailea detektatu
13:	Berezko MATLAB funtzioaren bitartez etiketen informazioa lortu
14:	R biraketa matrizea XYZ Euler angeluetara pasatu
15:	Markatzailearen posizioa lortu kamerarekiko
16:	Kameraren posizioa erreferentzia sistema finkoan kalkulatu etiketa absolutuekiko
17:	Etiketen posizio absolutua gorde
18:	end
19:	AprilTag guztiak irudikatu
20:	Planoak izango balira maneiatu hauen intersektzioak bilatuz
21:	Mapa grafikatu

Behin mapa lortuta, lokalizazioa egiteko AprilTag-en detekzio soilarekin nahikoa izango da ibilgailua dagoen posizioa lortzeko. Ataza hau aurrera eramateko, garrantzi handia dauka kameraren zehaztasuna hau da, doitasun altua izateaz gain bereizmen egokia eta distortsio txikia izatea. Modu honetan, kamerak inguruneko edozein baldintzatan emaitza berberak lortu beharko ditu, aldaketa txikiak detektatuz, errealitatearekin bat etorritz. Esan beharra dago gainera, prozesu honetan erabiltzen den kamera *Logitech HD Pro C920* eredua dela.

3.1.Kameraren kalibrazioa

Behar den zehaztasuna lortzeko, lehenengo pausua kameraren kalibrazioa egitea da. Aplikazio honetan kamerak pisu garrantzitsuena jasotzen du beraz, argi dago, honen parametroak ezagutu eta kodigoan jarri behar direla. Hauek, kalibrazio-matrizeari eta kameraren distortsio-matrizeari dagozkio eta ezinbestekoak dira mundu errealean dagoen 3D puntu baten eta irudian dagoen 2D pixelaren arteko erlazio zehatza definitzeko.

Bi matrize hauek mota ezberdineko informazioa dute barneratuta. Alde batetik kamera sistemaren barne parametroak daude, hau da, parametro intrintsekoak, distantzia fokala, zentro optikoa eta lentearen errotazio distortsio koefizienteak jasotzen dituenak. Bestetik, kanpo parametro edo parametro estrintsekoak kameraren errotazio eta translazioak jasotzen ditu, erreferentzia sistema global batekiko. Informazio hau eskuragarria dela kontuan izanda, AV-aren posizioa lortu daiteke zehaztasunez edozein ingurunetan. Ondorengo **Ekuazioa (41)**-ak 2D irudiko (x_p, y_p) pixelaren eta

mundu erreala (X, Y, Z) arteko erlazioa deskribatzen du. Bertan f_x eta f_y , kameraren distantzia fokala errepresentatzen dute eta c_x eta c_y zentro optikoaren balioak, parametro intrintsekoak jasoz.

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (41)$$

Parametro estrintsekoek dagokion matrizea ohiko errotazio-translazio matrize bat da baina honen kalibrazio egokirako distortsioa kontsideratu behar da (x_d, y_d) pixel distortsionatuak izanda, honen parametro erradial eta tangenzialekin. Distortsio erradiala hurrengo erlazioekin zuzentzen da, non $r^2 = x_d^2 + y_d^2$ den eta k_1, k_2 eta k_3 zuzenketa koefiziente erradialak diren:

$$x_{erradiala} = x_d(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (42)$$

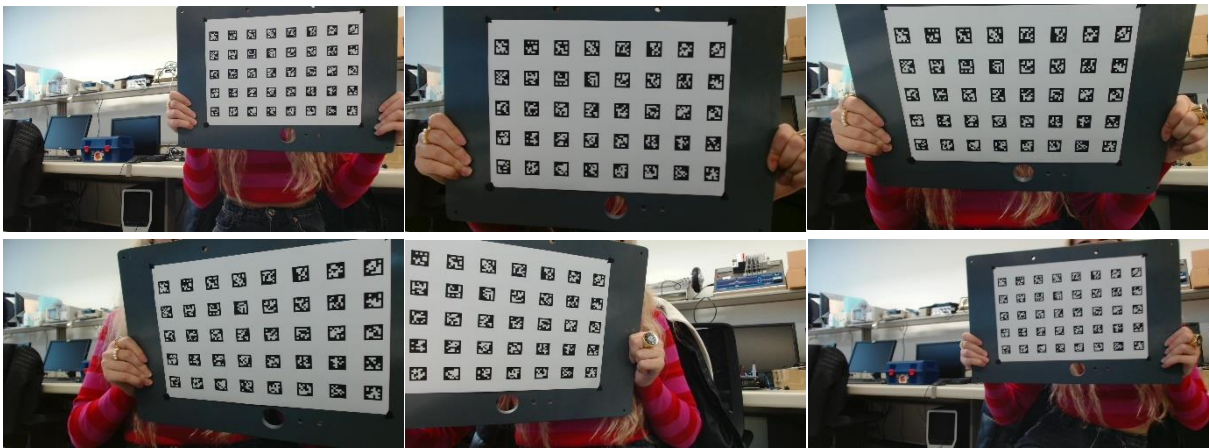
$$y_{erradiala} = y_d(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (43)$$

Distortsio tangenzialaren kasuan hurrengo formulak erabiltzen dira, non p_1 eta p_2 zuzenketa koefiziente tangenzialak diren:

$$x_{tangenziala} = x_d + [2p_1 x_d y_d + p_2 (r^2 + 2x_d^2)] \quad (44)$$

$$y_{tangenziala} = y_d + [p_1 (r^2 + 2y_d^2) + 2p_2 x_d y_d] \quad (45)$$

Era honetara distortsio koefizienteen matrizea lortzeko aukera dago baina era analitikoan egitea oso konplexua da. MATLAB plataformak eginkizun hau betetzeko tresna bat dauka. Hau *Camera Calibration* [419] aplikazioa da, *Computer Vision Toolbox*-aren barnean aurki daitekeena. Bertan, AprilTag-ek osatutako txantilo bat dago eskuragarri, eta kamera finko utziz patroia hau mugitu behar da irudiak ateratzeko. Hurrengo **Irudia 41**-an ikus daitezke helburu honetarako lortu eta erabili diren irudi ezberdinen adibideak.

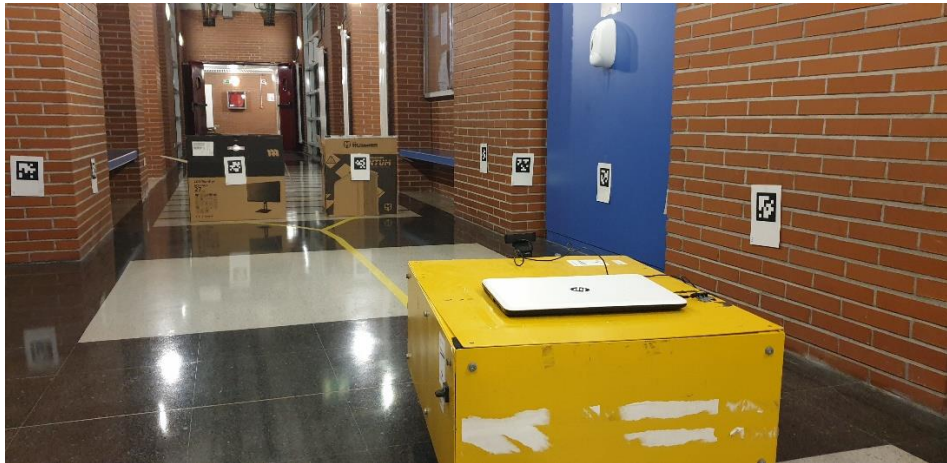


Irudia 41: Kamera kalibrazioarako AprilTag txantiloien irudiak

Txantilo honetan dauden AprilTag-ak, ezberdinak izateaz gain, erraz detekta daitezke irudi batean. Karratu forma dutenez, hauen erpinak kokapen samurra ahalbidetzen dute, bi norabidetan ebakitako gradienteak baitituzte. Honetaz gain, txantiloien lerroak elkarguneak adierazten dituzte, ertzak aurkitzeko ataza sinplifikatuz. Alderdi hauek, karratuen ertzak kalibrazio txantiloian sendotasunez kokatzeko aukera ematen dute. Guzti honekin, MATLAB-eko aldagai bat lortzen da, kargatzerakoan kameraren kalibrazio parametroak definituta geratuz.

3.2. Markatzaileen detekzioa

Ibilgailua martxan jartzerakoan AprilTag guztien miaketa egin behar du, langile batek gainbegiratuta. Algoritmoaren fluxu diagraman (ikusi **Taula 13**) hauteman ahal izan den bezala bi etapa nagusi daude. Lehenengo detekzioan identifikatzen diren etiketak erreferentzia absolutua finkatuko dute. Ondoren, hauen ostean datozen AprilTag-en erreferentziak lehenengo hauetara moldatuko dira mapa egokitasunez eraiki ahal izateko. Aipatzekoa da, mapan agertu nahi den horma oro egon beharko dela etiketatuta. Kasu honetan Vitoria-Gasteizko ingeniaritza eskolako pasilloetako gunea baten mapa eraiki da eredu bezala, **Irudia 42**-an agertzen den gunea hain zuzen ere.

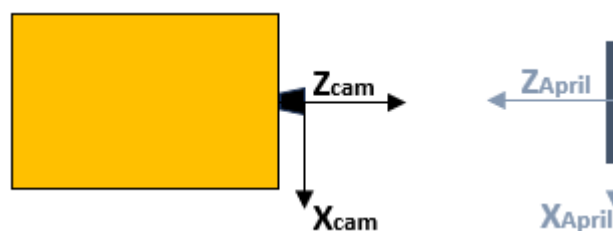


Irudia 42: Eskolako pasilloetan mugatutako gunea

Mapa hau eraikitzean beraz, ikaskuntza sakoneko algoritmoan aplikatu daiteke ingurune propioan agentea kalkulatzeko.

3.2.1. MARKATZAILE ABSOLUTUAK

Ibilgailu autonomoa martxan jartzerakoan, lehenengo etiketen detekzio bat egiten da. Markatzaile hauek absolutu bezala definituko dira eta horretarako aldaketa batzuen beharra dute. AprilTag-en detekzioa MATLAB-ek barneratzen duen *readAprilTag* funtzioaren bidez egiten da eta honek, informazio ezberdina eskuratzeko aukera ematen du. Informazioa, kameraren zentro optikoarekiko jasotzen dela jakina da baina kontuan izatekoa da etiketen eta ibilgailu propioaren erreferentzia sistemak ezberdinak direla, hurrengo irudian irudikatzen den bezala. Aipatzekoa da erreferentzia sistemen Y bektorea ez dela agertzen aplikazio honetarako alboratu egiten delako, AV-a ingurune lau batean erabiliko bait da eta etiketak altuera berdinean jarriko direlako. Nolanahi ere, lurrera zuzenduta doan bektore bat izango da. Bestalde, AV eta kameraren erreferentzia sistemak bat datoz norabideetan, ibilgailuarena honen zentroan kokatuta egonik.



Irudia 43: Kameraren eta AprilTag-en erreferentzia sistemen alderaketa

Antzeman daitekeen bezala, errotazio bat existitzen da bi sistemen artean eta hau funtzioak eskaintzen duen ohiko R errotazio matrizearen bidez konpontzen da, baina balio hauekin operatzeko Euler angeluetara transformatzeko beharra dago. Erreferentzia sistemen posizioari dagokionez, P translazio bektore bat eskaintzen da, etiketarekiko kamerak duen (x, y, z) balioekin. Datu hauekin beraz, AV-aren eta AprilTag-aren arteko posizio erlatiboa kalkula daiteke. Sei osagarri behar dira hain zuzen ere eta hauek, P -ren (x, y, z) parametroak eta R -ren *pitch* (θ), *roll* (Φ) eta *yaw* (Ψ) balioak dira. *readAprilTag* funtzioak kalkulatzeko duen transformazio homogeneoaren T matrizearen barneko parametroak dira guztiak eta hurrengo ekuazioan ikus daitekeen itxura dute.

$$T = \begin{bmatrix} R_{3 \times 3} & P_{3 \times 1} \\ 0 & 0 \end{bmatrix} \quad (46)$$

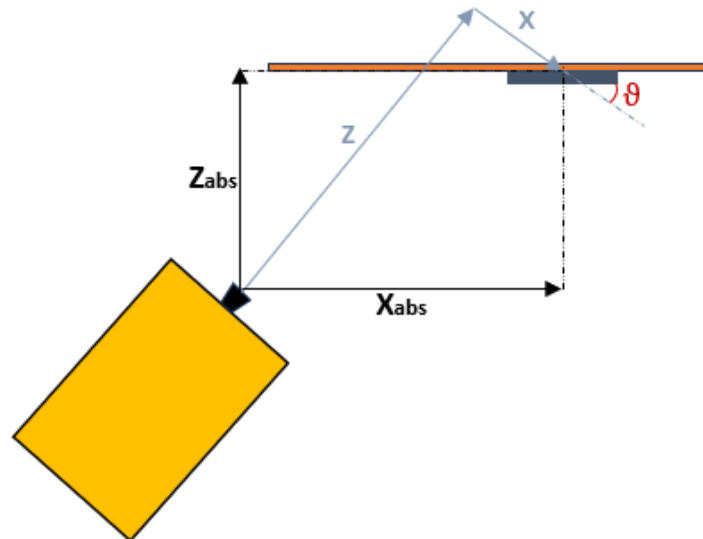
Era berean, kameraren orientazioa definitzeko dituzten parametroak hurrengo erlazioek definitzen dituzte, hauetako r balioak errotazio matrizearen terminoak izanik:

$$\theta = \arctan\left(\frac{-r_{31}}{\pm\sqrt{-r_{32}^2 + r_{33}^2}}\right) \quad (47)$$

$$\Phi = \arctan\left(\frac{r_{21}}{r_{11}}\right) \quad (48)$$

$$\Psi = \arctan\left(\frac{r_{32}}{r_{33}}\right) \quad (49)$$

Esan beharra dago kasu honetan soilik *pitch* (θ) angelua erabiliko dela, etiketak lauk diren hormetan itsatsita egongo direlako eta bai lurrarekiko, bai beste hormekiko 90° -ko angeluak osatuko dituztelako. Datu hauekin, AprilTag-en detekzioa planteatu daiteke. **Irudia 44**-ean irudikatzen dira kamerak etiketekiko irakurtzen dituen neurriak. Hauek, x , z , eta ϑ balioak dira. Azken hau, aipatu bezala, *pitch* angeluaren balioa izango da, Euler transformazioaren ostean.



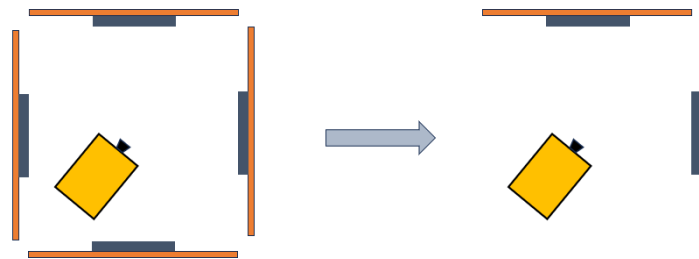
Irudia 44: AprilTag-en posizio erreferentzia absolutuaren kalkulua

Aplikazio honen interesa Z_{abs} eta X_{abs} balioak lortzean dago. Helburu horrekin hurrengo ekuazioak planteatzen dira, jakinda irudiak erakusten duen posizioan *readAprilTag* funtzioaren bidez lortutako balio guztiak positiboak izango direla.

$$Z_{abs} = z \cdot \cos \vartheta - x \cdot \sin \vartheta \quad (50)$$

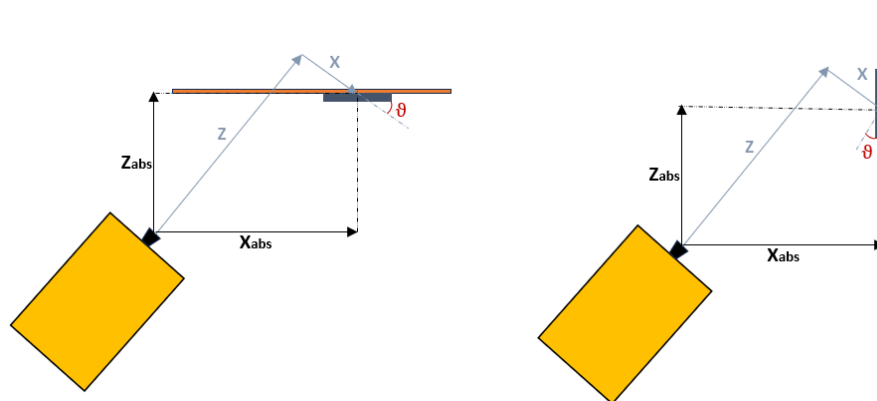
$$X_{abs} = z \cdot \sin \vartheta + x \cdot \cos \vartheta \quad (51)$$

Modu honetan eta lehenengo detekzioa dela gogoratuz, ibilgailuaren posizio absolutua finkatuko da. Hemendik aurrera detektatzen diren gainontzeko etiketak beraz hauekiko erreferentziatuko dira. Angeluak ere zehaztu behar dira, ibilgailuak ez duelako zertan beti norabide berdina eraman. Honek, AprilTag-en bisten aldaketa suposatuko du, ϑ -ren balioarekin bat etorritz. **Irudia 42**-an gainera antzeman daiteke, Apriltag-ak plano ezberdinetan kokatuta egongo direla. Gasteizko ingeniari-tza eskolaren kasuan, hormak 90º-ko alde dute bata eta bestearen artean beraz lau posizio ezberdinetan egon ahalko dira etiketak kokatuta, hurrengo irudian adierazten diren bezala. Interesa planoan dago eta ez etiketaren aurrealdearen posizioan, beraz bi egoratara sinplifikatu daitezke.



Irudia 45: Planoen sinplifikazioa

Plano aldaketa honek *readAprilTag* funtzioak egiten dituen irakurketetan aldaketak suposatzen ditu, ϑ balioan hain zuzen ere (ikusi **Irudia 46**). Honek **Ekuazioa (50)** eta **Ekuazioa (51)**-ak irakurketa okerrak ematera eramaten du. Hormak perpendikularrak direnez ekuazio horietan ϑ parametroari 90 balioa kentzearekin arazo hau konponduta geratzen da.

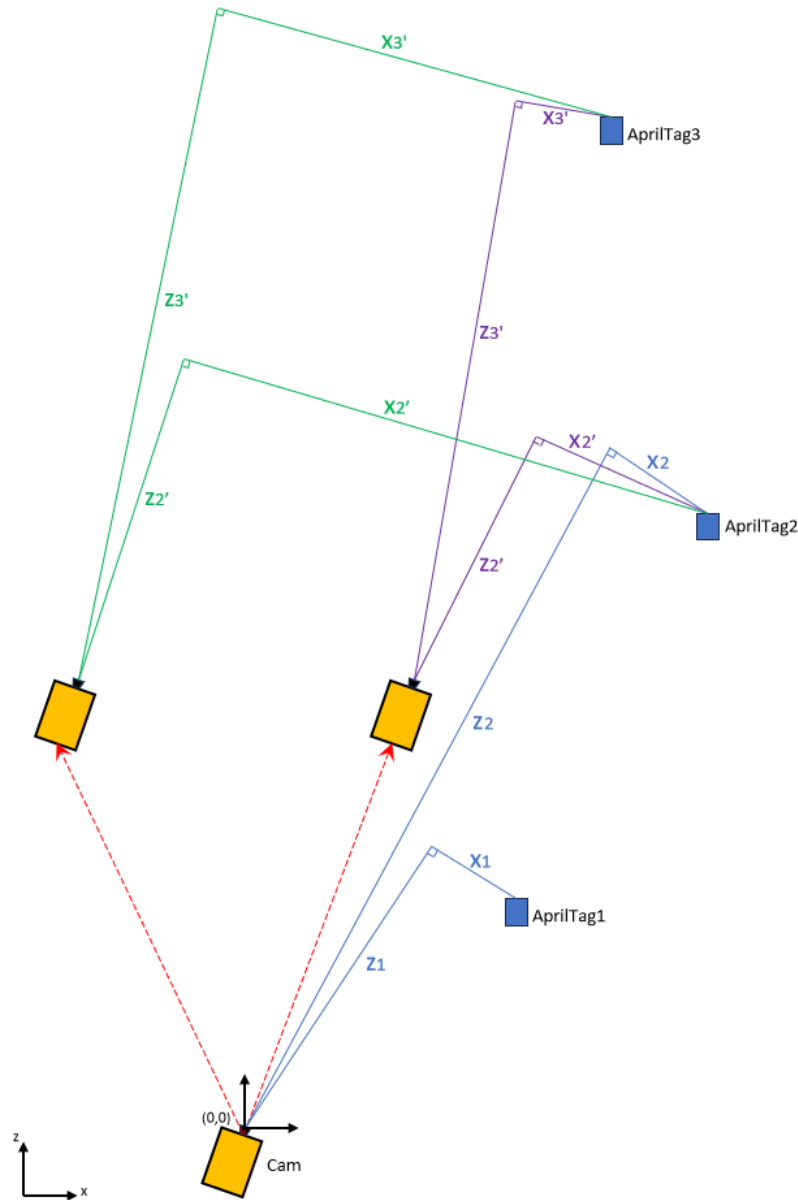


Irudia 46: AprilTag-en irakurketaren bariazioak

Planoen ildotik jarraituz, beste arazo bat suertatzen da. Etiketen irakurketetan ez dago etiketa era perpendikularrean kokatuta dagoela adierazten duen parametririk beraz nolabaiteko indikatzaile bat behar da 90º graduko ezberdintasun hori aplikatu behar zaien AprilTag-ak identifikatzeko. Etiketek dutena aldiz identifikazio zenbaki bat da, bakoitzeko propioa dena. Hauek zenbaki osoak dira eta hauetaz baliatzeko aukera dago. Ondorioz, plano berean dauden etiketek ID bakoitia izango dute eta 90º-tara daudenak aldiz, bikoitia.

3.2.2. MARKATZAILE ERLATIBOAK

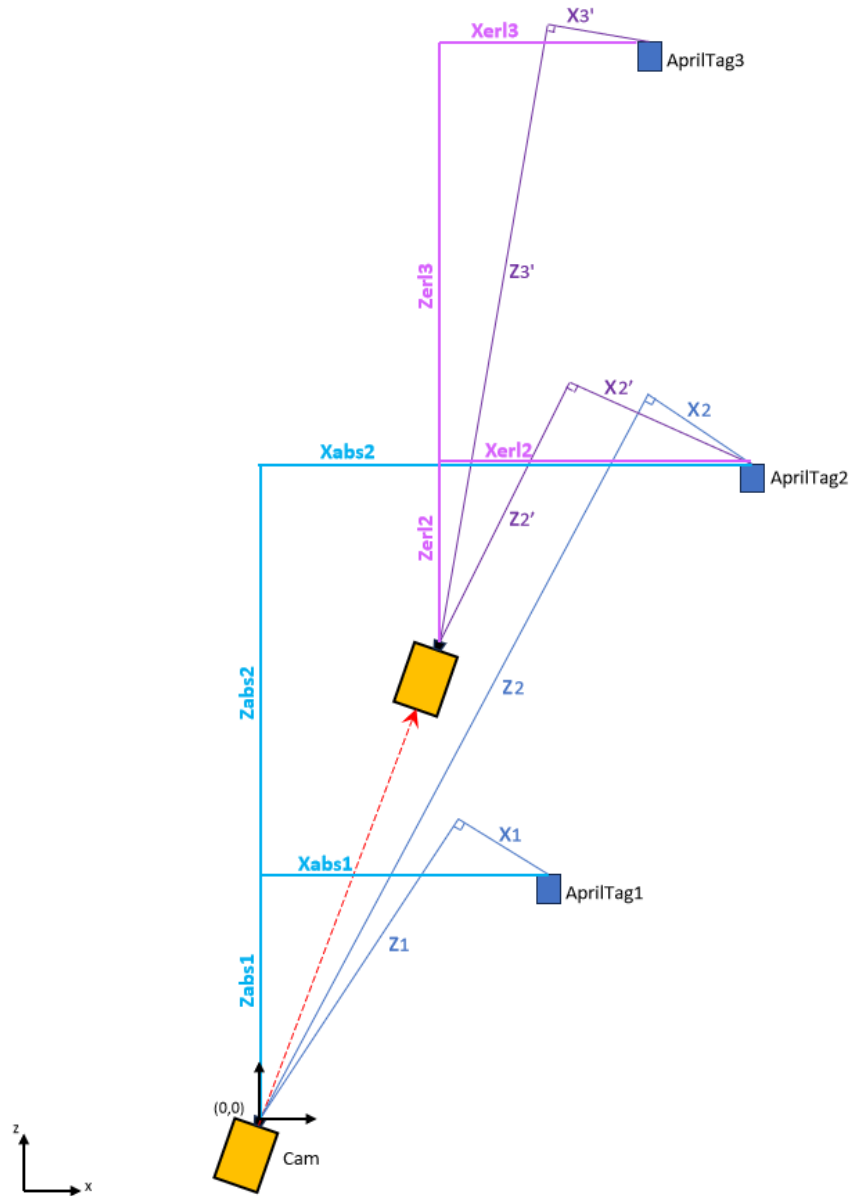
Lehenengo detekzioaren ostean, erreferentzia absolutuak definituta geratzen dira baina ingurune osoan AprilTag gehiago egongo dira sakabanatuta, mapa osatzeko helburuarekin. Ibilgailu autonomoak aurrera egiterakoan ez dauka zertan bide zuzen bat eraman eta hau, etiketen detekzioan kontuan izatekoa da. Hau jakinda hurrengo bi egoerak aurkezten dira: lehenik eta behin, ibilgailua aurrera egiterakoan hormara hurbiltzen deneko kasua, bigarrenik, etiketatik aldentzen deneko kasua. Zuzen jarraitzen dueneko egoera lehenengo kasuarekin parekatzen da.



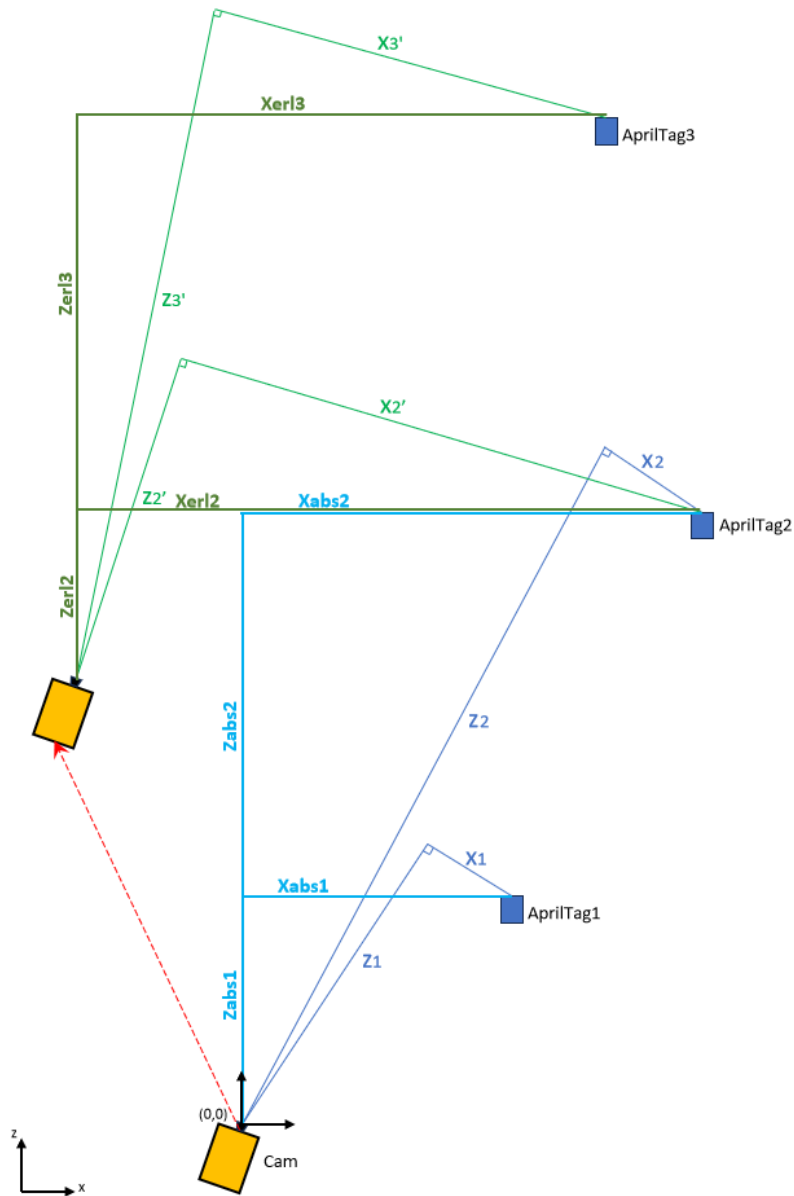
Irudia 47: Ibilgailua mugitzerakoan eman daitezkeen bi kasuak

Irudia 47-an, beheko aldean dagoen ibilgailuak lehen posizioari egiten dio erreferentzia. Kasu honetan bi AprilTag begiztatzen ditu kamerak, baina bakar bat begiztatzearekin ere nahikoa da, edota bi baino gehiago. Lerro urdinek “AprilTag1” eta “AprilTag2”-rekiko neurtuko diren distantziak errepresentatzen dute era erlatiboan, hau da, (x_1, z_1) eta (x_2, z_2) , **Ekuazio (50)** eta **Ekuazio (51)**-ekin absolututara pasa aurretik. Robot mugikorrek aurrera egingo du eta hormatik alendu edo gerturatu daiteke, irudian gezi gorriek norabide hauek irudikatuz. AV-ak aurrera egiterakoan beraz etiketa berriak ikusteko aukera izango du, esaterako “AprilTag3” baina oso garrantzitsua da aurrez

detektatutako AprilTag bat gutxienez bistan mantentzea, neurrien konparazioa egin ahal izateko. Posizio berri hauetan irakurketak (x_2', z_2') eta (x_3, z_3) izango dira, hauetako lehenengoa jada egindako irakurketa baten beste balio bat izanda, posizio ezberdin batetik, alderaketa egiteko. Hurrengo bi irudiek bi mugimenduen xehatze bat aurkezten dute, **Irudia 44**-ean planteatu diren erreferentzia aldaketekin.



Irudia 48: Ibilgailua hormara hurbiltzen deneko erreferentzia aldaketak



Irudia 49: Ibilgailua hormatik aldentzen deneko erreferentzia aldaketak

Aipatu beharra dago, bi irudietan (x_{erl2}, z_{erl2}) eta (x_{erl3}, z_{erl3}) kamerak irakurtzerakoan egingo duen eraldaketa dela baina hauek erreferentzia absolutura pasa behar direla. Aldakuntza hauek bi kasuetarako planteamendu berdina jarraitzen dute eta eragiketa berdina dute oinarritzat. Aurreko atalean oinarrituz eta kasuan kasu, erlazio hauek eratzen dira:

$$AprilTag1 = (x_1, z_1) \rightarrow AprilTag1_{abs} = (X_{1abs}, Z_{1abs}) \quad (52)$$

$$AprilTag2 = (x_2, z_2) \rightarrow AprilTag2_{abs} = (X_{2abs}, Z_{2abs}) \quad (53)$$

Esleipen hau lehenengo irakurketan egiten da, hau da, markatzaile absolutuak definitzerakoan. Ibilgailuarekin aurrera egiten da, esan bezala, hauetako bat bistan mantenduz eta gutxienez AprilTag berri bat bistaratu arte. Hauen balioak aurreko ekuaziotan bezala gordetzen dira, **Ekuazio (50)** eta **Ekuazio (51)**-ei jarraituz.

$$AprilTag2' = (x_2', z_2') \rightarrow AprilTag2_{erl} = (X_{2erl}, Z_{2erl}) \quad (54)$$

$$AprilTag3 = (x_3, z_3) \rightarrow AprilTag3_{erl} = (X_{3erl}, Z_{3erl}) \quad (55)$$

Absolutuen datu bakarria irakurketa bikoitza izan duen AprilTag-ari esker lortzen da eta hauen arteko neurrien diferentzia kalkulatzeko da **Ekuazio (56)** eta **Ekuazio (57)**-etan erakusten den modura:

$$Difx_{abs_erl} = X_{2abs} - X_{2erl} \quad (56)$$

$$Difz_{abs_erl} = Z_{2abs} - Z_{2erl} \quad (57)$$

Absolututan ez dagoen bakarria AprilTag berria da, irudien kasuan "AprilTag3"-a, beraz honen transformazioa egin eta gorde behar da, hurrengo erara:

$$AprilTag3_{abs} = ((X_{3erl} + Difx_{abs_erl}), (Z_{3erl} + Difz_{abs_erl})) \quad (58)$$

Azken operazio hau AprilTag berri guztiei aplikatuko zaie. Nahiz eta eragiketa sinpleak izan, bidean arkitutako etiketa guztiak, lehenengo detekzioan finkatutako erreferentziekiko moldatzeko aukera ematen dute. Behin uneko etiketa guztiak neurri absolututan gordeta daudenean AV-ak aurrera egin dezake berriz ere neurrien eraldaketa prozesua errepikatuz, AprilTag berririk ez dagoen arte.

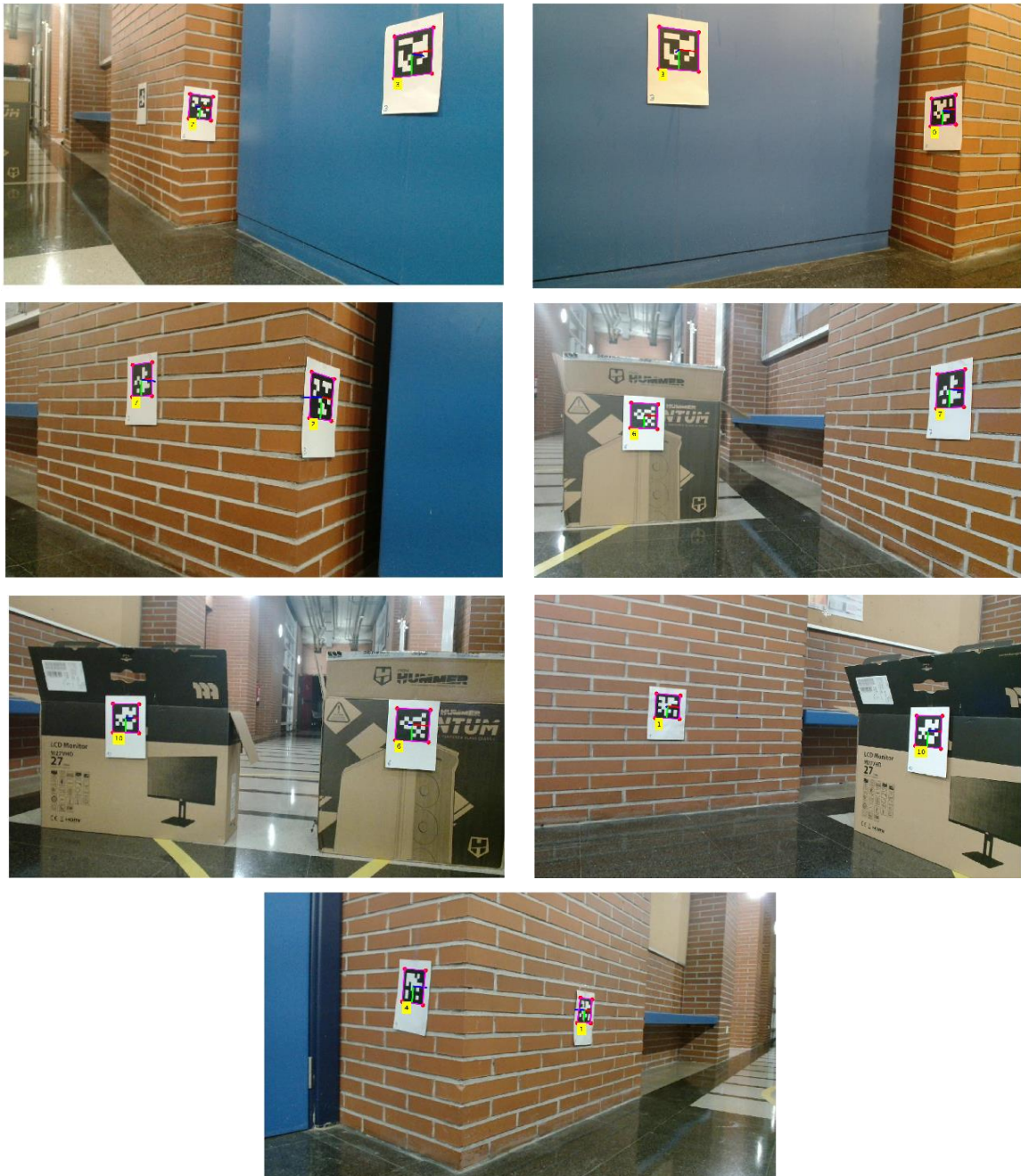
3.2.3. MAPAREN ERAIKITZEA

Esan bezala lehenengo detekzioan irakurtzen diren AprilTag-ek erreferentzia sistemaren zeroa definituko dute, hau ibilgailua mugimendua hasten den posizioarekin bat etorritik. Kasu honetan, lehenengo detektatu direnak ID 3 eta 5-a duten AprilTag-ak izan dira, plano berean egonik biak. Hurrengo irudian ikus daiteke MATLAB-en bidez etiketei egite zaien marka, hauen erreferentzia ardatzak irudikatuz. Aipatzekoa da, irudien kalitatea igotzeko kamerari 1920x1080-ko bereizmena esleitu zaiola.



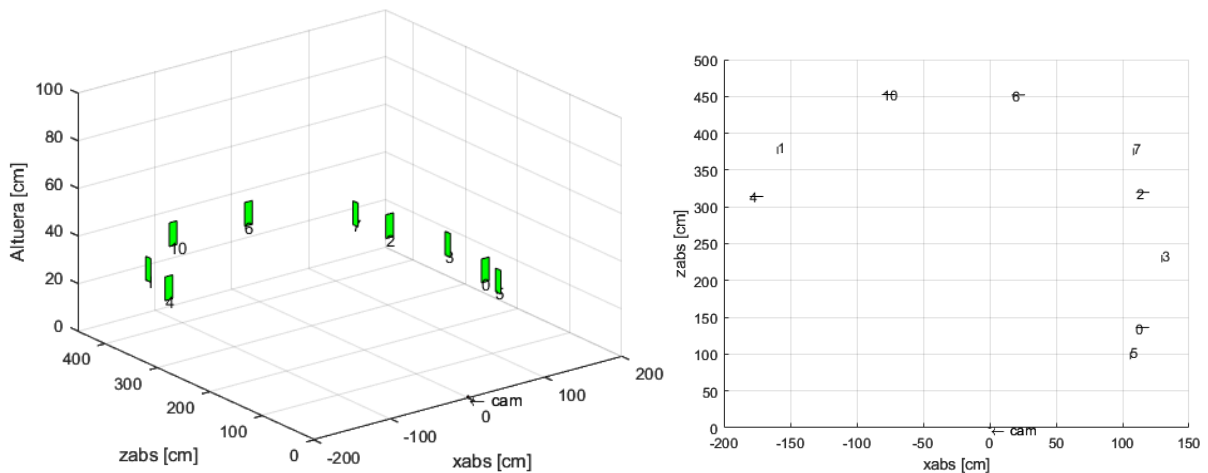
Irudia 50: Pasilloko AprilTag-en lehenengo detekzioa

Hurrengo detekzioetarako ibilgailu autonomoa aurrerantz abiarazten da betiere gogoratuz aurreko detekzioetako AprilTag bat gutxienez kameraren bistan izan behar dela. Horrela izanda, datozenak dira lortutako irudi guztiak, ordenean.



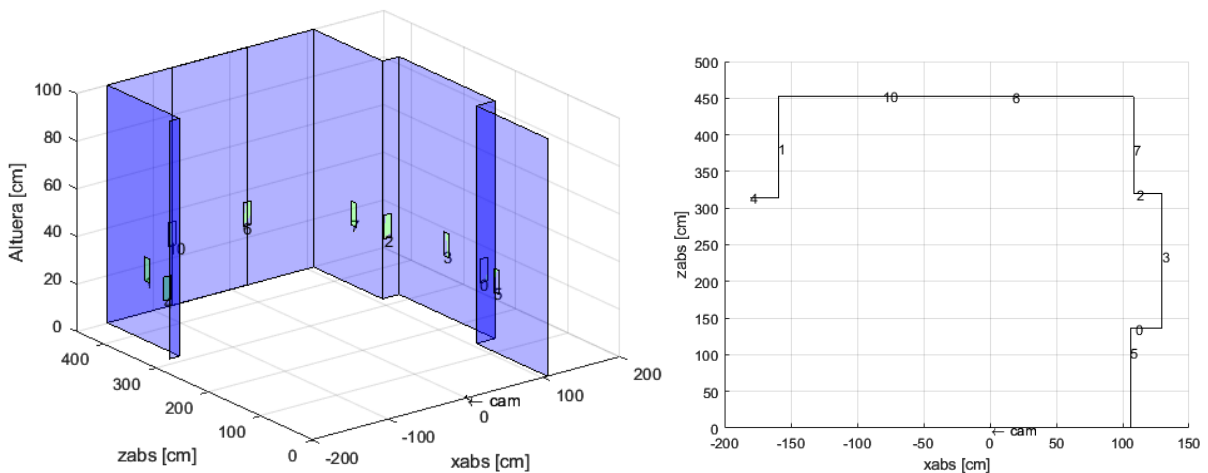
Irudia 51: Pasilloko gainontzeko AprilTag-en detekzioa

Behin AprilTag guztiak detektatuta daudela eta hauen posizio guztiak erreferentzia absolutuarekiko kalkulaturik daudela, inguruko mapa sortzeko aukera dago era birtualean. Lehenik eta behin, AprilTag-guztien irudikapena erakusten da, ingurunean duten kokapenarekin.



Irudia 52: AprilTag-en irudikapena espazioan

Ikusten denez AprilTag guztiak era egokian kokatuta daude dagozkien posiziotan. Ondoren, etiketa hauen planoetan dagoenez interesa, haiek luzatzeko aukera dago AprilTag-en arteko intersekzioetaraino eta modu horretan, inguruko mapa bat osatuz. Hau **Irudia 53**-an antzeman daiteke.



Irudia 53: Inguruko maparen irudikapena

Ibilgailua dagoen inguruko mapa birtual baten sorrera lortzen da operazio hauek aurrera eramanda. Mapa hau beste aplikazioetarako erabiltzeko aproposa da gainera, guneko limiteak markatzen bait ditu, edozein gunetan eraikigarria izateaz gain. Emaitzak aztertzeko, **Taula 14**-a aurkezten da. Bertan, AprilTag bakoitzeko identifikazio zenbakian agertzen dira, hauetarako kalkulatu diren distantzia absolutuez gain. Horretaz aparte, metro batekin eskuz neurtutako distantziak ere adierazten dira, konparaketa erara nahiz eta hauen neurketak oso zehatzak ez izan.

Taula 14: Kalkulatutako neurriak eta errealitatean neurtutakoak

AprilTag identifikazio zenbakia	Xabs [cm]	Zabs [cm]	X errealia [cm]	Z errealia [cm]
5	105,81	97,96	107	97
0	114,96	136,20	115	145
3	129,67	229,92	127	231
2	115,15	319,86	115	315
7	108,23	374,99	107	370
6	21,59	452,12	30	450
10	-76,03	452,65	-67	450
1	-159,59	376,93	-146	385
4	-175,40	314,19	-170	315

Antzeman daitekeenez errealitatean neurtutako balioak eta ikusmenaren bidez lortutakoak ordena berekoak dira. Lehenengo detekzioan egiten direnak zehatzenak dira baina, nabari da detekzioak aurrera egin ahala prezisioa galtzen doala, hau da, prezisio galera akumulatu egiten da. Akatsik handienak x ardatzean aurki daitezke, errorerik handiena ID 1 duen AprilTag-en aurkituz. Batez ere errore hau zeinu negatiboa dutenetan da nabaria. Honetaz gain, SLAM-a gauzatzeko sistemaren simple eta eraginkorra da eta beste ibilgailu autonomoen sistemekin guztiz bateragarria da.

4. Oztopen Distantzien Kalkulua

Ibilgailu autonomoen sistemek segurtasun metodo bat barneratuta izateak garrantzi handia dauka. Nabigazioa librea izango denez, AV-ak bere ibilbidean gurutzatuko den edozein oztopen distantzia jakiteak, arriskuak ekiditeko metodo bezala erabili daitezke. Horrela, distantziaren araberako akzioak hartu daitezke, hala nola, abiadura murrizketak, biraketak edota geldialdiak.

Lan hau burutzeko, bi sare neuronal sakonen erabilera planteatzen da, bata diseinatua eta bestea berrerabilia. Sareetako lehenak inguruko sakontasuna kalkulatzeko gai izango da, bigarrenak oztopenak detektatzeko lana egiten duen bitartean. Sareek entrenamendua behar dutela jakina da eta horretarako datuak. Ondorioz, sarearen beraren diseinuan baino gehiago, ataza honen konplexutasuna, entrenatzeko jasotzen den informazioaren kudeaketan dago. Informazio hori, eskuarki, bai kamera monokular batek, bai LiDAR sentsore batek jasotzen dute, sakontasun sarearen kasuan bi datu motak behar izanik adibidez. Kasu honetan, LiDAR-ak, distantziak ezberdintzen dituen puntu-hodei batean aurkezten du informazioa eta horrek prozesatze bat eskatzen du sarea entrenatu aurretik. Behin bi sareak entrenatuta daudela, AV-ak kamera bakarrik behar izango du eta irudiekin bakarrik maneiaturako da oztopen distantzia lortzeko.

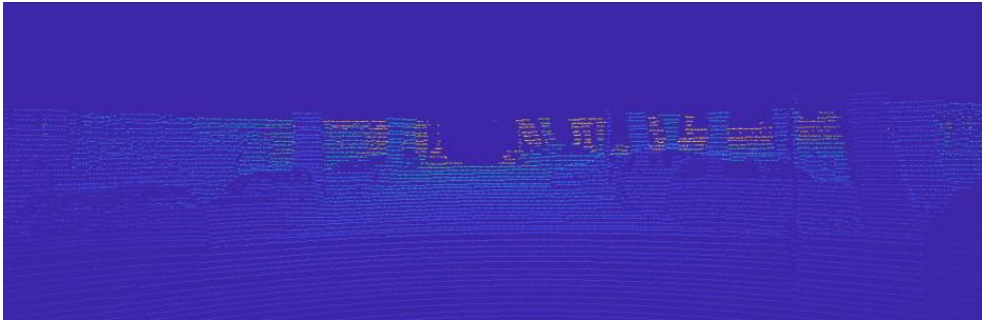
4.1. Datu-basearen sortzea

Bibliografiaren analisisian antzeman daitezkeen bezala, oso zaila da barneko inguruneetako datu-baseak lortzea, batez ere industria arlokoak. Datu hauek orokorrean izaera pribatukoak izaten dira eta ez daude eskuragarri aplikazioa industrian bertan martxan jartzea eskatzen duten arte. Beraz, ikerketa taldeek eskaintzen duten datu-baseetatik abiarazi behar da eta, kasu honetan, KITTI *Dataset*-a erabiltzea erabaki da. Ataza desberdinak burutzeko irudi kopuru handia dute eskuragarri eta interesekoa den *Depth Prediction* atalak, RGB irudiak eta LiDAR sentsoreak jasotako puntu-hodeiak biltzen ditu, bikoteka. Hauen eredu bat ondoren dagoen **Irudia 54**-an ikusi daitezke.



Irudia 54: KITTI datu-basearen irudia eta puntu-hodeia

Ohartu daitezke erabiltzen duten kamerak angeluar handiko objektibo bat duela, ohikoak diren kameretatik ezberdinduz. Beraz, dimentsio horietako irudirik ezin izango dela lortu jabetu ahal da, betiere ingeniariak eskolan eskuragarri dagoen *webcam*-arekin. Aipatzekoa da berriro ere, erabiltzen den kamera *Logitech HD Pro C920* eredu dela. Zehaztuz, KITTI-an dauden irudiek 352x1216-ko bereizmena dute eta ibilgailu autonomoan kokatuta dagoen kamerak 480x640koa du parametrizatuta. Beste alde batetik, LiDAR-ak jasotzen duen informazioak bereiztasun bat ere aurkezten du. Hurrengo **Irudia 55**-an ikus daitezkeen bezala, distantziak koloreztatuta dituen irudian, goiko eremuak datu galera bat aurkezten du.

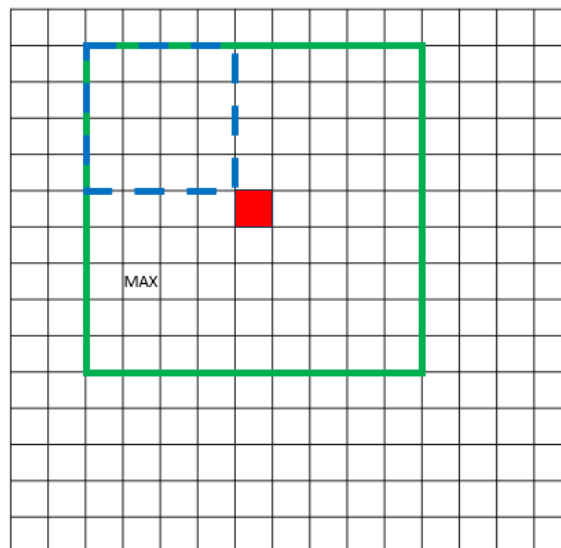


Irudia 55: LiDAR distantzia puntu-hodeia koloreztatuta

Informazio galera hau sentsoreak jasotako irudi guztietan gertatzen da, sare neuronalaren entrenatzeko arazo batean bihurtuz. Entrenamenduarentzako beste konplikazio bat, antzeman daitekeen bezala, irudiak puntuaren artean informazio falta duten gune gehiago erakusten dituela da, datuekiko irudi osatugabea izanik. Ondorioz, datu-basearen prozesatzearen beharra nabaria da. Irudien tamaina egokitu aurretik, puntu-hodeien leuntze bat egin behar da, datuak lainoaren %90-a baino gehiago izan daitezen, informazioa osatuz.

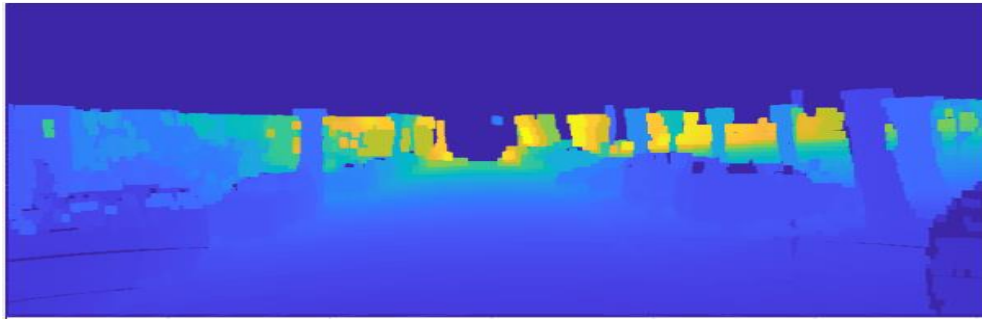
4.1.1. DATUEN LEHUNTZEA ETA IRUDIEN EGOKITZEA

Azken finean, puntu-hodeien irudia matrize handi bat da eta 0 balioa duten gelaxka asko ditu, informazioa ez dagoela adieraziz. Hauek betetzeko helburuarekin, “auzokidearen” teknikan oinarritu daiteke, tamaina finkatu bateko “leiho” bat eraikiz. Kasu honetan, leiho hori 4 balioa izatea erabaki da. Modu honetan, gelaxka bat leihoaren tamaina duten karratuz inguratzen da eta 9x9 tamaina duen karratu bat osatzen zaio inguruan, **Irudia 56**-an antzeman daitekeen bezala. Gorriz, n-iterazioko gelaxka irudikatuta dago, urdinez leihoaren tamaina eta berdez, inguruan sortuko den karratu osoaren eremua.



Irudia 56: Auzokide teknikaren bidez sortutako gelaxken irudikapena

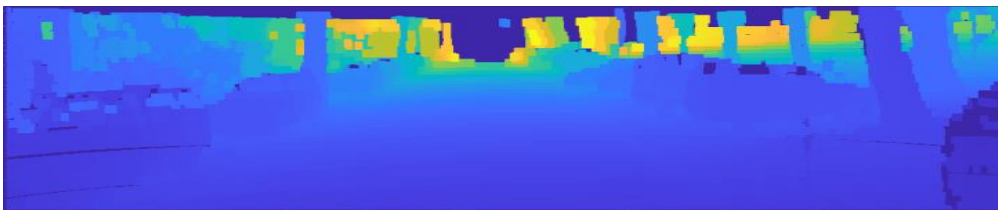
Behin gelaxkaren inguruko eremua finkatuta dagoela, bertako balio maximoa topatzen da eta gelaxkari balio hori esleitzen zaio. Prozesu hau irudian dauden gelaxka guztiekin egiten da, datuen osaketa bat lortuz eta emaitzatzat hurrengo irudia lortuz.



Irudia 57: LiDAR irudia leunduta

Ikusten den bezala, informazioz osaturiko irudi bat lortzen da. Egia da lausoa dela eta bertan irudikatuta dauden formak bereiztea zaildu egiten dela baina entrenamendua egiteko aproposa da. Hala ere, goiko gunea informazio gabekoa izaten jarraitzen du beraz, ez da erabilgarria eta irudiaren tamaina egokipenean kontuan izan beharko da.

Jarraituz, bigarren pausua datu-basearen irudien dimentsio egokipena izango da, eskuragarri dagoen kamerarekin bateragarriak izan daitezten eta exekuzioaren momentuan arazorik ez suertatzeko. Esan bezala, KITTI datuen bereizmena eta ibilgailuaren kamerarena ezberdina da. Lehenengo ezabaketa, aipatu den bezala, goiko aldeko guneari dagokio, duen informazio gabeziaren ondorioz. Baztertze hau egitean irudia 240x1216-ko bereizmenarekin geratzen da, **Irudia 58**-an agertzen den bezalako irudiak lortuz.



Irudia 58: Datuen goiko partearen ezabaketa

Bazterketekin jarraituz eta zabalera desberdintasunean oinarrituz, argi dago AV-ko kamerak ezin izango dituela KITTI irudietan agertzen diren alboko gunek lortu, ibilgailuan kamera aurrerantz zuzenduta bait dago. Funtsean, gune zentrala bakarrik eskuratu ahal izango da, irudiaren bi zati gehiago baztertuz. Erabiltzen den kameraren bereizmenaren proportzioa 4/3-ekoa dela kalkulatu daiteke. Hau jakinda eta irudiak uztargarriak izan behar direla kontuan hartuz, altueraren 240 pixelak kontsideratuz, zabalera lortu daiteke. Kasu honetan irudiaren zabalera 320 pixeletakoa izan beharko da, 240x320 bereizmeneko irudiak lortuz. Ondorioz, irudien erdi guneko 240x320 tamaina duen zati batekin lan egingo da. Horrela, entrenamendu eta exekuzioko datuen arteko erlazioa lortzen da. Kontuan izan behar da egokipen hau bai KITTI-ko RGB irudiei, bai LiDAR sentsorekoei aplikatu behar zaiela.

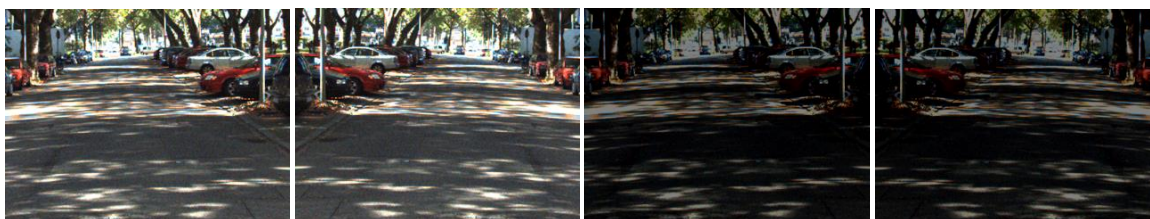
4.1.2. DATU BASEAREN OSAKETA

Aurreko ataleko prozesamenduaren ostean **Irudia 59**-ean ikus daitekeen irudi-bikoteekin amaitzen da. Hauen tamaina sarearen diseinuan kontuan izango da beraz, industriako irudi errealak lortzerako orduan ez da arazorik egongo bateragarritasunarekin.



Irudia 59: KITTI datu-basearen irudia eta puntu-hodeia

Datu-base honek, sakontasunaren kalkulurako 1000 irudi-bikote soilik eskaintzen du. Aplikazio honetarako kopuru hori oso murriztua da eta datu-basearen datu kopuruaren areagotze baten beharra sortzen da. Hau, *data augmentation* teknikari dagoi eta jatorrizko irudian aldaketak egitean datza hauen kopurua handitzeko. Eraldaketa hauek, nahiz eta sinpleak izan, sareak era ezberdinean tratatzen ditu, honen ikaskuntza hobetuz. Helburu honekin, lehendabizi irudiei *flip* edo irauli bat aplikatuko zaie. Horrela 2000 irudi-bikote lortuko dira. Behin hau eginda, RGB irudiei soilik distira eta kontraste parametroak aldatuko zaizkie, parametro hauen balioak 0,2 eta 0,5-ean finkatuz hurrenez hurren. Hauek, dagokien puntu-hodeiekin parekatuko dira berriro, guztira 4000 irudi-bikotez osatutako datu-basea sortuz. Eraldaketa hauek hurrengo irudian ikus daitezke.



Irudia 60: Datu-basea osatuko duten irudi motak

Datu kopuru honekin, sareak nahikoa izan beharko luke entrenamendu egoki bat egiteko. Gainera, irudi-bikote guzti hauetatik, 3700 entrenamendurako gordeko dira eta gainontzekoak balidazio partean erabiliko dira. Azpimarratu beharra dago, testatzea aurrera eramateko, irudi orijinaletatik 5 gordeko direla proban jartzeko, beraz entrenamendua 3695 iruditakoa izaten amaituko du. Aipatzekoa da ere, LiDAR sentsoreak lortutako irudiak maneiatu egin behar direla lehendabizi distantzia informazioa unitate ulerkor batean emateko. Hau eragiketa sinple baten bidez egiten da; datu gordinak *uint16* formatuan datorren informazioa daukate eta metro unitatean lortzeko *float* formatura pasa behar dira 256.0 balioaz zatituz. Eragiketa honen ostean, 0 balioa duten pixelei 100 [m]-ko balioa ematen zaie, definituta ez zeudenak guztiz finkatzeko, orokorrean urrun dauden datuak bait dira antzeman delako.

4.2.Sare neuronalaren diseinu eta entrenamendua

Distantziak kalkulatzeko sare bat diseinatzeko ataza aurkezten da. Behin datu-base egoki bat izanda, sare neuronal bat sortzeko aukera dago, zeinetan eskuragarri duen informazioaz baliatuz objektuen distantzia kalkulatzeko ikasten den. Modu horretan, ibilgailu autonomoa lantegian martxan dagoenean, irudiak lortzeko gai izango da eta segurtasuna bermatzeko akzioak hartu ditzake. Hurrengo **Irudia 61**-an auteman daiteken sare neuronal sakonaren egitura proposatzen da. Zenbait kapa ezberdinez osatuta dago, bakoitza bere eginkizunarekin.



Irudia 61: Distantziak kalkulatzeko sarearen egitura

Lehenengo kapa sarrera da, hau da, RGB irudia jasoko duen geruza 240x320x3-ko tamainarekin, kanalak ere kontuan izan behar bait dira. Ondoren 4 geruzaz osatzen diren bi multzo bereiz daitezke, bata bestearen ostean. Lehenengo multzoaren konboluzio geruzan 3x3 tamaina duten 8 filtro aplikatzen dira, *padding* teknika mantenduz. Hurrengo geruza *Batch Normalization* funtzioari dagokio, sarea bizkorragoa eta egonkorragoa egiteko sarreraren eskalatzeen aldaketan aurrean. *ReLU* geruza dator ostean, gradiente beherakorrek azkarrago bat egin dezan. Azkenik, *Max Pool* geruza, 2x2ko leihoekin ezaugarrien balio maximoa lortzeko. Berriz ere lau geruza hauen egitura errepikatzen da, oraingoa konboluzioa 16 iragazkiekin eginik.

Konboluzioa errepikatzen da “conv_3” geruzan, “conv_2” geruzako iragazkiak bikoiztuz, hau da 32 erabiliz. *Batch Normalization* eta *ReLU* geruzek jarraitzen dute 500 irteera duen *Fully Connected* geruzari pausu emanez, bere *ReLU* funtzio bikotearekin, nodo guztiak konektatuz. 1000 irteera dituen beste FC geruza bat aplikatzen da *ReLU*-arekin jarraituz eta, beste behin, 76800 irteera bituen “fc_3” eta “relu_6” geruzak aplikatzen dira. Azken *Fully Connected* hauen irteera kopurua, irudiak dituen ezaugarrien kopuruari dagokio, honen dimentsioak biderkatuz lortzen den balioari hain zuzen ere.

Azken geruza, “layer” izenekoa, berezko diseinua duen funtzio geruza bat da. Kontuan izan behar da RGB irudi baten dimentsioak [240,320,3,1]-koak direla eta *SSCB* formatua dutela, hala nola, *Spacial, Spacial, Channel* eta *Batch*. Irteera aldiz ez da RGB irudi bat izango beraz, egokitzen den geruza baten beharra dago *SSB* formatura pasako duena, hau da, *Channel* elementua galduko duena, baina ezaugarri kopurua mantenduko dituen. Hurrengo taulak, aurreko parrafoetan adierazitakoaren laburpen bat biltzen du.

Taula 15: Sare neuronalaren ezaugarriak

Geruzaren izena	Mota	Aktibazioak	Propietate Ikasgarriak
imageinput	Image Input	240(S) × 320(S) × 3(C) × 1(B)	-
conv_1	2-D Convolution	240(S) × 320(S) × 8(C) × 1(B)	Weights 3 × 3 × 3 × 8 Bias 1 × 1 × 8
batchnorm_1	Batch Normalization	240(S) × 320(S) × 8(C) × 1(B)	Offset 1 × 1 × 8 Scale 1 × 1 × 8
relu_1	ReLU	240(S) × 320(S) × 8(C) × 1(B)	-
maxpool_1	2-D Max Pooling	120(S) × 160(S) × 8(C) × 1(B)	-
conv_2	2-D Convolution	120(S) × 160(S) × 16(C) × 1(B)	Weights 3 × 3 × 8 × 16 Bias 1 × 1 × 16
batchnorm_2	Batch Normalization	120(S) × 160(S) × 16(C) × 1(B)	Offset 1 × 1 × 16 Scale 1 × 1 × 16
relu_2	ReLU	120(S) × 160(S) × 16(C) × 1(B)	-
maxpool_2	2-D Max Pooling	60(S) × 80(S) × 16(C) × 1(B)	-
conv_3	2-D Convolution	60(S) × 80(S) × 32(C) × 1(B)	Weights 3 × 3 × 16 × 32 Bias 1 × 1 × 32
batchnorm_3	Batch Normalization	60(S) × 80(S) × 32(C) × 1(B)	Offset 1 × 1 × 32 Scale 1 × 1 × 32
relu_3	ReLU	60(S) × 80(S) × 32(C) × 1(B)	-
fc_1	Fully Connected	500(C) × 1(B)	Weights 500 × 153600 Bias 500 × 1
relu_4	ReLU	500(C) × 1(B)	-
fc_2	Fully Connected	1000(C) × 1(B)	Weights 1000 × 500 Bias 1000 × 1
relu_5	ReLU	1000(C) × 1(B)	-
fc_3	Fully Connected	76800(C) × 1(B)	Weights 76800 × 1000 Bias 76800 × 1
relu_6	ReLU	76800(C) × 1(B)	-
layer	Function	240(S) × 320(S) × 1(B)	-

Sare neuronalak guztiz definitu ondoren, haren entrenamendua gauzatzen da. Horretarako, ezinbestekoa da galera edo *Loss* funtzioa definitzea. Bi “distantzia matrizeen” konparaketa egin behar denez, egokia izango da MSE (*Mean Squared Error*) funtzioa erabiltzea. Batez besteko errore koadratikoaren errendimendu normalizatuko funtzioak, errore kopurua neurtzen du ereduaren eta hurrengo ekuazioaren bidez definitzen da:

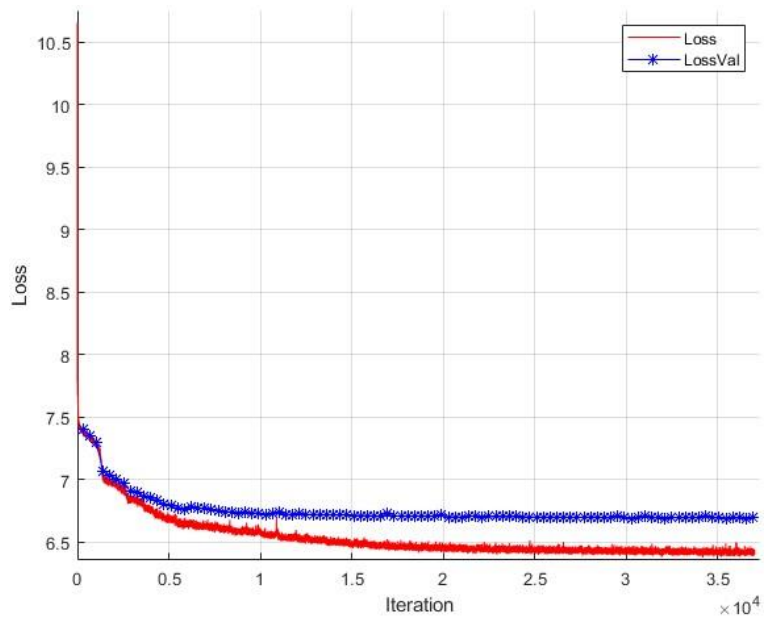
$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (59)$$

Ekuazioan erreparaturik, y_i aldagaiak LiDAR-ak lortutako distantzien matrizeari dagokio eta \hat{y}_i , sareak egindako distantzia iragarpen balioei egiten dio erreferentzia, LiDAR matrize horri parekatuta doan irudiarekiko. Sarearen gainontzeko entrenamendu balioak hurrengo **Taula 16**-an biltzen dira. Hauek garai kopuruari, irudien multzo zatiei eta ikasketa ratioari dagozkie.

Taula 16: Sare neuronalaren entrenamendu grafika

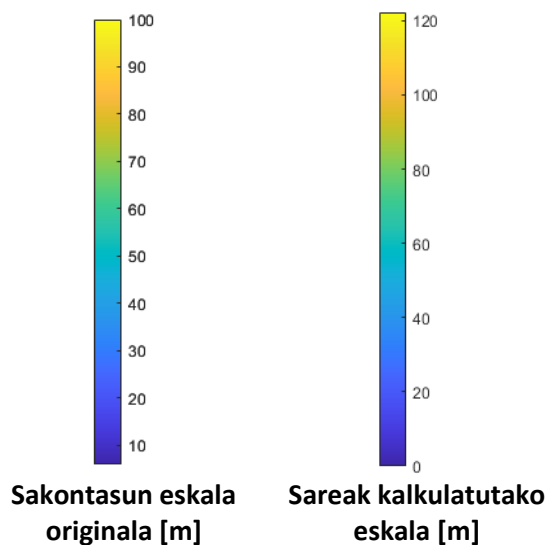
Parametroa	Balioa
Num Epochs	1000
Mini Batch Size	100
Learning Ratio	0,001

Sarea trebatu ostean ondoren erakusten den entrenamendu grafika lortzen da, zeinetan marra gorriak entrenamenduaren *Loss*-a errepresentatzen duen eta urdinak balidazioaren galera. Espezifikatzeko eta *MSE* funtzioagatik kalkulatuak izan direla gogoratu, entrenamenduaren galerak $Loss = 6,412$ [m]-ko balioarekin amaitu du eta $LossVal = 6,692$ [m]-koa izan da, 37000 iteraziotan.



Irudia 62: Sare neuronalaren entrenamendu grafika


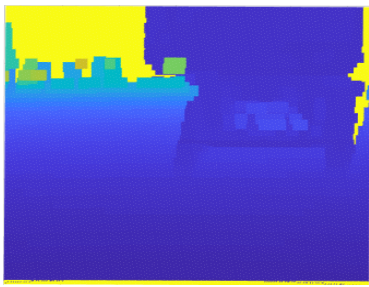
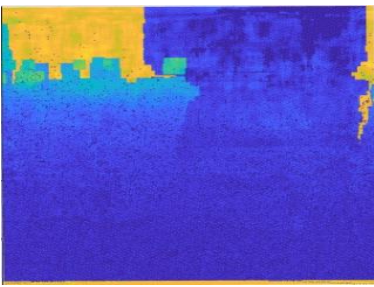

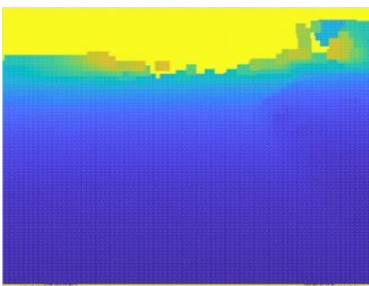
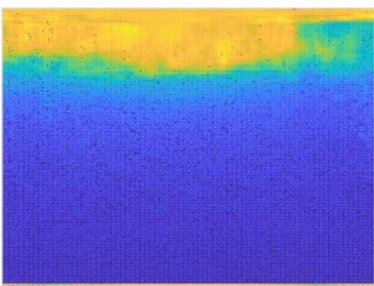

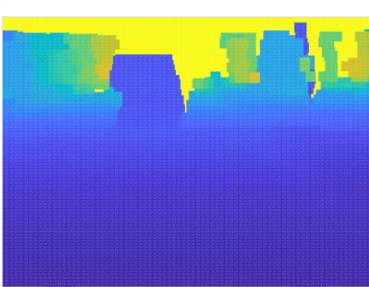
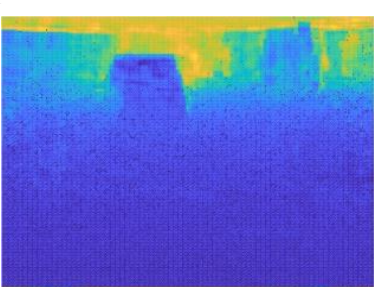

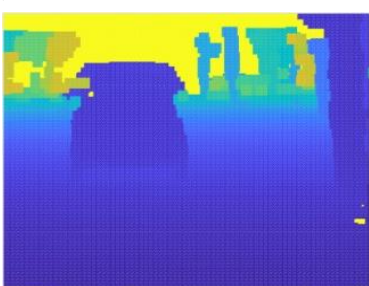
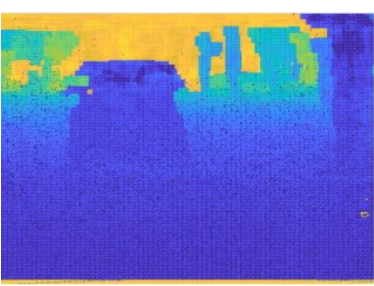

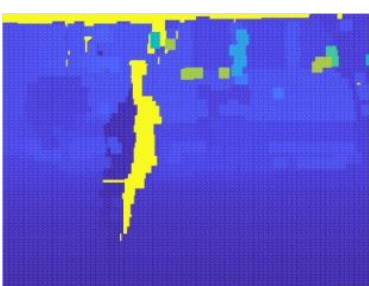
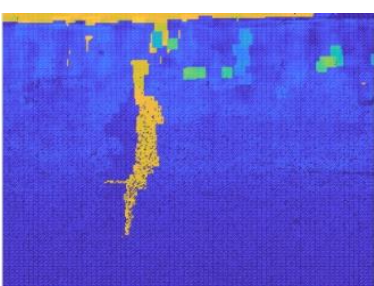
Grafikan lortutako balioak irudikatzeko eta bisualki esanguratsuagoak diren emaitzak aurkezteko, testatze irudietan frogatzen da sare neuronalaren irudi horiek ondo ulertzeko, lehendabizi, hauen eskalak agertzen dira hurrengo irudian.



Irudia 63: Sakontasunaren eta Sareak emandako eskalak [m]-tan

Hurrengo **Taula 17**-ak beraz, irudi originala, sakontasun originala eta sarearen bidez lortzen den distantzia emaitza erakusten du testatzeko erabiltzen diren bost irudi ezberdinetan. Hauetan antzeman daiteke sarearen emaitzak asko hurbiltzen direla errealitatera eta eskalan erreparaturaz, distantzien kalkulua ere egokia dela ikus daiteke. Azken irudian arazoak egon dira eta ez ditu emaitza aproposenak lortu baina sakonduz, LiDAR-ak jasotako datuen arazoa dela ohartu da. Gainera, sareak aurrezandako irudietan puntu urdinak antzeman daitezke, *overfitting*-a egon daitekeenaren seinale.

Taula 17: Distantzia sarearen entrenamenduaren emaitzak

Irudi originala	Sakonera originala	Sarearen emaitza
		
		
		
		
		

Sarearen diseinua egokia dela balioztatzeko eta konparaketa erara, bost aldaketa proposatzen dira topologian. Ondorengo taulak ondorioz, aldaketa horiek erakusten ditu, barneratu diren geruzekin. Galera balioak ere zehazten dira alderaketa numerikoa egin ahal izateko. Aipatu beharra dago aldaketa hauek 300 garaitan entrenatu direla eta aurreko sareak iterazio berdinetan, hau da, 10800 iteraziotan $Loss = 6,433$ [m] eta $LossVal = 6,704$ [m] balioak izan dituela.

Taula 18: Sarearen aldaketen datuak eta emaitzak

Topologiaren aldaketa	Gehitu edo kendutako geruzak	Aktibazioak	Loss [m]	LossVal [m]
-	2-D Transposed Convolution Batch Normalization ReLU <i>(relu_3 eta fc_1 artean)</i>	$121(S) \times 161(S) \times 32(C) \times 1(B)$ $121(S) \times 161(S) \times 32(C) \times 1(B)$ $121(S) \times 161(S) \times 32(C) \times 1(B)$	6,343	6,590
-	2-D Transposed Convolution Batch Normalization ReLU 2-D Transposed Convolution Batch Normalization ReLU <i>(relu_3 eta fc_1 artean)</i>	$121(S) \times 161(S) \times 32(C) \times 1(B)$ $121(S) \times 161(S) \times 32(C) \times 1(B)$ $121(S) \times 161(S) \times 32(C) \times 1(B)$ $243(S) \times 323(S) \times 16(C) \times 1(B)$ $243(S) \times 323(S) \times 16(C) \times 1(B)$ $243(S) \times 323(S) \times 16(C) \times 1(B)$	6,485	6,674
-	2-D Convolution Batch Normalization ReLU <i>(relu_3 eta fc_1 artean)</i>	$60(S) \times 80(S) \times 64(C) \times 1(B)$ $60(S) \times 80(S) \times 64(C) \times 1(B)$ $60(S) \times 80(S) \times 64(C) \times 1(B)$	6,360	6,585
Fully Connected ReLU Fully Connected ReLU <i>(fc_1, relu_4, fc_2 eta relu_5)</i>	-	$2000(C) \times 1(B)$ $2000(C) \times 1(B)$ $2000(C) \times 1(B)$ $2000(C) \times 1(B)$	6,451	6,654
-	Fully Connected ReLU <i>(relu_3 eta fc_1 artean)</i>	$100(C) \times 1(B)$ $100(C) \times 1(B)$	6,501	6,753
-	2-D Convolution Batch Normalization ReLU <i>(kendu topologiatik)</i>	$60(S) \times 80(S) \times 32(C) \times 1(B)$ $60(S) \times 80(S) \times 32(C) \times 1(B)$ $60(S) \times 80(S) \times 32(C) \times 1(B)$	6,624	6,776

Aurreko taularen lehenengo zutabeak sarearen geruzek jasan dituzten balio aldaketak jasotzen ditu. Bigarrenak aldiz, sarean sartutako edo saretik eliminatutako geruzak, hauen posizioekin, aktibazioak hirugarren zutabearen adieraziz. Azkenik, galera funtzioaren balioa eta balidazio galeraren datuak jasotzen dira. Emaitza guztiak alderatuz, proposatutako aldaketek ez dutela gehiegizko aldaketa suposatzen nabarmentzen da. Lortutako emaitzak ingurukoak dira, 0,1 [m]-ko mailan bariazioak dituzte soilik aipagarria den egitura bat lortu gabe. Prozesua “oinarri sarea”-rekin egiten jarraitzea erabaki da, entrenamendu luzeagoa jaso duelako eta emaitzak ez direlako oso ezberdinak.

4.2.1. OBJEKTUEN DETEKZIOA ETA DISTANTZIA

Segurtasun atala osatzeko, objektuen detekzioaren beharra dago. Garrantzia distantziaren kalkuluan dagoenez eta jada objektuak detektatzeko sare asko existitzen direnez MATLAB-ek eskaintzen duen “*Object Detection Using Faster R-CNN Deep Learning*” [420] kodigoan oinarritzea erabaki da.

Sare hau entrenatzeko aurreko sarean erabilitako datu basea berreskuratuko da baina moldaketak egin ostean, hau da, **Irudia 60**-an azaltzen diren irudiak dituen *dataset*-a. Hauetatik soilik RGB irudiak beharko dira bigarren sare honetan aplikatzeko. Irudiez gain, sareari oztopoak non dauden erakutsi behar zaio eta hori MATLAB-en *Image Labeler* [421] aplikazioaren bidez egin da. Enpresako ingurunera gehiago hurbiltzeko asmoz, bi etiketa sortu dira. Bata “*vehicle*” izenekoa, industria ingurunean egongo diren gainontzeko ibilgailu mugikorrei erreferentzia egiteko. Bestea aldiz, “*people*” izenekoa, langileen errepresentaziotzat sortu da. Aurrez aipatu den bezala, KITTI *Dataset*-ak kanpoaldeko ikuspegiak biltzen ditu beraz etiketatzen autoak eta jendea bildu dira, hurrengo irudian agertzen den erara.



Irudia 64: Irudietan sortutako label-ak

Eredu honek *Faster R-CNN* sare neuronal bat erabiltzen du helburua lortzeko, hau da eskualdeak erabiltzen dituen sare neuronal konboluzionala. Gogoratu, ohiko CNN sareak lehendabizi irudiaren eskualde bat aurkitzen du, objektua eduki ahal duena. Proposatutako eskualde horietatik ezaugarriak erazten ditu, azkenik ezaugarri horien arabera klasifikazio bat egiteko. R letrak, eskualdeak karratuak erabiliz aukeratzen direnaren kontzeptuari egiten dio erreferentzia eta *Faster* adjektiboak eskualde proposamenen sare bat gehitzen dela adierazten du, *Bounding Box*-en erabilerarekin detekzioa eginez [422]. Aipatu beharra dago ezaugarri eraztearen geruza *ResNet50* bat dela. Entrenamendua egiteko hurrengo konfigurazioa finkatzen da:

Taula 19: Oztopoen detekzioa egiten duen sare neuronalaren entrenatzeko balioak

Parametroa	Balioa
Max Epochs	10
Mini Batch Size	2
Initial Learning Ratio	0,001

Entrenamendua egin ostean ondorengo emaitzak lortzen dira:

Taula 20: Oztopoen detekzioa egiten duen sare neuronalaren emaitzak

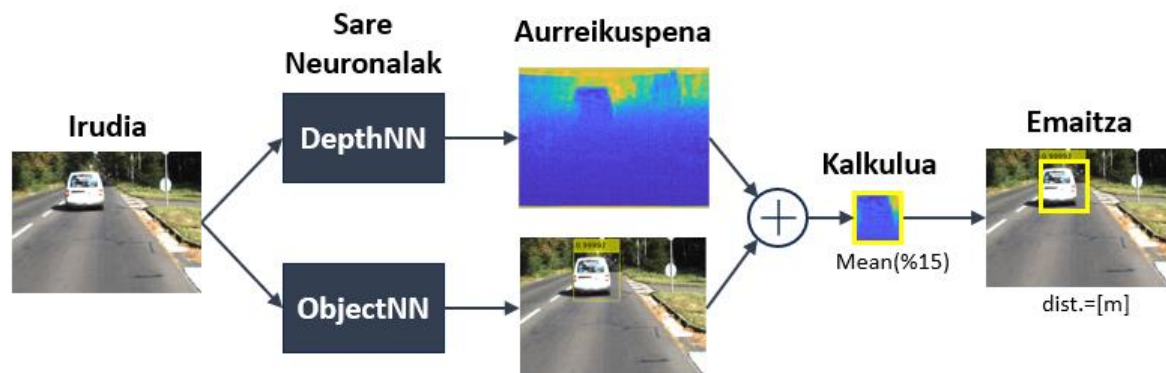
Epoch	Iteration	Time Elapsed	Mini-batch Loss	Mini-batch Accuracy	Mini-batch RMSE
10	15900	15:40:45	0.0607	99,79%	0,12
RPN Mini-batch Accuracy	RPN Mini-batch RMSE	Validation Loss	Validation Accuracy	Validation RMSE	Base Learning Rate
99,61	0,19	0,2061	99,51%	0,12	0,0010

Emaitza hauek bisualki interpretatzeko, testatzeko erabili den irudi multzoa aurkezten da ondoren. Honetan, sare neuronalak aurreikusten duen *Bounding Box*-ak agertzen dira irudi bakoitzeko, “vehicle” edo “people” izateko probabilitatearekin 0 eta 1 artean.



Irudia 65: Oztopoen sareak detektatutako objektuak

Hauteman daitekeen bezala, gertuago dauden oztopoen aurreikuspena zuzenagoa da ia %100-eko igartze probabilitatea emanez. Geroz eta gehiago aldenuz, asmatze probabilitatea jaisten da baina detekzioa egiten jarraitzen du modu egokian. Orain, behin distantziak neurtzen dituen sare bat eta objektuak detektatzen dituen beste sare bat izanda, bien irteerak elkartuko dira oztopoen distantzia estimatzeko. Hurrengo irudiak sistemaren planteamendua erakusten du.



Irudia 66: Oztopoaren distantzia kalkulua

Hasteko, irudia hartzen da eta diseinatu diren bi sareetatik pasatzen da hauek aurreikuspenak egin dezaten. Ondoren, *Bounding Box*-aren tamaina eta posizioa kontuan izanda, sakontasunaren emaitzaren pixel berdina aukeratu dira. Sakontasunaren aurreikuspenean zenbait datu 0 balioa har dezaketenez, balio horiek alde batera uzten dira eta, *Bounding Box*-a osatzen duten pixelen %15-a kalkulatu da. Sakontasun datu kopuru hori hartzen da, beti ere minimoak aukeratu. %15-a datua errealagoa izan dadin finkatu da, datu bakarra hartu ezkerok neurri okerra kalkulatzeko aukera dagoelako. %15 minimoaren batez bestekoa egiten da eta hori izango da oztopoaren distantzia ibilgailu autonomoarekiko.



Irudia 67: Oztopoen distantzien emaitzak

Aipatzekoa da, irudi bat ateratzen denetik distantzia datua lortzen den arte $t = 0,6657$ [s] behar izaten dituela batez beste. Sakontasuna kalkulatu duen sareak $t = 0,3025$ [s] behar ditu eta oztopoen detekzioarenak $t = 0,59425$ [s], batez beste. Denbora datu hauek Intel® Core™ i9-9800H CPU 2,30Ghz 3,30Ghz prozesadore bat erabili lortu dira. KITTI datu base originala hartzen da, leundu eta tamaina ajustatu ostean, prozesu berdinetik pasatzeko. Horrela, emaitzen doitasuna konparatzeko neurri bat lortzen da, hau da, neurri errealak.

Taula 21: Test-irudien objektuen neurri errealak

Test Irudia 1	Test Irudia 2	Test Irudia 3	Test Irudia 4	Test Irudia 5
9,937 [m]	100 [m]	18,8108 [m]	11,7889 [m]	6,7098 [m]

Kontuan izatekoa da sakontasun sareak ematen duen errorea emaitza hauen konparaketan. Aurreikusitako datuak datu errealen ingurukoan dira, errore maximoa hirugarren irudian egonik, 5,1448 [m]-takoa izanik eta minimoa azken test irudian ikus daiteke, 0,1158 [m]-ko balioaz.

Irudi batean dauden oztopoak detektatzeko eta hauek ibilgailu autonomoarekiko duten distantzia kalkulatzeko prozesua hurrengo taulan zehazten da.

Taula 22: Oztopoen distantzia kalkulatzeko algoritmoaren fluxua

Oztopoen detekzioa eta distantzia kalkulua	
1:	Sakontasun sarea kargatu
2:	Objektuek detekzioaren sarea kargatu
3:	while ibilgailua martxan
4:	Irudia atera
5:	Irudia sakontasun saretik pasatu
6:	Irudia objektu detekzio saretik pasatu
7:	for i=oztopo kopurua
8:	0 diren pixelak baztertu
9:	<i>Bounding Box</i> -eko %15 pixel kopurua kalkulatu
10:	<i>Bounding Box</i> tamainaren sakontasuna lortu
11:	%15 kopuruarekin bat datozen balio minimoak gorde
12:	Balio horien batez bestekoa kalkulatu
13:	end
14:	end

Modu honetan, AV-ak nabigatzen hari den bitartean irudiak jaso ditzake, eta oztupoaren distantziaren arabera eta abiadura propioaren arabera, erabakiak hartu segurtasuna bermatzeko. Kontuan izan beharko da gainera ibilgailu autonomoak daramatzen abiadura. Distantzia bat kalkulatzeko denbora 0,6 [s] ingurukoa bada, abiaduraren arabera, distantzia hori kontuan izan beharko da. Adibidez, $V = 1$ [m/s]-koa bada, distantzia balioa kalkulatzeko duenerako 0,6 [m] aurrerago egongo da eta balio hori oztupoaren distantziari kendu beharko zaio neurketa aproposa izateko.

ATALBURUA 5

AUTOREAREN EKARPENAK

*Ez izan beldurrik zailtasunei,
onena horietatik etortzen da.*

RITA LEVI-MONTALCINI

1. Kontribuzioa

Doktorego-tesi hau burutu den heinean, argitalpen ezberdinak egiten joan dira. Hauek, ibilgailu autonomoen tekniken garapenean oinarrituta daude batez ere. Unibertsitatean aurki daitekeen AGV plataforma mugikorrean aplikatzeko daude pentsatuta, beraz honen ezaugarri dinamikoak integratzen dira lan guztietan. Beste alde batetik, zenbait lanetan kolaboratu da ere eta hauek sare neuronalen ikaskuntza prozesuarekin erlazionatuta daude bereziki. Ondorioz, argitalpen hauek bi multzotan sailka daitezke non, lehengo multzoan lehenengo autore bezala garatutako lanak bilduko diren eta bigarrenik, beste autoreekin kolaboratutako lanak aipatuko diren. Gainera, kongresutan ere parte hartu da, idatzizko lanak aurkeztuz. Hurrengo **Taula 23**-an, guztira argitaratu diren lanak biltzen dira.

Taula 23: Autoreak egindako ekarpenak

Izenburua	Aldizkaria/Kongresua	Erreferentzia
Dynamical Analysis of a Navigation Algorithm	MDPI Mathematics (Q1)	[423]
Stability Analysis for Autonomous Vehicle Navigation Trained over Deep Deterministic Policy Gradient	MDPI Mathematics (Q1)	[424]
A Study of Learning Issues in Feedforward Neural Networks	MDPI Mathematics (Q1)	[425]
Modification of Learning Ratio and Drop-Out for Stochastic Gradient Descendant Algorithm	MDPI Mathematics (Q1)	[426]
Dynamic Window Approach for Autonomous Vehicles with Mecanum Wheels	Basque Conference on Cyber Physical Systems and Artificial Intelligence	[427]

Hurrengo ataletan, lan hauen deskribapen labur bat emango da, bakoitzean garatutako ideia orokorrak bilduz.

1.1. Lehen autore gisa egindako lanak

Lehenengo egile bezala, tesi honen autoreak bi lan argitaratu ditu. Hauetako lehena, **“Dynamical Analysis of a Navigation Algorithm”** [423] izenarekin aurkitu daiteke MDPI-ko *Mathematics* aldizkarian. 2021-eko abenduaren 21-ean argitaratutako lan hau ibilgailu autonomoen nabigazioekin erlazio zuzena dauka, horretarako algoritmo bat aurkezten bait da. Gainera, ordenagailu bidezko ikusmena eta sare neuronalak ere barneratuta daude. Stanley algoritmoaren egokitzapen bat egiten da unibertsitatean dagoen AGV plataformarentzako, bertan egiten bait dira frogak jatorrizko algoritmoarekin konparatzeko. Egokitzapen hau, sendotasuna handiagotzeko diseinatu zen eta hori bermatzeko *Lyapunov* teknikaren bitartez egonkortasun analisi bat egiten da, algoritmoari gehitzen zaizkion bi konstante balioztatuz. Honetaz gain, AGV-ak lortutako irudien segmentazio semantiko bat egiten da sare neuronal bat erabiliz, jarraitu behar den ibilbidea kalkulatzuz eta robotaren lokalizazioa bidearen menpe egonez.

Bigarren lana 2022-ko abenduaren 27-an kaleratu zen MDPI-ko *Mathematics* aldizkarian ere. Kasu honetan izena **“Stability Analysis for Autonomous Vehicle Navigation Trained over Deep Deterministic Policy Gradient”** [424] da. Honetan, errefortzuzko ikaskuntzarekin lan egiten da, honen funtzionamendua barnerratzeko. Aplikazio simple bat aurkezten da zeinetan gelaxka baten barruan dagoen ibilgailu bat bertatik irten behar den oztopoekin talka egin gabe. Sare neuronalak sortu eta entrenatzen da eta antzeman daiteke entrenamendu mota honek akats ezberdinak izaten dituela, espero den emaitza lortzeko ezintasunaz. Artikuluan, Lyapunov ekuazioa metrika gisa erabil daitekeela frogatzen da DDPG-tik lortutako agente baten egokitasuna ebaluatzeko. Guztia, esan bezala, aplikazio jakin baterako da, kasu honetan AGV-en nabigaziorako eta egonkortasun analisi bat egiten da ibilgailua entrenatzeko diseinatzen den sari-ekuazioan oinarrituta. Simulazioan geratzen da lana.

1.2. Kolaborazioak

Beste autore batzuekin egindako kolaborazioei dagokienez, bi lan aurkitu daitezke. Alde batetik, **“A Study of Learning Issues in Feedforward Neural Networks”** [425] izena duen lana, MDPI-ko *Mathematics* aldizkarian argitaratua 2022-ko irailaren 5-ean. Hau, sare neuronalen arkitekturekin du erlazioa eta hauen ikasketan sortzen diren arazoekin. Bertan, *Feedforward* sare neuronal bat entrenatzeko erabiltzen diren zenbait patroik behar bezala ikasteko gai ez izateko bi arazo posible azaltzen dira. Aztertutako lehenengo arrazoa, jakobiarrak prozesu geruzetan desagertzearen zergatia da. Bigarrena aldiz, guztiz konektatutako geruzen jakobiar horiek hartzen duten norabidea da.

Beste aldetik, MDPI-ko *Mathematics* aldizkaria aukeratu zen ere **“Modification of Learning Ratio and Drop-Out for Stochastic Gradient Descendant Algorithm”** [426] izena duen lana argitaratzeko. 2023-ko otsailaren 28-an kaleratu zen lan honetan, *Stochastic Gradient Descendant* algoritmoan zenbait doikuntza egiten dira. Hau, galera-funtzioa murrizten saiatzeko helburuarekin egin zen, erabiltzen den sare neuronalaren konplexutasuna nahitaez igo beharra izan gabe. Gainera, sare neuronal honen diseinua aldakuntza horiekin entrenamendu denbora igo ez dezan burutu zen. Horretarako, froga ezberdinak planteatu ziren adibidez, *Loss* funtzioa geruzaz geruza transmititzen den proba edota *Dropout*-a sarearen geruza bakoitzean exekutatzen den proba.

1.3. Kongresuak

Kongresuetan ere parte hartu da, hain zuzen ere **“Basque Conference on Cyber Physical Systems and Artificial Intelligence”** [427] kongresuan, 2022-ko maiatzaren 18-a eta 19-a bitartean ospatua. Bertako *proceedings paper*-etara **“Dynamic Window Approach for Autonomous Vehicles with Mecanum Wheels”** ikerketa bidali zen. Aurkeztutako lana, DWA algoritmoaren aplikazio bat da *Mecanum* gurpilak dituen ibilgailu autonomo batentzako. Kontuan izan behar, da gurpil hauek direkzio-anitzak hartzearen gaitasuna ematen diotela robotari. Bertan, gurpil mota horiek erabiltzen direnean jarraitu beharreko estrategia berri bat zehazten da.

ATALBURUA 6

ONDORIOAK

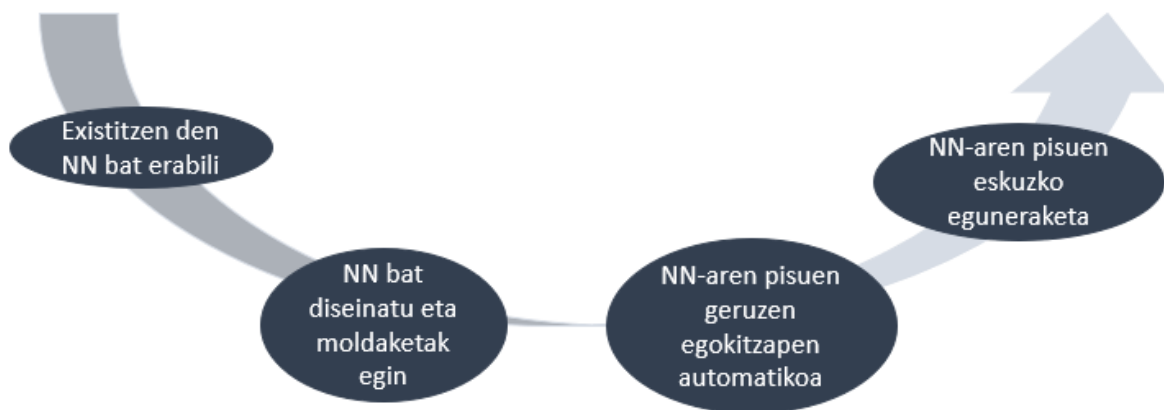
*Bide zuzenean zaudela baldin badakizu, sentitzen baduzu, orduan...
ez du inporta zer dioten.*

BARBARA McCLINTOCK

1. Deliberamendua

Lan honek barneko ibilgailuentzako nabigazio sistema bat aurkezten du. Ideia horretatik abiatuz emaitza egoki ezberdinak aurkitzen jardun da. Bibliografiaren analisiarekin, teknologia honek oraindik lan asko duela aurretik ikusi ahal izan da, zehaztasuna eta fidagarritasuna guztiz lortzeko. Antzeman denez, aurkeztutako nabigazio sistema atal ezberdinez osatuta dago. Alde batetik bidea lortzeko errefortzuzko ikaskuntzaren planteamendua. Ondoren, ingurunea eta lokalizazioa aurrea eramateko lana eta azkenik, oztopoen distantzia detektatzen duen prozedura, segurtasuna bermatzeko.

Tesi proiektu lan honetan sare neuronalek duten etapa guztiak aztertu nahi izan dira metodologian zehar. Hasteko, NN-batekin egin daitekeen zeregin sinpleena sare bat hartu eta norberaren aplikazio baterako erabiltzea da, hau da, *Transfer Learning*-a egitea. Kasu honetan, datu base propioa aplikatu zaio objektuak detektatzen zituen sare bati, detektatu nahi ziren klaseak zehaztuz. Bigarren aurrera-pausu batean, sare neuronal baten diseinua egin da, honen geruza bat ere formulatuz. Hau sakontasuna neurtzen duen sare neuronal izan da eta topologiaren moldaketak egin dira emaitza hobetuz. Topologia aldaketa hauek eskuz egin beharrean era automatikoan egin daitezke eta tesi honetan, planteamendu ebolutiboa erabili da helburu honetarako. Modu horretan, sare neuronalaren aurreikuspenaren arabera topologia egokitu egiten da. Azkenik entrenamendu prozesua pausuz pausuz planteatu da, pisuen eguneraketa eginez eta garai bakoitzean lortutako emaitzak aztertuz. Hurrengo irudiak etapa hauek aurkezten ditu.



Irudia 68: Sare neuronalaren etapa proiektuan

Metodologian aurkitzen den lehengo garapena errefortzuzko ikasketaren optimizazio analitiko bat da. Azaldu den erara, agentea lortzeko era sendo bat aurkitu dela esan daiteke. Entrenamendu ugari egin dira eta beti agente berbera eskuratzen dela konprobatu da, arbitrariotasuna galduz. Gainera, ibilgailuaren ibilbide galeren aurrean, agentea, hau da, sarea, berriz exekutatu daiteke bide berri bat lortuz. Honetaz gain, entrenatzeko modu eraginkor bat aurkezten da pausuz-pausuzko prozesuarekin.

Halabaina, agente egoki bat lortzeko entrenamendua luzea da eta ohiko *Reinforcement Learning* teknikek metodo bizkorragoak aurkezten dute, adibidez, MATLAB-eko *Reinforcement Learning Episode Manager*-arekin konparatuz. Hau konbergentzia handia behar duelako da. Nahiz eta prozesu hau motela izan, *offline* exekutatzen dela gogoratu behar da, *online* exekuzioa agenteari solik aplikatzen bait zaio eta hau azkarra da, sarearen sinpletasunagatik.

Reinforcement Learning Episode Manager—aren bidez entrenatua izan da eboluzioaren bidezko lortutako agentea, alegia, lan honetako bigarren garapena. Honetan baldintzak azkarren betetzen dituen agentea lortzea bermatzen da. Mota honetako entrenamenduaren izaera dela eta, ezin da ziurtatu agenterik onena lortu denik eta hori emaitzetan islatzen da. Beraz, sistema honen planteamendua egokia dela uste da baina lana du oraindik egiteko, diseinatutako sistema arrunta bait da agenterik egokiena lortzen dela bermatzeko.

Sarea optimizatu nahi izan da, hasierako populazioan kromosoma bakar bat egongo balitz bezala. Bideak kromosomatzat hartuta eta hauek egiten duten tartea aztertuz emaitza egokiak lortzen dituzte Bakdi et al. [242]-ek. Inguruneko biderik egokiena lortzen dute, baina *fuzzy* logikarekin lan egiten dute. Lan honen arazo berdinerako diseinatzuten Santiago R et al. [245]-ek ere sistema bat, algoritmo genetikoez baliatuz. Mapa kargatzen dute eta baldintza batzuk betetzen dituzten bideak erabiltzen dituzte kromosoma bezala berriz ere. Ez dute biderik optimoena lortzen baina bai talka gabeko nabigazioa egitea bermatzen duena. Du et al. [256]-en kasuan neurona sareetan jartzen dute arreta, honen topologian hain zuzen ere. Sarearen nodoen loturak hartzen dituzte kromosomatzat eta biderik egokiena lortzeko optimizatzen dituzte. Ondorioz, planteamendua beste ikuspuntu batetik hartu behar dela nabaria da.

Proposatutako alternatiba ebolutiboak, hau da, sare neuronalak erabili ezean K parametro baten doikuntza egiten duen sistemak, nabigazio gogobeteko bat lortzen du. Optimizazio hau zuzenean akzioan du eragina, sareek hau kalkulatu beharrean era analitikoan eginik. Horrela, sistema sendo eta eraginkor bat lortu da, aurreko aukerak lortu ez duena. Arazorik gabe heltzen da hasierako puntu batetik helmugara, ibilgailu autonomoen nabigazioaren helburua lortuz. Gehitzeko, aurreko optimizazioek baino azkarrago lortzen du nahi den emaitza.

SLAM algoritmoari dagokionez, era eraginkor bat aurkezten da ikaskuntza sakoneko teknirik erabili gabe. Ikusmen artifiziala eta etiketen erabilerarekin aurrera eramaten da. Argi dago ez dela AprilTag-ak erabiltzen dituen lan bakarra eta bakoitzak bere berezitasunak ditu. Bertoni et al. [376]-ek ere aldibereko mapatze eta lokalizazio sistema bat aurkezten dute baina kasu honetan etiketak lurrean daude kokatuta, hormetan egon ordez. Errorea 0,11 [m]-tatik behera mantentzen dutela diote beraz lan honetan diseinatutako sistematik ez da asko aldentzen, lortutako errorerik handiena 0,13 [m]-koa bait da. Honek fidagarritasunarekin lehian dagoen sistema diseinatu dela ziurtatzen du.

Muñoz-Salinas et al.([377], [378])-ek hormetan sakabanatzen dituzte etiketak baina puntu-hodei bat sortzen dute etiketak multzokatuz, prozesua konplexuagoa bilakatzen duena. Gainera, lan honetan bezala errore metatu bat sortzen dute. Beste alde batetik, ezin izango zuten erraztasunez mapa handitu beraz ez du malgua bilakatzen.

Beste lanekin ezberdintasun handiena markatzen duena sistema honek kamera bakar baten erabilera egiten duela da. Honek prozesuaren sinpletasuna adierazten du eraginkortasuna bermatuz. Merkea dela ere argi dago, eta exekutatzeke erraza da. Antzeman dena baina, beste ikerketek aipatu duten bezala, argiztapenak arazoak sortzen dituela da eta erabilitako kamerak automatikoki fokuratzen duela. Bi alderdi hauek, askotan, AprilTag-en galera edota irakurketa okerretara eramaten du. Gainera, zehaztasunaren galera metatu egiten da detekzioekin aurrera egin ahala.

Hala ere, nahiz eta ingurune handietarako aplikazio zaila duela iruditu, hormen detekzioa era berean egingo zen, edozein inguruneetan aplikagarria izanez. Honek beraz, sendotasuna adierazten du. AprilTag berriak ere sartu ahalko ziren maparen tamaina eta egitura aldatuz, malgua izanez. Bestalde, AV-ko beste sistemekin konbinatzeko ez zen arazorik suertatuko.

Oztopoen distantzien kalkuluaren inguruan, sare neuronal konboluzional batekin batera, ataza betetzea lortzen da, bi sareek entrenatuta izan diren helbururako egokiak direla demostratuz. Hemen garrantzi handiena duena datu multzo egoki baten erabilera da, arazoa era ezin hobean definituz eta noski, ahalik eta azkarren emaitza lortzea. Kameraren erabilera hutsarekin egiten da berriz ere beraz, sentsore aldetik merkea da.

Sare neuronalei buruz, aipagarria da, sakontasunaren sarearen topologia aldaketek ez dutela hobekuntza nabarmenik adierazten eta hori ikerketaren beharra adierazten du. Sakontasun sare neuronalarekin jarraituz, sareak aurreikusten dituen emaitzetan arreta jarriz (ikusi **Taula 17**) puntu urdinak agertzen direla antzeman daiteke. Puntu hauek emaitza-irudian sakabanatuta agertzen dira eta *overfitting*-a egon dela adierazi dezakete. Gehiegizko doitze honek datuen nahasmena ekartzen du eta ondorioz, zehaztasunaren galera.

Objektuen detekzio sare neuronaleko emaitzetan ere zenbait adierazle nabaritu dira. Zenbat eta urrunago egon oztopoa, zenbat eta txikiagoa da klase probabilitatea eta honek fidagarritasun galera suposatzen du. Gainera, urrun dauden objektuak ere galtzera heltzen da, hau da, ez detektatzera. Oso gertu dauden oztopoen kasuan, nahiz eta leiho bateko erreflexuan egon detektatu egiten du eta horrek segurtasun erroreak suposatu dezake, existitzen ez den egoera bat aurreikusi duelako.

Bestalde, segurtasun sistema bat izateko motela da. Denbora asko pasatzen du irudiak prozesatzen eta hori ez da onargarria. Chang et al. [168]-ek CNN-aren erabilerarekin denbora arazo bat bihurtzen dela ere eztabaidatu zuten, lan honen denbora galtzea ohikoa dela ondorioztatuz. Mapetan sakontasun estimazioa egiteko, Wang et al. [176]-ek RNN-a erabiltzen dute eta ez dute ere denbora aldetik hobekuntza nabarmenagorik lortu denborarekiko. Gainera erabiltzen duten datuak konplexutasun handiagoa dute.

Orokorrean lan honen helburua lortu dela ondorioztatzen da, ibilgailu autonomoen nabigazio sistema batek duen atalen garapenarekin. Atal hauek era egokian burutzen dute egitekoa eta aproposak direla baloratzen da AV-ko sistema batean txertatzeko.

ATALBURUA 7

ETORKIZUNEKO IKERKETA ILDOAK

Nire ulermena

ulertu nahi dudan guztiaren frakzio infinitesimal bat baino ezin da izan.

ADA LOVELANCE

1. Lanaren Jarraipena

Hirugarren atalburuan aurkeztu diren beharren aurrean, zenbait eremu garrantzitsu daude ibilgailu autonomoen ikerketa ildo jarraitzeko. Hauek, bai industriaren barne, bai eremu akademikoan garatu daitezke eta lerro hauen hobekuntzak AV sistemen garrantzia handitu dezakete, segurtasuna eta fidagarritasuna lortzen lagunduz.

Errefortzuzko ikasketaren bidetik, ikerketa jarraitzen hari da gidaketa autonomoaren zenbait ataletatik, adibidez, kontrola eta bide plangintzan. Teknika hauek errendimendu ona lortu dute, AV sistemak ahalbidetzen dituzten politika ia optimoak ikasteko duten ahalmena adieraziz. Lan honetan bezala, bibliografian ikus daiteke metodo hauek simulazioan soilik frogatu direla edo gehienez, ingurune oso mugatuetan. Ondorioz RL sistemak ez dira oraindik guztiz fidagarriak ingurune errealetan nabigazioa gauzatzeko, segurtasuna dudan jarriz. Nahiz eta simuladore baten ingurune oso errealek sortu daitezken, ezin da espero baldintza errealetan portaera berdina izatea. *Offline* entrenatutako sistema hauekin beraz ikertzen jarraitu behar da, hauen ezarpenak ibilgailu autonomoen gainean fidagarriak izan daitezen.

Ikerketa lerroetako beste bat sare neuronal konboluzionalen bidetik doa. Sare hauek oso emaitza onak erakutsi dituzte ikusmen artifizialeko atazetan, esaterako, objektuen jarraipen eta saihestean edota abiadura kontrolean. Alabaina memoria askoren beharra izan ohi dute eta kostu konputazionalarentzako zama bat suposatzen dute, hauen erabilgarritasuna murriztuz. Lan honetan ikusi den bezala oso motela da objektuak detektatzeko orduan. Beraz, CNN eredu berrien diseinuaren beharra dago, ibilgailu autonomoen sistemak arintzeko.

Mugimendu plangintzaren eta kontrolaren artean datu fluxu zirkular bat izateak ere ikerketatzat hartu daiteke. Hauek biak ibilgailu autonomoen atal erabakigarriak dira, sistema osoaren funtzionamendu denboran oso garrantzitsuak bait dira. Orokorrean datuak planifikatzailetik kontrolera pasatzen dira baina interesgarria izango litzateke alderantzizko fluxua ere izatea, karga konputazionala orekatzeko eta abiadurarekin segurtasuna mantentzeko.

Ibilgailu autonomoen sendotasuna eta segurtasuna bermatzeko, orokorrean, sentsore askoz ekipatuta egoten dira. Jasotako datu hauek ikaskuntza sakoneko ereduak prozesatzen dituzte eta ildoetako bat sentsore guztien arteko balantza bat bilatzea izango litzateke. Entrenamendu prozesuak konputazio koste altua suposatzen dute datuen kopuruaren ondorioz eta online funtzionamenduan datu hauen berrizte prozesuak zama handia suposatzen du. Gainera behar bat ere sortzen da eredu hauei ahalik eta egoera kritiko gehien erakusteko.

Oztopo dinamikoen ildotik, ezin daiteke hauen portaera irudi bakar batetik lortu, eta irudi segida baten miaketa egin behar dela argi dago. Portaera bat aurreraketa konplexutasun maila igotzen du eta teknika horien beharra dago. Ondorioz, portaera hauen informazioa biltzen duten datu baseen behar bat existitzen da, segidak ikasten dituzten ereduak diseinatzeko. Datuekin jarraituz, erreferentziatzeko datu-base baten sortzea ezinbestekoa da. Nahiz eta *dataset* anitz dauden, orokor bat ez izateak, diseinatzen diren sistema guztien konparaketa egitea zailtzen du.

Azkenik, nahiz eta asko ikertzen ari den AV-en sistemen alderdi guztiak hobetzeko, industrializazio falta dago. Sistema hauek, eskala handian, fidagarriak izan beharko liriateke edozein industria ingurunetan eta moldagarriak. Gainera, beste funtzio askoren txertaketa, miaketa operazio modua esterako, interesgarria izango zen enpresei berain lehentasunei ajustatzeko aukera emanaz.

Fabrika bakoitza mundu bat da eta era unibertsal batean segurtasun terminoak ajustatzea, abiadura aldatzea edota funtzioak sartzekoa aukera ematea langileei AV-ak erakargarriagoak egingo ditu.

1.1. Tesi honen jarraipena

Proiektu honi dagokionez ere zenbait atzarekin jarraitu daiteke lanean. Orokorrena hiru sistemak barneratzen dituen kodigo baten garapena da, nabigazio osoa lortuz. Horrela, Vitoria-Gasteizko ingeniari-tza eskolan dagoen eredu inplementatzeko aukera egongo da eta bertan eremu-probak egin. Funtzionamendua ziurtatzeaz gain, denbora neurketak, kostu konputazionala eta bestelako balioak neurtzeko baliabide bezala erabili daiteke, existitzen diren beste ereduekin konparatzeko. Alabaina, algoritmo nagusira heldu baino lehenago zenbait kontzeptu daude oraindik garatzeko.

Errefortzuzko ikasketari dagokionez, entrenamendu luze bat burutzea izango zen lehengo lana. Hau hilabeteetako iraupena beharko duela aurreikusten da beraz ez da epe laburreko lan bat izango. Beste planteamendu bat ere eraman daiteke aurrera, DWA algoritmoaren funtzionamenduan oinarrituz. Honek helmugarako distantzia eta oztopoetarainokoa kontsideratu ordez, helmugarainoko desbideratze angeluaren errorea eta oztopoen distantzia hartzen ditu. Planteamendu hori ere interesgarria da baina formulen konplexutasuna igotzen du.

Eboluzionatu egiten duen sarearekin jarraitu nahi baldin bada, ikusi da beste lanek bideak erabiltzen dituztela kromosoma gisa. Beraz, agente batek lortzen dituen bide asko erabili daitezke eta adibidez, bide optimoa azkarren lortzen duen agentearekin geratu. Honek bidearen optimazioa ziurtatuko zuen eta agentea sare neuronalaren baitakoa denez, antzezlearen sarea ere. Algoritmo genetikoaren bidezko doikuntzan aldiz, hurrengo urratsa inguruan oztopoak izanda nola erantzungo duen konprobatzea da, ibilgailuan programazioa irauli aurretik.

AprilTag -etan egin diren sinplifikazioak kentzea izan daiteke SLAM-eko jarraipenetako bat, hormak 90º-tan egongo ez diren inguruneetan atentzioa jarritz. Gainera, irakurketen errorea leuntzen ere saiatzeko aukera dago, adibidez, argiztapen egokia duen ingurune batean frogatuz.

Objektuen detekzioari dagokionez, lehenengo ataza datu base propio baten sorrera izan beharko litzateke. Berezko irudi eta neurriak jasotzerakoan, algoritmoa ibilgailuaren funtzionamendu eremuarentzako aproposa bihurtuko du eta era eraginkorragoan lan egingo du. Beste alde batetik eta datu jasotzearekin erlazionatuz, LiDAR sentsoreak jasotzen dituen datuen analisi bat egin beharra dago. Galtzen duen informazioaren arrazoia bilatu eta hau berreskuratzeko metodo baten diseinuak, leuntze teknika alde batera utziko du, prezisio gehiagoz sarea entrenatzeko aukera emanez.

Sare neuronalaren ildotik, beste topologia baten bilaketa planteatu daiteke sakontasuna aurreikusteko. Sare honek ematen duen errorea altua da eta hori distantzia estimazioan erreflexatzen da. Bistakoa da errorea murriztu emaitza zehatzagoak lortuko direla. Sakontasun sare neuronalaren diseinua sinplea dela aprobeztatuz eta bi sare neuronal izan beharrean, prozesu honetarako erabiltzen diren bi sareak batzen dituen topologia baten diseinua planteatu daiteke. Sarrera berdina daukate eta irteera bien batuketa bat da azken finean beraz konbinazio bat proposatzeko aukera ikertu daiteke.

Azkenik, konprobatu da irudia bi sareetatik pasatzeko prozesua motela dela beraz topologiaren moldaketa beharrezkoa da konputazio denbora hau ahalik eta gehien murrizteko.

AMAIERA

Mila esker lan honen irakurketan murgildu diren pertsona guztiei

BIBLIOGRAFIA

ERREFERENTZIAK

- [1] 'Automated Guided Vehicles (AGV) Meaning & Types | 6 River Systems'. Accessed: Nov. 08, 2023. [Online]. Available: <https://6river.com/what-are-automated-guided-vehicles/>
- [2] 'Vehículo de guiado automático - Wikipedia, la enciclopedia libre'. Accessed: Nov. 09, 2023. [Online]. Available: https://es.wikipedia.org/wiki/Veh%C3%ADculo_de_guiado_autom%C3%A1tico#cite_note-:0-3
- [3] *ESTADO DEL ARTE- Historia de los AGV*. Accessed: Nov. 09, 2023. [Online]. Available: https://repository.eafit.edu.co/bitstream/handle/10784/5463/06_EstebanBetancur_JulianBetancur_JaimeBolívar_marcoTeorico_2011.pdf?sequence=16
- [4] R. Sell, A. Rassölkin, R. Wang, and T. Otto, 'Integration of autonomous vehicles and Industry 4.0', *Proceedings of the Estonian Academy of Sciences*, vol. 68, pp. 389–394, 2019, doi: 10.3176/proc.2019.4.07.
- [5] J. Bruna, P. #1, F. Gressler, and R. Seleme, 'Supply Chain 4.0: Autonomous Vehicles and Equipment to Meet Demand', *Int. J Sup. Chain. Mgt*, vol. 8, no. 4, p. 33, 2019, Accessed: Apr. 03, 2023. [Online]. Available: <http://excelingtech.co.uk/>
- [6] G. Sankeerthana and B. Raghuram Kadali, 'A strategic review approach on adoption of autonomous vehicles and its risk perception by road users', *Innovative Infrastructure Solutions*, vol. 7, no. 6, Dec. 2022, doi: 10.1007/S41062-022-00951-4.
- [7] M. Alawadhi, J. Almazrouie, M. Kamil, and K. A. Khalil, 'A systematic literature review of the factors influencing the adoption of autonomous driving', *International Journal of System Assurance Engineering and Management*, vol. 11, no. 6, pp. 1065–1082, Dec. 2020, doi: 10.1007/S13198-020-00961-4/FIGURES/3.
- [8] A. Taeihagh and H. S. M. Lim, 'Governing autonomous vehicles: emerging responses for safety, liability, privacy, cybersecurity, and industry risks', *Transp Rev*, vol. 39, no. 1, pp. 103–128, Jan. 2019, doi: 10.1080/01441647.2018.1494640.
- [9] N. E. Bezai, B. Medjdoub, A. Al-Habaibeh, M. L. Chalal, and F. Fadli, 'Future cities and autonomous vehicles: analysis of the barriers to full adoption', *Energy and Built Environment*, vol. 2, no. 1, pp. 65–81, Jan. 2021, doi: 10.1016/J.ENBENV.2020.05.002.
- [10] K. Othman, 'Public acceptance and perception of autonomous vehicles: a comprehensive review', *AI and Ethics*, vol. 1, no. 3, pp. 355–387, Aug. 2021, doi: 10.1007/S43681-021-00041-8.
- [11] F. Duarte and C. Ratti, 'The Impact of Autonomous Vehicles on Cities: A Review', *Journal of Urban Technology*, vol. 25, no. 4, pp. 3–18, Oct. 2018, doi: 10.1080/10630732.2018.1493883.
- [12] 'NHTSA en Español'. Accessed: Mar. 28, 2023. [Online]. Available: <https://www.nhtsa.gov/es>
- [13] D. Parekh *et al.*, 'A Review on Autonomous Vehicles: Progress, Methods and Challenges', *Electronics 2022, Vol. 11, Page 2162*, vol. 11, no. 14, p. 2162, Jul. 2022, doi: 10.3390/ELECTRONICS11142162.

- [14] R. Gupta, S. Tanwar, N. Kumar, and S. Tyagi, 'Blockchain-based security attack resilience schemes for autonomous vehicles in industry 4.0: A systematic review', *Computers & Electrical Engineering*, vol. 86, p. 106717, Sep. 2020, doi: 10.1016/J.COMPELECENG.2020.106717.
- [15] 'SAE Levels of Driving Automation™ Refined for Clarity and International Audience'. Accessed: Mar. 29, 2023. [Online]. Available: <https://www.sae.org/blog/sae-j3016-update>
- [16] A. Faisal, T. Yigitcanlar, M. Kamruzzaman, and G. Currie, 'Understanding autonomous vehicles: A systematic literature review on capability, impact, planning and policy', *J Transp Land Use*, vol. 12, no. 1, pp. 45–72, 2019, doi: 10.5198/JTLU.2019.1405.
- [17] M. De Ryck, M. Versteyhe, and F. Debrouwere, 'Automated guided vehicle systems, state-of-the-art control algorithms and techniques', *Journal of Manufacturing Systems*, vol. 54. Elsevier B.V., pp. 152–173, Jan. 01, 2020. doi: 10.1016/j.jmsy.2019.12.002.
- [18] F. Bonin-Font, A. Ortiz, and G. Oliver, 'Visual navigation for mobile robots: A survey', *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 53, no. 3, pp. 263–296, Nov. 2008, doi: 10.1007/S10846-008-9235-4/METRICS.
- [19] G. Fragapane, R. de Koster, F. Sgarbossa, and J. O. Strandhagen, 'Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda', *European Journal of Operational Research*, vol. 294, no. 2. Elsevier B.V., pp. 405–426, Oct. 16, 2021. doi: 10.1016/j.ejor.2021.01.019.
- [20] M. B. Alatise and G. P. Hancke, 'A Review on Challenges of Autonomous Mobile Robot and Sensor Fusion Methods', *IEEE Access*, vol. 8, pp. 39830–39846, 2020, doi: 10.1109/ACCESS.2020.2975643.
- [21] N. G. Hockstein, C. G. Gourin, R. A. Faust, and D. J. Terris, 'A history of robots: from science fiction to surgical robotics', *J Robot Surg*, vol. 1, no. 2, p. 113, 2007, doi: 10.1007/S11701-007-0021-2.
- [22] L. Lynch, J. Coleman, T. Newe, J. Walsh, J. Clifford, and D. Toal, 'Automated Ground Vehicle (AGV) and Sensor Technologies- A Review; Automated Ground Vehicle (AGV) and Sensor Technologies- A Review', *2018 12th International Conference on Sensing Technology (ICST)*, 2018.
- [23] A. A. Tubis and H. Poturaj, 'Risk Related to AGV Systems—Open-Access Literature Review', *Energies* 2022, Vol. 15, Page 8910, vol. 15, no. 23, p. 8910, Nov. 2022, doi: 10.3390/EN15238910.
- [24] W. Nakimuli, J. Garcia-Reinoso, J. Enrique Sierra-Garcia, P. Serrano, and I. Quintana Fernandez, 'Deployment and Evaluation of an Industry 4.0 Use Case over 5G; Deployment and Evaluation of an Industry 4.0 Use Case over 5G', *IEEE Communications Magazine*, vol. 59, 2021, doi: 10.1109/MCOM.001.2001104.
- [25] A. Fellan, C. Schellenberger, M. Zimmermann, and H. D. Schotten, 'Enabling Communication Technologies for Automated Unmanned Vehicles in Industry 4.0', 2018.

- [26] A. Martínez-gutiérrez, J. Díez-gonzález, R. Ferrero-guillén, P. Verde, R. Álvarez, and H. Perez, 'Digital twin for automatic transportation in industry 4.0', *Sensors*, vol. 21, no. 10, May 2021, doi: 10.3390/s21103344.
- [27] B. A. Kadir and O. Broberg, 'Human well-being and system performance in the transition to industry 4.0', *Int J Ind Ergon*, vol. 76, Mar. 2020, doi: 10.1016/j.ergon.2020.102936.
- [28] R. Cupek *et al.*, 'Autonomous guided vehicles for smart industries – the state-of-the-art and research challenges', in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Science and Business Media Deutschland GmbH, 2020, pp. 330–343. doi: 10.1007/978-3-030-50426-7_25.
- [29] G. Fragapane, · Dmitry Ivanov, M. Peron, F. Sgarbossa, and J. O. Strandhagen, 'Increasing flexibility and productivity in Industry 4.0 production networks with autonomous mobile robots and smart intralogistics', *Ann Oper Res*, vol. 308, pp. 125–143, 2022, doi: 10.1007/s10479-020-03526-7.
- [30] S. Cebollada, L. Payá, M. Flores, A. Peidró, and O. Reinoso, 'A state-of-the-art review on mobile robotics tasks using artificial intelligence and visual data', *Expert Syst Appl*, vol. 167, Apr. 2021, doi: 10.1016/J.ESWA.2020.114195.
- [31] H. Ning, R. Yin, A. Ullah, F. S.-I. T. on Intelligent, and undefined 2021, 'A survey on hybrid human-artificial intelligence for autonomous driving', *ieeexplore.ieee.org*, 2022, Accessed: Apr. 07, 2023. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9423526/>
- [32] C. Wilson, F. Marchetti, M. Di Carlo, A. Riccardi, and E. Minisci, 'Classifying intelligence in machines: A taxonomy of intelligent control', *Robotics*, vol. 9, no. 3, Sep. 2020, doi: 10.3390/ROBOTICS9030064.
- [33] T. De Bruin, J. Kober, K. Tuyls, and R. Babuška, 'The importance of experience replay database composition in deep reinforcement learning', 2015.
- [34] '¿Qué es el Transfer Learning?' Accessed: Jun. 01, 2023. [Online]. Available: <https://datascientest.com/es/que-es-el-transfer-learning>
- [35] 'Transfer Learning | Understanding Transfer Learning for Deep Learning'. Accessed: Jun. 01, 2023. [Online]. Available: <https://www.analyticsvidhya.com/blog/2021/10/understanding-transfer-learning-for-deep-learning/>
- [36] 'Deep Learning con redes pre-entrenadas en ImageNet | by Daniel Lerch | Neuron4 | Medium'. Accessed: Jun. 01, 2023. [Online]. Available: <https://medium.com/neuron4/redes-pre-entrenadas-en-imagenet-30d858c37b1f>
- [37] B. Namatherdhala, N. Mazher, G. Krishna Sriram, and G. Krishna Sriram, 'Uses of Artificial Intelligence in Autonomous Driving and V2X communication', *researchgate.net*, 2022.
- [38] J. Li, H. Gang, H. Ma, M. Tomizuka, and C. Choi, 'Important Object Identification with Semi-Supervised Learning for Autonomous Driving', in *Proceedings - IEEE International*

- Conference on Robotics and Automation*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 2913–2919. doi: 10.1109/ICRA46639.2022.9812234.
- [39] J. Zhang, H. Liu, and J. Lu, 'A semi-supervised 3D object detection method for autonomous driving', *Displays*, vol. 71, Jan. 2022, doi: 10.1016/j.displa.2021.102117.
- [40] A. Maarif, W. Rahmaniari, M. A. M. Vera, A. A. Nuryono, R. Majdoubi, and A. Cakan, 'Artificial Potential Field Algorithm for Obstacle Avoidance in UAV Quadrotor for Dynamic Environment', *10th IEEE International Conference on Communication, Networks and Satellite, Comnetsat 2021 - Proceedings*, pp. 184–189, Jul. 2021, doi: 10.1109/COMNETSAT53002.2021.9530803.
- [41] Y. Li, C. Le Bihan, T. Pourtau, and T. Ristorcelli, 'InsClustering: Instantly Clustering LiDAR Range Measures for Autonomous Vehicle; InsClustering: Instantly Clustering LiDAR Range Measures for Autonomous Vehicle', 2020, doi: 10.1109/ITSC45102.2020.9294467.
- [42] R. Domínguez, E. Onieva, J. Alonso, J. Villagra, and C. González, 'LIDAR based Perception Solution for Autonomous Vehicles', 2011, p. 1402.
- [43] W. Jiang, C. Song, H. Wang, M. Yu, and Y. Yan, 'Obstacle Detection by Autonomous Vehicles: An Adaptive Neighborhood Search Radius Clustering Approach', *Machines*, vol. 11, no. 1, Jan. 2023, doi: 10.3390/machines11010054.
- [44] V. Balaska, L. Bampis, M. Boudourides, and A. Gasteratos, 'Unsupervised semantic clustering and localization for mobile robotics tasks', *Rob Auton Syst*, vol. 131, Sep. 2020, doi: 10.1016/j.robot.2020.103567.
- [45] T. Kondo, 'A reinforcement learning with evolutionary state recruitment strategy for autonomous mobile robots control', *Elsevier*, 2004, Accessed: Apr. 09, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0921889003001842>
- [46] J. A. Placed and J. A. Castellanos, 'A deep reinforcement learning approach for active SLAM', *Applied Sciences (Switzerland)*, vol. 10, no. 23, pp. 1–21, Dec. 2020, doi: 10.3390/app10238386.
- [47] S. Bhatti, A. Desmaison, O. Miksik, N. Nardelli, N. Siddharth, and P. H. S. Torr, 'Playing Doom with SLAM-Augmented Deep Reinforcement Learning', Dec. 2016, [Online]. Available: <http://arxiv.org/abs/1612.00380>
- [48] J. Gao, W. Ye, J. Guo, and Z. Li, 'Deep reinforcement learning for indoor mobile robot path planning', *Sensors (Switzerland)*, vol. 20, no. 19, pp. 1–15, Oct. 2020, doi: 10.3390/s20195493.
- [49] R. Kamoshida and Y. Kazama, 'Acquisition of Automated Guided Vehicle Route Planning Policy Using Deep Reinforcement Learning', in *6th IEEE International Conference on Advanced Logistics and Transport, ICALT 2017 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., Nov. 2018, pp. 35–40. doi: 10.1109/ICAdLT.2017.8547000.
- [50] J. Xin, H. Zhao, D. Liu, and M. Li, 'Application of Deep Reinforcement Learning in Mobile Robot Path Planning', 2017.

- [51] S. Wen, Y. Zhao, X. Yuan, Z. Wang, D. Zhang, and L. Manfredi, 'Path planning for active SLAM based on deep reinforcement learning under unknown environments', *Intell Serv Robot*, vol. 13, no. 2, pp. 263–272, Apr. 2020, doi: 10.1007/s11370-019-00310-w.
- [52] S. Wen, X. Chen, C. Ma, H. K. Lam, and S. Hua, 'The Q-learning obstacle avoidance algorithm based on EKF-SLAM for NAO autonomous walking under unknown environments', *Rob Auton Syst*, vol. 72, pp. 29–36, Oct. 2015, doi: 10.1016/j.robot.2015.04.003.
- [53] E. S. Low, P. Ong, and K. C. Cheah, 'Solving the optimal path planning of a mobile robot using improved Q-learning', *Rob Auton Syst*, vol. 115, pp. 143–161, May 2019, doi: 10.1016/j.robot.2019.02.013.
- [54] B. Khamidehi and E. Sousa, 'A double Q-learning approach for navigation of aerial vehicles with connectivity constraint', in *ICC 2020-2020 IEEE*, 2020. Accessed: Apr. 14, 2023. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9148608/>
- [55] J. Yu, Y. Su, and Y. Liao, 'The Path Planning of Mobile Robot by Neural Networks and Hierarchical Reinforcement Learning', *Front Neurobot*, vol. 14, Oct. 2020, doi: 10.3389/fnbot.2020.00063.
- [56] N. Botteghi, B. Sirmacek, R. Schulte, M. Poel, and C. Brune, 'Reinforcement learning helps slam: Learning to build maps', in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, International Society for Photogrammetry and Remote Sensing, Aug. 2020, pp. 329–336. doi: 10.5194/isprs-archives-XLIII-B4-2020-329-2020.
- [57] D. Tamilselvi, SM Shalinie, and G. Nirmala, 'Q learning for mobile robot navigation in indoor environment', in *IEEE-International Conference on Recent Trends in Information Technology, ICRTIT 2011*, 2011. Accessed: Apr. 14, 2023. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/5972477/>
- [58] G. Kontoudis and K. Vamvoudakis, 'Kinodynamic motion planning with continuous-time Q-learning: An online, model-free, and safe navigation framework', *IEEE Trans Neural Netw Learn Syst*, vol. 30, 2019, Accessed: Apr. 14, 2023. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8677294/>
- [59] N. T. T. Van, N. M. Tien, N. M. Cuong, H. T. K. Duyen, and N. D. Duy, 'Constructing an Intelligent Navigation System for Autonomous Mobile Robot Based on Deep Reinforcement Learning', in *Studies in Computational Intelligence*, vol. 981, Springer Science and Business Media Deutschland GmbH, 2021, pp. 251–261. doi: 10.1007/978-3-030-76620-7_22.
- [60] P. Sadhukhan and R. R. Selmic, 'Proximal policy optimization for formation navigation and obstacle avoidance', *Int J Intell Robot Appl*, vol. 6, no. 4, pp. 746–759, Dec. 2022, doi: 10.1007/S41315-022-00245-Z.
- [61] A. Staroverov and A. I. Panov, 'Landmark Policy Optimization for Object Navigation Task', Sep. 2021, [Online]. Available: <http://arxiv.org/abs/2109.09512>
- [62] A. Staroverov and A. Panov, 'Hierarchical landmark policy optimization for visual indoor navigation', *ieeexplore.ieee.org*, 2022, Accessed: Apr. 14, 2023. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9795006/>

- [63] N. Toan and KG Woo, 'Mapless navigation with deep reinforcement learning based on the convolutional proximal policy optimization network', in *International Conference on Big Data and Smart Computing (BIGCOMP)*, 2021. Accessed: Apr. 14, 2023. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9373298/>
- [64] X. Ruan, D. Ren, X. Zhu, and J. Huang, 'Mobile Robot Navigation based on Deep Reinforcement Learning', in *Proceedings of the 31st Chinese Control and Decision Conference (2019 CCDC) : 3-5 June, 2019, Nanchang, China*, 2019.
- [65] E. Marchesini and A. Farinelli, 'Discrete Deep Reinforcement Learning for Mapless Navigation; Discrete Deep Reinforcement Learning for Mapless Navigation', in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020. doi: 10.0/Linux-x86_64.
- [66] K. Yokoyama and K. Morioka, 'Autonomous Mobile Robot with Simple Navigation System Based on Deep Reinforcement Learning and a Monocular Camera', in *IEEE/SICE International Symposium on System Integration*, 2020, p. 1405.
- [67] H. Surmann, C. Jestel, R. Marchel, F. Musberg, H. Elhadj, and M. Ardani, 'Deep Reinforcement learning for real autonomous mobile robot navigation in indoor environments', May 2020, [Online]. Available: <http://arxiv.org/abs/2005.13857>
- [68] J. C. Jesus, J. A. Bottega, M. A. S. L. Cuadros, and D. F. T. Gamarra, *Deep Deterministic Policy Gradient for Navigation of Mobile Robots in Simulated Environments*. 2019. doi: 10.0/Linux-x86_64.
- [69] F. Wang *et al.*, 'Intervention aided reinforcement learning for safe and practical policy optimization in navigation', in *2nd Conference on Robot Learning (CoRL 2018)*, 2018. Accessed: Apr. 14, 2023. [Online]. Available: <http://proceedings.mlr.press/v87/wang18a.html>
- [70] M. Chowdhury and F. Erden, 'RSS-based Q-learning for indoor UAV navigation', in *MILCOM 2019-2019 IEEE*, 2019. Accessed: Apr. 14, 2023. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9020894/>
- [71] L. Khriji, F. Touati, ... K. B.-I. J. of, and undefined 2011, 'Mobile robot navigation based on Q-learning technique', *journals.sagepub.com*, vol. 8, no. 1, pp. 45–51, 2011, Accessed: Apr. 14, 2023. [Online]. Available: <https://journals.sagepub.com/doi/pdf/10.5772/10528>
- [72] T. Ribeiro, F. Goncalves, I. Garcia, G. Lopes, and A. F. Ribeiro, 'Q-Learning for Autonomous Mobile Robot Obstacle Avoidance', in *IEEE International Conference on Autonomous Robot Systems and Competitions*, 2019.
- [73] B.-Q. Huang, G.-Y. Cao, and Min Guo, 'Reinforcement learning neural network to the problem of autonomous mobile robot obstacle avoidance', *ieeexplore.ieee.org*, 2005, Accessed: Apr. 09, 2023. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/1526924/>
- [74] M. Duguleana, G. M.-E. S. with Applications, and undefined 2016, 'Neural networks based reinforcement learning for mobile robots obstacle avoidance', *Elsevier*, Accessed: Apr. 09, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0957417416303001>

- [75] C. C. E. Chewu and V. Manoj Kumar, 'Autonomous navigation of a mobile robot in dynamic indoor environments using SLAM and reinforcement learning', in *IOP Conference Series: Materials Science and Engineering*, Institute of Physics Publishing, 2018. doi: 10.1088/1757-899X/402/1/012022.
- [76] P. Mohanty, A. Sah, V. Kumar, and S. Kundu, 'Application of deep Q-learning for wheel mobile robot navigation', in *2017 International Conference on Computational Intelligence and Networks*, 2017. Accessed: Apr. 14, 2023. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8307337/>
- [77] H. Wicaksono, 'Q learning behavior on autonomous navigation of physical robot', in *8th International Conference on Ubiquitous Robots and Ambient Intelligence (URA I)*, 2011. Accessed: Apr. 14, 2023. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/6145931/>
- [78] J. Popper, V. Yfantis, and M. Ruskowski, 'Simultaneous Production and AGV Scheduling using Multi-Agent Deep Reinforcement Learning', in *Procedia CIRP*, Elsevier B.V., 2021, pp. 1523–1528. doi: 10.1016/j.procir.2021.11.257.
- [79] K. Takahashi and S. Tomah, 'Online optimization of AGV transport systems using deep reinforcement learning', 2020. [Online]. Available: www.bnccs.org,
- [80] P. A. Corrales and F. A. Gregori, 'Swarm AGV Optimization Using Deep Reinforcement Learning', 2020, doi: 10.1145/3426826.3426839.
- [81] D. Kozjek, A. Malus, and R. Vrabič, 'Reinforcement-learning-based route generation for heavy-traffic autonomous mobile robot systems', *Sensors*, vol. 21, no. 14, Jul. 2021, doi: 10.3390/s21144809.
- [82] Y. Lecun, Y. Bengio, and G. Hinton, 'Deep learning', *Nature*, vol. 521, no. 7553. Nature Publishing Group, pp. 436–444, May 27, 2015. doi: 10.1038/nature14539.
- [83] W. Zhang, K. Liu, W. Zhang, Y. Zhang, and J. Gu, 'Deep Neural Networks for wireless localization in indoor and outdoor environments', *Neurocomputing*, vol. 194, pp. 279–287, Jun. 2016, doi: 10.1016/j.neucom.2016.02.055.
- [84] K. Wu, H. Wang, M. A. Esfahani, and S. Yuan, 'Achieving Real-Time Path Planning in Unknown Environments Through Deep Neural Networks', *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 2093–2102, Mar. 2022, doi: 10.1109/TITS.2020.3031962.
- [85] R. Chai, A. Tsourdos, A. Savvaris, S. Chai, Y. Xia, and C. L. P. Chen, 'Six-DOF Spacecraft Optimal Trajectory Planning and Real-Time Attitude Control: A Deep Neural Network-Based Approach', *IEEE Trans Neural Netw Learn Syst*, vol. 31, no. 11, pp. 5005–5013, Nov. 2020, doi: 10.1109/TNNLS.2019.2955400.
- [86] J. Lai, Z. Ren, Z. Wu, Y. Liu, and S. Xie, 'Deep Neural Network-Based Real-time Trajectory Planning for an Automatic Guided Vehicle with Obstacles', in *Proceedings - 2020 Chinese Automation Congress, CAC 2020*, Institute of Electrical and Electronics Engineers Inc., Nov. 2020, pp. 6311–6316. doi: 10.1109/CAC51589.2020.9327280.
- [87] Z. Ren, J. Lai, Z. Wu, and S. Xie, 'Deep neural networks-based real-time optimal navigation for an automatic guided vehicle with static and dynamic obstacles',

- Neurocomputing*, vol. 443, pp. 329–344, Jul. 2021, doi: 10.1016/j.neucom.2021.02.034.
- [88] R. Chai, A. Tsourdos, A. Savvaris, S. Chai, Y. Xia, and C. L. Philip Chen, ‘Design and Implementation of Deep Neural Network-Based Control for Automatic Parking Maneuver Process’, *IEEE Trans Neural Netw Learn Syst*, vol. 33, no. 4, pp. 1400–1413, Apr. 2022, doi: 10.1109/TNNLS.2020.3042120.
- [89] C. Badgajar, S. Das, D. M. Figueroa, D. Flippo, and S. Welch, ‘Deep neural networks to predict autonomous ground vehicle behavior on sloping terrain field’, *J Field Robot*, Jun. 2023, doi: 10.1002/rob.22163.
- [90] P. Krapež, M. Vidmar, and M. Munih, ‘Distance measurements in uwb-radio localization systems corrected with a feedforward neural network model’, *Sensors*, vol. 21, no. 7, Apr. 2021, doi: 10.3390/s21072294.
- [91] A. Payal, C. S. Rai, and B. V. R. Reddy, ‘Analysis of Some Feedforward Artificial Neural Network Training Algorithms for Developing Localization Framework in Wireless Sensor Networks’, *Wirel Pers Commun*, vol. 82, no. 4, pp. 2519–2536, Jun. 2015, doi: 10.1007/s11277-015-2362-x.
- [92] S. Chen, C.-Y. Wen, Y. Zou, and W. Chen, ‘Stereo Visual Inertial Pose Estimation Based on Feedforward-Feedback Loops’, Jul. 2020, [Online]. Available: <http://arxiv.org/abs/2007.02250>
- [93] C. C. Lee, M. H. F. Rahiman, R. A. Rahim, and F. S. A. Saad, ‘A Deep Feedforward Neural Network Model for Image Prediction’, in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Jun. 2021. doi: 10.1088/1742-6596/1878/1/012062.
- [94] K. Kolegain, F. Leonard, S. Zimmer-Chevret, A. Ben Attar, and G. Abba, ‘A feedforward deflection compensation scheme coupled with an offline path planning for robotic friction stir welding’, Elsevier B.V., Jan. 2018, pp. 728–733. doi: 10.1016/j.ifacol.2018.08.405.
- [95] T. L. Bui, P. T. Doan, H. K. Kim, and S. B. Kim, ‘Trajectory tracking controller design for AGV using laser sensor based positioning system’, in *2013 9th Asian Control Conference, ASCC 2013*, 2013. doi: 10.1109/ASCC.2013.6606202.
- [96] A. Pandey, V. S. Panwar, M. Ehtesham Hasan, and D. R. Parhi, ‘V-REP-based navigation of automated wheeled robot between obstacles using PSO-tuned feedforward neural network’, *J Comput Des Eng*, vol. 7, no. 4, pp. 427–434, Aug. 2020, doi: 10.1093/jcde/qwaa035.
- [97] N. Barakat and R. Rajagopalan, ‘Speed Control of a DC Motor Using a Feedforward Computed Torque Control Scheme’, in *Proceedings of the 1996 IEEE International Symposium on Intelligent Contro*, 1996.
- [98] N. R. Kapania and J. C. Gerdes, ‘Design of a feedback-feedforward steering controller for accurate path tracking and stability at the limits of handling’, *Vehicle System Dynamics*, vol. 53, no. 12, pp. 1687–1704, Dec. 2015, doi: 10.1080/00423114.2015.1055279.

- [99] M.-T. Laghmar, T. Boudali, J. Laurain, R. Ledy, J.-P. Orjuela, and M. B. Lauffenburger, *Obstacle Avoidance, Path Planning and Control for Autonomous Vehicles; Obstacle Avoidance, Path Planning and Control for Autonomous Vehicles*. 2019. doi: 10.0/Linux-x86_64.
- [100] J. I. Suráez, B. M. Vinagre, and Yangquan Chen, 'A fractional adaptation scheme for lateral control of an AGV', in *JVC/Journal of Vibration and Control*, Sep. 2008, pp. 1499–1511. doi: 10.1177/1077546307087434.
- [101] L. Bian, L. Liu, and Y. Zhou, 'Lateral Controller with Feedforward Compensator for Autonomous Ground Vehicle Tracking Path on Sloped Terrain', in *IEEE Vehicular Technology Conference*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/VTC2022-Fall57202.2022.10012771.
- [102] K. Sailan and Klaus. D. Kuhnert, 'Modeling and Design of Cruise Control System with Feedforward for all Terrian Vehicles', Academy and Industry Research Collaboration Center (AIRCC), Nov. 2013, pp. 339–349. doi: 10.5121/csit.2013.3828.
- [103] H. Luu *et al.*, 'Feedforward Structure Of Kalman Filters For Low Cost Navigation', 2007. [Online]. Available: <https://www.researchgate.net/publication/234004065>
- [104] P. Gao *et al.*, 'Improved Position Estimation Algorithm of Agricultural Mobile Robots Based on Multisensor Fusion and Autoencoder Neural Network', *Sensors*, vol. 22, no. 4, Feb. 2022, doi: 10.3390/s22041522.
- [105] A. Schlichting and U. Feuerhake, 'Global Vehicle Localization by Sequence Analysis Using LiDARFeatures Derived by an Autoencoder', in *IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2018.
- [106] B. Chidlovskii and L. Antsfeld, 'Semi-supervised Variational Autoencoder for WiFiIndoor Localization', in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2019.
- [107] J. Liu, N. Liu, Z. Pan, and X. You, 'AutLoc: Deep Autoencoder for IndoorLocalization with RSS Fingerprinting', in *10th International Conference on Wireless Communications and Signal Processing (WCSP) : October 18-20, Hangzhou, China.*, 2018.
- [108] H. Xu, H. Luo, Z. Wu, F. Wu, L. Bao, and F. Zhao, 'Towards Predicting the Measurement Noise Covariance with a Transformer and Residual Denoising Autoencoder for GNSS/INS Tightly-Coupled Integrated Navigation', *Remote Sens (Basel)*, vol. 14, no. 7, Apr. 2022, doi: 10.3390/rs14071691.
- [109] M. Bianchi and T. D. Barfoot, 'UAV Localization Using Autoencoded Satellite Images', *IEEE Robot Autom Lett*, vol. 6, no. 2, pp. 1761–1768, Apr. 2021, doi: 10.1109/LRA.2021.3060397.
- [110] A. Papachristodoulou, C. Kyrkou, and T. Theocharides, 'DriveGuard: Robustification of Automated Driving Systems with Deep Spatio-Temporal Convolutional Autoencoder', in *WACV*, 2021.
- [111] M. Leonardi and A. Stahl, 'Convolutional Autoencoder aided loop closure detection for monocular SLAM *', in *IFAC*, Elsevier B.V., Jan. 2018, pp. 159–164. doi: 10.1016/j.ifacol.2018.09.486.

- [112] Z. Wang, Z. Peng, Y. Guan, and L. Wu, 'Manifold Regularization Graph Structure Auto-Encoder to Detect Loop Closure for Visual SLAM', *IEEE Access*, vol. 7, pp. 59524–59538, 2019, doi: 10.1109/ACCESS.2019.2914943.
- [113] Y. Luo, Y. Xiao, Y. Zhang, and N. Zeng, 'Detection of loop closure in visual SLAM: a stacked assorted auto-encoder based approach', *Optoelectron Lett*, vol. 17, no. 6, pp. 354–360, Jun. 2021, doi: 10.1007/s11801-021-0156-9.
- [114] D. N. Jagadish, A. Chauhan, and L. Mahto, 'Autonomous Vehicle Path Prediction Using Conditional Variational Autoencoder Networks', in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Science and Business Media Deutschland GmbH, 2021, pp. 129–139. doi: 10.1007/978-3-030-75762-5_11.
- [115] D. N. Jagadish, A. Chauhan, and L. Mahto, 'Conditional Variational Autoencoder Networks for Autonomous Vehicle Path Prediction', *Neural Process Lett*, vol. 54, no. 5, pp. 3965–3978, Oct. 2022, doi: 10.1007/s11063-022-10802-z.
- [116] X. Feng, Z. Cen, J. Hu, and Y. Zhang, 'Vehicle Trajectory Prediction Using Intention-based Conditional Variational Autoencoder', in *2019 IEEE Intelligent Transportation Systems Conference, ITSC 2019*, Institute of Electrical and Electronics Engineers Inc., Oct. 2019, pp. 3514–3519. doi: 10.1109/ITSC.2019.8917482.
- [117] D. Huang *et al.*, 'A Variational Autoencoder Based Generative Model of Urban Human Mobility', in *Proceedings - 2nd International Conference on Multimedia Information Processing and Retrieval, MIPR 2019*, Institute of Electrical and Electronics Engineers Inc., Apr. 2019, pp. 425–430. doi: 10.1109/MIPR.2019.00086.
- [118] C. Walker, P. Graham, and A. Philippides, 'Using deep autoencoders to investigate image matching in visual navigation', in *Conference on Biomimetic and Biohybrid Systems, 2017*, pp. 465–474. [Online]. Available: <http://www.creative-robotics.com/?q=unibot>
- [119] X. Gu, Y. Han, and J. Yu, 'A novel lane-changing decision model for autonomous vehicles based on deep autoencoder network and XGBoost', *IEEE Access*, vol. 8, pp. 9846–9863, 2020, doi: 10.1109/ACCESS.2020.2964294.
- [120] A. Amini, W. Schwarting, G. Rosman, B. Araki, S. Karaman, and D. Rus, 'Variational Autoencoder for End-to-End Control of Autonomous Driving with Novelty Detection and Training De-biasing', in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018.
- [121] C. Gebauer and M. Bennewitz, 'The Pitfall of More Powerful Autoencoders in Lidar-Based Navigation', Feb. 2021, [Online]. Available: <http://arxiv.org/abs/2102.02127>
- [122] C. Richter and N. Roy, 'Safe visual navigation via deep learning and novelty detection', in *Robotics: Science and Systems*, MIT Press Journals, 2017. doi: 10.15607/rss.2017.xiii.064.
- [123] L. P. Perera and B. Mo, 'Ship performance and navigation data compression and communication under autoencoder system architecture', *Journal of Ocean Engineering and Science*, vol. 3, no. 2, pp. 133–143, Jun. 2018, doi: 10.1016/j.joes.2018.04.002.

- [124] S. Atakishiyev, M. Salameh, H. Yao, and R. Goebel, 'Explainable Artificial Intelligence for Autonomous Driving: A Comprehensive Overview and Field Guide for Future Research Directions', *Elsevier*, 2023.
- [125] M. Ballesta, L. Payá, S. Cebollada, O. Reinoso, and F. Murcia, 'A cnn regression approach to mobile robot localization using omnidirectional images', *Applied Sciences (Switzerland)*, vol. 11, no. 16, Aug. 2021, doi: 10.3390/app11167521.
- [126] T. H. Wang, H. J. Huang, J. T. Lin, C. W. Hu, K. H. Zeng, and M. Sun, 'Omnidirectional CNN for Visual Place Recognition and Navigation', in *Proceedings - IEEE International Conference on Robotics and Automation*, Institute of Electrical and Electronics Engineers Inc., Sep. 2018, pp. 2341–2348. doi: 10.1109/ICRA.2018.8463173.
- [127] S. Xu, W. Chou, and H. Dong, 'A robust indoor localization system integrating visual localization aided by CNN-based image retrieval with Monte Carlo localization', *Sensors (Switzerland)*, vol. 19, no. 2, Jan. 2019, doi: 10.3390/s19020249.
- [128] M. Ibrahim, M. Torki, and M. ElNainay, 'CNN based Indoor Localization using RSSTime-Series', in *IEEE Symposium on Computers and Communications (ISCC)*, 2018.
- [129] W. Njima, W. Njima, R. Zayani, M. Terre, and R. Bouallegue, 'Deep cnn for indoor localization in iot-sensor systems', *Sensors (Switzerland)*, vol. 19, no. 14, Jul. 2019, doi: 10.3390/s19143127.
- [130] S. Ito, S. Hiratsuka, M. Ohta, H. Matsubara, and M. Ogawa, 'Small imaging depth LIDAR and DCNN-based localization for automated guided vehicle', *Sensors (Switzerland)*, vol. 18, no. 1, Jan. 2018, doi: 10.3390/s18010177.
- [131] S. Ito, M. Soga, S. Hiratsuka, H. Matsubara, and M. Ogawa, 'Quality index of supervised data for convolutional neural network-based localization', *Applied Sciences (Switzerland)*, vol. 9, no. 10, May 2019, doi: 10.3390/app9101983.
- [132] M. Afif, R. Ayachi, Y. Said, E. Pissaloux, and M. Atri, 'Indoor Object Classification for Autonomous Navigation Assistance Based on Deep CNN Model', in *2019 IEEE International Symposium on Measurements and Networking, M and N 2019 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., Jul. 2019. doi: 10.1109/IWMN.2019.8805042.
- [133] S. Protasov *et al.*, 'CNN-based Omnidirectional Object Detection for HermesBot Autonomous Delivery Robot with Preliminary Frame Classification', in *2021 20th International Conference on Advanced Robotics, ICAR 2021*, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 517–522. doi: 10.1109/ICAR53236.2021.9659386.
- [134] H. Gao, B. Cheng, J. Wang, K. Li, J. Zhao, and D. Li, 'Object Classification Using CNN-Based Fusion of Vision and LIDAR in Autonomous Vehicle Environment', *IEEE Trans Industr Inform*, vol. 14, no. 9, pp. 4224–4230, Sep. 2018, doi: 10.1109/TII.2018.2822828.
- [135] G. Liu, R. Zhang, Y. Wang, and R. Man, 'Road scene recognition of forklift agv equipment based on deep learning', *Processes*, vol. 9, no. 11, Nov. 2021, doi: 10.3390/pr9111955.

- [136] Y. Lyu and X. Huang, 'Road Segmentation Using CNN with GRU', Apr. 2018, [Online]. Available: <http://arxiv.org/abs/1804.05164>
- [137] S. Chiodini, L. Torresin, and M. Pertile, 'Evaluation of 3D CNN Semantic Mapping for Rover Navigation', in *2020 IEEE International Workshop on Metrology for AeroSpace : virtual conference*, , 2020.
- [138] L. Deng, M. Yang, Y. Qian, C. Wang, and B. Wang, 'CNN based semantic segmentation for urban traffic scenes using fisheye camera', in *IEEE Intelligent Vehicles Symposium, Proceedings*, Institute of Electrical and Electronics Engineers Inc., Jul. 2017, pp. 231–236. doi: 10.1109/IVS.2017.7995725.
- [139] S. Yang, W. Wang, C. Liu, and W. Deng, 'Scene understanding in deep learning-based end-to-end controllers for autonomous vehicles', *IEEE Trans Syst Man Cybern Syst*, vol. 49, no. 1, pp. 53–63, Jan. 2019, doi: 10.1109/TSMC.2018.2868372.
- [140] F. Foroughi, Z. Chen, and J. Wang, 'A CNN-based system for mobile robot navigation in indoor environments via visual localization with a small dataset', *World Electric Vehicle Journal*, vol. 12, no. 3, 2021, doi: 10.3390/wevj12030134.
- [141] H. Qin, M. Huang, J. Cao, and X. Zhang, 'Loop closure detection in SLAM by combining visual CNN features and submaps', in *Proceedings - 2018 4th International Conference on Control, Automation and Robotics, ICCAR 2018*, Institute of Electrical and Electronics Engineers Inc., Jun. 2018, pp. 426–430. doi: 10.1109/ICCAR.2018.8384713.
- [142] D. Song, G. M. Tian, and J. Liu, 'Real-time localization measure and perception detection using multi-sensor fusion for Automated Guided Vehicles', in *Chinese Control Conference, CCC*, IEEE Computer Society, Jul. 2021, pp. 3219–3224. doi: 10.23919/CCC52363.2021.9550235.
- [143] W. Liu, W. Sun, and Y. Liu, 'DLOAM: Real-time and Robust LiDAR SLAM System Based on CNN in Dynamic Urban Environments', *IEEE Open Journal of Intelligent Transportation Systems*, pp. 1–1, Sep. 2021, doi: 10.1109/ojits.2021.3109423.
- [144] Z. Zhang, J. Zhang, and Q. Tang, 'Mask R-CNN Based Semantic RGB-D SLAM for Dynamic Scenes; Mask R-CNN Based Semantic RGB-D SLAM for Dynamic Scenes', in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2019. doi: 10.0/Linux-x86_64.
- [145] A. Torres Angonese and P. Fernando Ferreira Rosa, 'Multiple People Detection and Identification System Integrated with a Dynamic Simultaneous Localization and Mapping System for an Autonomous Mobile Robotic Platform', in *2017 International Conference on Military Technologies (ICMT).*, IEEE, 2017.
- [146] A. Steenbeek and F. Nex, 'CNN-Based Dense Monocular Visual SLAM for Real-Time UAV Exploration in Emergency Conditions', *Drones*, vol. 6, no. 3, Mar. 2022, doi: 10.3390/drones6030079.
- [147] S. J. Lee, H. Choi, and S. S. Hwang, 'Real-time Depth Estimation Using Recurrent CNN with Sparse Depth Cues for SLAM System', *Int J Control Autom Syst*, vol. 18, no. 1, pp. 206–216, Jan. 2020, doi: 10.1007/s12555-019-0350-8.

- [148] Y. Jin, L. Yu, Z. Chen, and S. Fei, 'A Mono SLAM Method Based on Depth Estimation by DenseNet-CNN', *IEEE Sens J*, vol. 22, no. 3, pp. 2447–2455, Feb. 2022, doi: 10.1109/JSEN.2021.3134014.
- [149] K. Tateno, F. Tombari, I. Laina, and N. Navab, 'CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction', in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [150] P. Karpyshev *et al.*, 'MuCaSLAM: CNN-Based Frame Quality Assessment for Mobile Robot with Omnidirectional Visual SLAM', in *IEEE International Conference on Automation Science and Engineering*, IEEE Computer Society, 2022, pp. 368–373. doi: 10.1109/CASE49997.2022.9926621.
- [151] Z. Xu, J. Yu, C. Yu, H. Shen, Y. Wang, and H. Yang, 'CNN-based Feature-point Extraction for Real-time Visual SLAM on Embedded FPGA', in *Proceedings - 28th IEEE International Symposium on Field-Programmable Custom Computing Machines, FCCM 2020*, Institute of Electrical and Electronics Engineers Inc., May 2020, pp. 33–37. doi: 10.1109/FCCM48280.2020.00014.
- [152] X. Wang, 'Autonomous Mobile Robot Visual SLAM Based on Improved CNN Method', in *IOP Conference Series: Materials Science and Engineering*, Institute of Physics Publishing, Dec. 2018. doi: 10.1088/1757-899X/466/1/012114.
- [153] H. Bae, G. Kim, J. Kim, D. Qian, and S. Lee, 'Multi-robot path planning method using reinforcement learning', *Applied Sciences (Switzerland)*, vol. 9, no. 15, Aug. 2019, doi: 10.3390/app9153057.
- [154] Z. Ma and D. Wang, 'A CNN Based Q-learning Algorithm for Path Planning of Automated Guided Vehicle', in *2021 IEEE International Conference on Electrical Engineering and Mechatronics Technology, ICEEMT 2021*, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 704–708. doi: 10.1109/ICEEMT52412.2021.9601907.
- [155] D. Sartori, D. Zou, L. Pei, and W. Yu, 'CNN-based path planning on a map', in *2021 IEEE International Conference on Robotics and Biomimetics, ROBIO 2021*, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 1331–1338. doi: 10.1109/ROBIO54168.2021.9739331.
- [156] J. Zhang, Y. Xia, and G. Shen, 'A novel learning-based global path planning algorithm for planetary rovers', *Neurocomputing*, vol. 361, pp. 69–76, Oct. 2019, doi: 10.1016/j.neucom.2019.05.075.
- [157] Y. Lu, W. Wang, and L. Xue, 'A Hybrid CNN-LSTM Architecture for Path Planning of Mobile Robots in Unknow Environments', in *32nd Chinese Control and Decision Conference (CCDC 2020)*, 2020.
- [158] R. Farkh, M. T. Quasim, K. Al Jaloud, S. Alhuwaimel, and S. T. Siddiqui, 'Computer Vision-Control-Based CNN-PID for Mobile Robot', *Computers, Materials & Continua*, vol. 68, no. 1, pp. 1065–1079, Mar. 2021, doi: 10.32604/CMC.2021.016600.
- [159] C. Chuixin and C. Hanxiang, 'AGV Robot Based on Computer Vision and Deep Learning', in *3rd International Conference on Robotics and Automation Sciences (ICRAS)*, IEEE, 2019.

- [160] Y. H. Kim, J. I. Jang, and S. Yun, 'End-to-end deep learning for autonomous navigation of mobile robot', in *2018 IEEE International Conference on Consumer Electronics, ICCE 2018*, Institute of Electrical and Electronics Engineers Inc., Mar. 2018, pp. 1–6. doi: 10.1109/ICCE.2018.8326229.
- [161] V. Rausch, A. Hansen, E. Solowjow, C. Liu, E. Kreuzer, and J. K. Hedrick, 'Learning a deep neural net policy for end-to-end control of autonomous vehicles', in *Proceedings of the American Control Conference*, Institute of Electrical and Electronics Engineers Inc., Jun. 2017, pp. 4914–4919. doi: 10.23919/ACC.2017.7963716.
- [162] D. Teso-Fz-Betoño, E. Zulueta, A. Sánchez-Chica, U. Fernandez-Gamiz, and A. Saenz-Aguirre, 'Semantic segmentation to develop an indoor navigation system for an autonomous mobile robot', *Mathematics*, vol. 8, no. 5, May 2020, doi: 10.3390/MATH8050855.
- [163] Y. Quan, L. Huang, L. Ma, Y. He, and R. Wang, 'Neural Network-Based Indoor Autonomously-Navigated AGV Motion Trajectory Data Fusion', *Automatic Control and Computer Sciences*, vol. 55, no. 4, pp. 334–345, Jul. 2021, doi: 10.3103/S0146411621040076.
- [164] C. Chen, Z. H. Hu, and L. Wang, 'Scheduling of agvs in automated container terminal based on the deep deterministic policy gradient (Ddpq) using the convolutional neural network (cnn)', *J Mar Sci Eng*, vol. 9, no. 12, Dec. 2021, doi: 10.3390/jmse9121439.
- [165] R. Bin Issa *et al.*, 'Double deep Q-learning and faster R-CNN-based autonomous vehicle navigation and obstacle avoidance in dynamic environment', *Sensors*, vol. 21, no. 4, pp. 1–24, Feb. 2021, doi: 10.3390/s21041468.
- [166] C. Liu, B. Zheng, C. Wang, Y. Zhao, S. Fu, and H. Li, 'CNN-based vision model for obstacle avoidance of mobile robot', in *MATEC Web of Conferences*, EDP Sciences, Dec. 2017. doi: 10.1051/mateconf/201713900007.
- [167] P. Chakravarty, K. Kelchtermans, T. Roussel, S. Wellens, T. Tuytelaars, and L. Van Eycken, 'CNN-based single image obstacle avoidance on a quadrotor', in *Proceedings - IEEE International Conference on Robotics and Automation*, Institute of Electrical and Electronics Engineers Inc., Jul. 2017, pp. 6369–6374. doi: 10.1109/ICRA.2017.7989752.
- [168] N.-H. Chang, Y.-H. Chien, H.-H. Chiang, W.-Y. Wang, and C.-C. Hsu, 'A Robot Obstacle Avoidance Method Using Merged CNN Framework', 2019.
- [169] S. Bai, M. Yan, Q. Wan, L. He, X. Wang, and J. Li, 'DL-RNN: An Accurate Indoor Localization Method via Double RNNs', *IEEE Sens J*, vol. 20, no. 1, pp. 286–295, Jan. 2020, doi: 10.1109/JSEN.2019.2936412.
- [170] M. T. Hoang, B. Yuen, X. Dong, T. Lu, R. Westendorp, and K. Reddy, 'Recurrent Neural Networks for Accurate RSSI Indoor Localization', *IEEE Internet Things J*, vol. 6, no. 6, pp. 10639–10651, Dec. 2019, doi: 10.1109/JIOT.2019.2940368.
- [171] A. Gomez Chavez, C. A. Mueller, A. Birk, A. Babic, and N. Miskovic, 'Stereo-vision based diver pose estimation using LSTM recurrent neural networks for AUV navigation guidance', in *OCEANS 2017*, 2017.

- [172] T. Yan, 'Positioning of logistics and warehousing automated guided vehicle based on improved LSTM network', *International Journal of Systems Assurance Engineering and Management*, Apr. 2021, doi: 10.1007/s13198-021-01243-3.
- [173] A. Förster, A. Graves, and J. Schmidhuber, 'RNN-based Learning of Compact Maps for Efficient Robot Localization'. [Online]. Available: www.openrobotino.de
- [174] X. Liang, X. Shen, J. Feng, L. Lin, and S. Yan, 'Semantic Object Parsing with Graph LSTM', in *Computer Vision -ECCV*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., in Lecture Notes in Computer Science, vol. 9905. Cham: Springer International Publishing, 2016. doi: 10.1007/978-3-319-46448-0.
- [175] Y. Xiang and D. Fox, 'DA-RNN: Semantic Mapping with Data Associated Recurrent Neural Networks', Mar. 2017, [Online]. Available: <http://arxiv.org/abs/1703.03098>
- [176] R. Wang, J.-M. Frahm, and S. M. Pizer, 'Recurrent Neural Network for Learning DenseDepth and Ego-Motion from Video', May 2018, [Online]. Available: <http://arxiv.org/abs/1805.06558>
- [177] M. Chancán and M. Milford, 'DeepSeqSLAM: A Trainable CNN+RNN for Joint Global Description and Sequence-based Place Recognition', Nov. 2020, [Online]. Available: <http://arxiv.org/abs/2011.08518>
- [178] R. Liu, J. Shen, C. Chen, and J. Yang, 'SLAM for Robotic Navigation by Fusing RGB-D and Inertial Data in Recurrent and Convolutional Neural Networks', in *2019 IEEE 5th International Conference on Mechatronics System and Robots (ICMSR 2019)*, 2019.
- [179] C. Gao, W. Wang, and M. Chen, 'Semantic segmentation with Recurrent Neural Networks on RGB-D videos', in *Chinese Automation Congress (CAC2019)*, 2019.
- [180] J. Cheng, Y. Sun, and M. Q.-H. Meng, 'A Dense Semantic Mapping System based on CRF-RNN Network', in *18th International Conference on Advanced Robotics (ICAR)*, 2017.
- [181] C. Tu, E. Takeuchi, A. Carballo, and K. Takeda, 'Point Cloud Compression for 3D LiDAR Sensor using RecurrentNeural Network with Residual Blocks', in *International Conference on Robotics and Automation (ICRA)*, 2019.
- [182] R. Azzam, Y. Alkendi, T. Taha, S. Huang, and Y. Zweiri, 'A Stacked LSTM-Based Approach for Reducing Semantic Pose Estimation Error', *IEEE Trans Instrum Meas*, vol. 70, 2021, doi: 10.1109/TIM.2020.3031156.
- [183] H. Brahmi, B. Ammar, and A. M. Alimi, '2013 International Conference on Advanced Logistics and Transport (ICALT).', in *2013 International Conference on Advanced Logistics and Transport (ICALT).*, 2013.
- [184] K. Min, D. Kim, J. Park, and K. Huh, 'RNN-Based Path Prediction of Obstacle Vehicles with Deep Ensemble', *IEEE Trans Veh Technol*, vol. 68, no. 10, pp. 10252–10256, Oct. 2019, doi: 10.1109/TVT.2019.2933232.
- [185] R. S. Nair and S. P., 'Robotic Path Planning Using Recurrent Neural Networks'.

- [186] J. Kong, J. Huang, H. Yu, H. Deng, J. Gong, and H. Chen, 'RNN-based default logic for route planning in urban environments', *Neurocomputing*, vol. 338, pp. 307–320, Apr. 2019, doi: 10.1016/j.neucom.2019.02.012.
- [187] M. Inoue, T. Yamashita, and T. Nishida, 'Robot path planning by LSTM network under changing environment', in *Advances in Intelligent Systems and Computing*, Springer Verlag, 2019, pp. 317–329. doi: 10.1007/978-981-13-0341-8_29.
- [188] F. Nicola, Y. Fujimoto, and R. Oboe, 'A LSTM Neural Network applied to Mobile Robots Path Planning', 2018.
- [189] J. Yuan, H. Wang, C. Lin, D. Liu, and D. Yu, 'A Novel GRU-RNN Network Model for Dynamic Path Planning of Mobile Robot', *IEEE Access*, vol. 7, pp. 15140–15151, 2019, doi: 10.1109/ACCESS.2019.2894626.
- [190] R. Xie, Z. Meng, L. Wang, H. Li, K. Wang, and Z. Wu, 'Unmanned Aerial Vehicle Path Planning Algorithm Based on Deep Reinforcement Learning in Large-Scale and Dynamic Environments', *IEEE Access*, vol. 9, pp. 24884–24900, 2021, doi: 10.1109/ACCESS.2021.3057485.
- [191] N. Guo, C. Li, D. Wang, Y. Song, G. Liu, and T. Gao, 'Local Path Planning of Mobile Robot Based on Long Short-Term Memory Neural Network', *Automatic Control and Computer Sciences*, vol. 55, no. 1, pp. 53–65, Jan. 2021, doi: 10.3103/S014641162101003X.
- [192] N. Guo, C. Li, T. Gao, G. Liu, Y. Li, and D. Wang, 'A Fusion Method of Local Path Planning for Mobile Robots Based on LSTM Neural Network and Reinforcement Learning', *Math Probl Eng*, vol. 2021, 2021, doi: 10.1155/2021/5524232.
- [193] S.-H. Chan, X. Xu, P.-T. Wu, M.-L. Chiang, and L.-C. Fu, 'Real-time Obstacle Avoidance using Supervised Recurrent Neural Network with Automatic Data Collection and Labeling', in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2019.
- [194] N. Bin, C. Xiong, Z. Liming, and X. Wendong, 'Recurrent Neural Network for Robot Path Planning', 2004.
- [195] H. Wang *et al.*, 'Risk Assessment and Mitigation in Local Path Planning for Autonomous Vehicles With LSTM Based Predictive Model', *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 4, pp. 2738–2749, Oct. 2022, doi: 10.1109/TASE.2021.3075773.
- [196] V. A. Vijayakumar, J. Shanthini, and S. Karthick, 'Convolutional Recurrent Neural Network Framework for Autonomous Driving Behavioral Model', in *Lecture Notes on Data Engineering and Communications Technologies*, vol. 57, Springer Science and Business Media Deutschland GmbH, 2021, pp. 761–772. doi: 10.1007/978-981-15-9509-7_62.
- [197] F. Altché and A. De La Fortelle, 'IEEE ITSC 2017 : 20th International Conference on Intelligent Transportation Systems : Mielparque Yokohama in Yokohama, Kanagawa, Japan, October 16-19, 2017.', in *20th International Conference on Intelligent Transportation Systems*, 2017.

- [198] S. Patel, B. Griffin, K. Kusano, and J. J. Corso, 'Predicting Future Lane Changes of Other Highway Vehicles using RNN-based Deep Models', Jan. 2018, [Online]. Available: <http://arxiv.org/abs/1801.04340>
- [199] P. F. Lv, B. He, and J. Guo, 'Position Correction Model Based on Gated Hybrid RNN for AUV Navigation', *IEEE Trans Veh Technol*, vol. 70, no. 6, pp. 5648–5657, Jun. 2021, doi: 10.1109/TVT.2021.3080134.
- [200] B. Xing, E. Xu, J. Wei, and Y. Meng, 'Recurrent neural network non-singular terminal sliding mode control for path following of autonomous ground vehicles with parametric uncertainties', *IET Intelligent Transport Systems*, vol. 16, no. 5, pp. 616–629, May 2022, doi: 10.1049/itr2.12161.
- [201] Z. Bai, B. Cai, W. ShangGuan, and L. Chai, 'Deep Learning Based Motion Planning ForAutonomous Vehicle Using Spatiotemporal LSTM Network', in *2018 Chinese Automation Congress (CAC)*, IEEE, 2018.
- [202] Y. S. Quan and C. C. Chung, 'Approximate Model Predictive Control with Recurrent Neural Network forAutonomous Driving Vehicles', in *2019 58th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, 2019.
- [203] M. Zhang, M. Zhang, Y. Chen, and M. Li, 'IMU Data Processing For Inertial Aided Navigation: A Recurrent Neural Network Based Approach', in *Proceedings - IEEE International Conference on Robotics and Automation*, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 3992–3998. doi: 10.1109/ICRA48506.2021.9561172.
- [204] H. fa Dai, H. wei Bian, R. ying Wang, and H. Ma, 'An INS/GNSS integrated navigation in GNSS denied environment using recurrent neural network', *Defence Technology*, vol. 16, no. 2, pp. 334–340, Apr. 2020, doi: 10.1016/j.dt.2019.08.011.
- [205] M. X. Jiang, C. Deng, Z. G. Pan, L. F. Wang, and X. Sun, 'Multiobject tracking in videos based on LSTM and deep reinforcement learning', *Complexity*, vol. 2018, 2018, doi: 10.1155/2018/4695890.
- [206] Z. Yan, B. Ouyang, D. Li, H. Liu, and Y. Wang, 'Network Intelligence Empowered Industrial Robot Control in the F-RAN Environment', *IEEE Wirel Commun*, vol. 27, no. 2, pp. 58–64, Apr. 2020, doi: 10.1109/MWC.001.1900346.
- [207] Y. Jeong, S. Kim, and K. Yi, 'Surround vehicle motion prediction using lstm-rnn for motion planning of autonomous vehicles at multi-lane turn intersections', *IEEE Open Journal of Intelligent Transportation Systems*, vol. 1, no. 1, pp. 2–14, 2020, doi: 10.1109/OJITS.2020.2965969.
- [208] L. Li, W. Zhao, C. Xu, C. Wang, Q. Chen, and S. Dai, 'Lane-Change Intention Inference Based on RNN for Autonomous Driving on Highways', *IEEE Trans Veh Technol*, vol. 70, no. 6, p. 5499, 2021, doi: 10.1109/TVT.2021.3079263^y.
- [209] Y. Jeong, 'Interactive Lane Keeping System for Autonomous Vehicles Using LSTM-RNN Considering Driving Environments', *Sensors*, vol. 22, no. 24, Dec. 2022, doi: 10.3390/s22249889.

- [210] S. Liu, P. Chang, W. Liang, N. Chakraborty, and K. Driggs-Campbell, 'Decentralized Structural-RNN for Robot Crowd Navigation with Deep Reinforcement Learning', in *Proceedings - IEEE International Conference on Robotics and Automation*, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 3517–3524. doi: 10.1109/ICRA48506.2021.9561595.
- [211] C. Song *et al.*, 'Human trajectory prediction for automatic guided vehicle with recurrent neural network', *The Journal of Engineering*, vol. 2018, no. 16, pp. 1574–1578, Nov. 2018, doi: 10.1049/joe.2018.8264.
- [212] H. Chen and X. Zhang, 'Path Planning for Intelligent Vehicle Collision Avoidance of Dynamic Pedestrian Using Att-LSTM, MSFM, and MPC at Unsignalized Crosswalk', *IEEE Transactions on Industrial Electronics*, vol. 69, no. 4, pp. 4285–4295, Apr. 2022, doi: 10.1109/TIE.2021.3073301.
- [213] C. Lin, H. Wang, J. Yuan, D. Yu, and C. Li, 'An improved recurrent neural network for unmanned underwater vehicle online obstacle avoidance', *Ocean Engineering*, vol. 189, Oct. 2019, doi: 10.1016/j.oceaneng.2019.106327.
- [214] C. Lin, H. Wang, J. Yuan, D. Yu, and C. Li, 'Research on UUV Obstacle Avoiding Method Based on Recurrent Neural Networks', *Complexity*, vol. 2019, 2019, doi: 10.1155/2019/6320186.
- [215] 'What is a Genetic Algorithm? - Definition from Techopedia'. Accessed: Jun. 09, 2023. [Online]. Available: <https://www.techopedia.com/definition/17137/genetic-algorithm>
- [216] 'Unit 3) Genetic Algorithms (Part 1) | by Brandon Morgan | Towards Data Science'. Accessed: Jun. 09, 2023. [Online]. Available: <https://towardsdatascience.com/unit-3-genetic-algorithms-part-1-986e3b4666d7>
- [217] L. Moreno, J. M. Armingol, S. Garrido, A. De, L. A. Escalera, and M. A. Salichs, 'A Genetic Algorithm for Mobile Robot Localization Using Ultrasonic Sensors', 2002.
- [218] X. Gao, J. Wang, and W. Chen, 'Land-mark Placement for Reliable Localization of Automatic Guided Vehicle in Warehouse Environment', in *2015 IEEE International Conference on Robotics and Biomimetics : IEEE-ROBIO 2015*, 2015.
- [219] G.-F. Nan, M.-Q. Li, and J. Li, 'Estimation Of Node Localization With A Real-Coded Genetic Algorithm In WSNS', in *Sixth International Conference on Machine Learning and Cybernetics*, IEEE Xplore, 2007.
- [220] T. Najeh, H. Sassi, and N. Liouane, 'A Novel Range Free Localization Algorithm in Wireless Sensor Networks Based on Connectivity and Genetic Algorithms', *Int J Wirel Inf Netw*, vol. 25, no. 1, pp. 88–97, Mar. 2018, doi: 10.1007/s10776-017-0375-y.
- [221] B. Peng and L. Li, 'An improved localization algorithm based on genetic algorithm in wireless sensor networks', *Cogn Neurodyn*, vol. 9, no. 2, pp. 249–256, Mar. 2015, doi: 10.1007/s11571-014-9324-y.
- [222] G. Sharma and A. Kumar, 'Improved range-free localization for three-dimensional wireless sensor networks using genetic algorithm', *Computers and Electrical Engineering*, vol. 72, pp. 808–827, Nov. 2018, doi: 10.1016/j.compeleceng.2017.12.036.

- [223] M. Romoozi and H. Ebrahimpour-komleh, 'A Positioning Method in Wireless Sensor Networks using Genetic Algorithms', *Phys Procedia*, vol. 33, pp. 1042–1049, 2012, doi: 10.1016/j.phpro.2012.05.171.
- [224] Q. Zhang, J. Wang, C. Jin, J. Ye, M. Changlin, and W. Zhang, 'Genetic algorithm based wireless sensor network localization', in *Proceedings - 4th International Conference on Natural Computation, ICNC 2008*, 2008, pp. 608–613. doi: 10.1109/ICNC.2008.206.
- [225] J. Díez-González, R. Álvarez, D. González-Bárcena, L. Sánchez-González, M. Castejón-Limas, and H. Perez, 'Genetic algorithm approach to the 3D node localization in TDOA systems', *Sensors (Switzerland)*, vol. 19, no. 18, Sep. 2019, doi: 10.3390/s19183880.
- [226] J. Zhao, J. Jiang, and X. Zang, 'Cooperative Multi-Robot Map-building based on Genetic Algorithms', in *2006 IEEE International Conference on Information Acquisition*, 2006.
- [227] Q. Sun, Y. Zhao, H. Du, M. Gao, and X. Jin, 'A Multi-Drone Joint Map Building Method Based On Multi-variable Layer Directed Cross Genetic Algorithm', in *Proceedings of the 41st Chinese Control Conference*, 2022.
- [228] K. C. Mertens, L. P. C. Verbeke, E. I. Ducheyne, and R. R. De Wulf, 'Using genetic algorithms in sub-pixel mapping', *Int J Remote Sens*, vol. 24, no. 21, pp. 4241–4247, Nov. 2003, doi: 10.1080/01431160310001595073.
- [229] B. L. Boada, D. Blanco, C. Castejón, C. Castejón, and L. Moreno, 'A genetic solution for the slam problem', in *The 11th International Conference on Advanced Robotics*, 2003. [Online]. Available: <https://www.researchgate.net/publication/244956950>
- [230] T. Duckett, 'A Genetic Algorithm for Simultaneous Localization and Mapping', 2003. [Online]. Available: <http://www.aass.oru.se>
- [231] G. Mingas, E. Tsardoulas, and L. Petrou, 'An FPGA implementation of the SMG-SLAM algorithm', *Microprocess Microsyst*, vol. 36, no. 3, pp. 190–204, May 2012, doi: 10.1016/j.micpro.2011.12.002.
- [232] O. El Hamzaoui and B. Steux, 'SLAM Algorithm with Parallel Localization Loops : tinySLAM 1.1', in *IEEE International Conference on Automation and Logistics*, 2011.
- [233] M. Begum, G. K. I. Mann, and R. G. Gosine, 'An Evolutionary SLAM Algorithm for MobileRobots', in *The proceedings of 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems : IROS 2006*, IEEE, 2006, p. 5774.
- [234] M. Liu, S. Huang, and G. Dissanayake, 'Feature Based SLAM Using Laser Sensor Data with Maximized Information Usage', in *2011 IEEE International Conference on Robotics and Automation.*, IEEE, 2011.
- [235] C. H. Wu, Y. H. Chen, Y. Y. Lee, and C. H. Tsai, 'A fast genetic SLAM approach for mobile robots', in *SNPD 2013 - 14th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, 2013, pp. 563–568. doi: 10.1109/SNPD.2013.12.
- [236] X. Mu, G. Yue, N. Zhou, and C. Chen, 'Occupancy Grid-Based AUV SLAM Method with Forward-Looking Sonar', *J Mar Sci Eng*, vol. 10, no. 8, Aug. 2022, doi: 10.3390/jmse10081056.

- [237] S. Ortiz and W. Yu, 'Autonomous navigation in unknown environment using sliding mode SLAM and genetic algorithm', *Intelligence & Robotics*, 2021, doi: 10.20517/ir.2021.09.
- [238] S. Ortiz, W. Yu, and X. Li, 'Autonomous Navigation Using Robust SLAM and Genetic Algorithm', in *IEEE International Conference on Automation Science and Engineering*, IEEE Computer Society, Aug. 2021, pp. 1346–1351. doi: 10.1109/CASE49439.2021.9551539.
- [239] B. Chakraborty, T. Maeda, and G. Chakraborty, 'Multiobjective Route Selection for Car Navigation System using Genetic Algorithm', in *IEEE Mid-Summer Workshop on Soft Computing in Industrial Applications*, 2005.
- [240] C. Lamini, S. Benhlima, and A. Elbekri, 'Genetic algorithm based approach for autonomous mobile robot path planning', in *Procedia Computer Science*, Elsevier B.V., 2018, pp. 180–189. doi: 10.1016/j.procs.2018.01.113.
- [241] Z. Han, D. Wang, F. Liu, and Z. Zhao, 'Multi-AGV path planning with double-path constraints by using an improved genetic algorithm', *PLoS One*, vol. 12, no. 7, Jul. 2017, doi: 10.1371/journal.pone.0181747.
- [242] A. Bakdi, A. Hentout, H. Boutami, A. Maoudj, O. Hachour, and B. Bouzouia, 'Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control', *Rob Auton Syst*, vol. 89, pp. 95–109, Mar. 2017, doi: 10.1016/j.robot.2016.12.008.
- [243] Y. Hu and S. X. Yang, 'A Knowledge Based Genetic Algorithm for Path Planning of a Mobile Robot', in *2004 IEEE international Conference on Robotics & Automatio*, 2004.
- [244] A. Elshamli, H. A. Abdullah, and S. Areibi, 'Genetic Algorithm for Dynamic Path Planning', 2004.
- [245] R. M. C. Santiago, A. Louise De Ocampo, A. T. Ubando, A. A. Bandala, and P. Dadios. Elmer, 'Path Planning for Mobile Robots Using Genetic Algorithm and Probabilistic Roadmap', in *2017 IEEE National Aerospace and Electronics Conference (NAECON)*, 2017.
- [246] J. Liu, Z. Chen, Y. Zhang, and W. Li, 'Path Planning of Mobile Robots based on Improved Genetic Algorithm', in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Oct. 2020, pp. 49–53. doi: 10.1145/3438872.3439054.
- [247] K. K. Kwaśniewski and Z. Gosiewski, 'Genetic algorithm for mobile robot route planning with obstacle avoidance', *Acta Mechanica et Automatica*, vol. 12, no. 2, pp. 151–159, Jun. 2018, doi: 10.2478/ama-2018-0024.
- [248] K. H. Sedighi, K. Ashenayi, and T. W. Manikas, 'Autonomous Local Path Planning for a Mobile Robot Using a Genetic Algorithm', in *CEC2004 : proceedings of the 2004 Congress on Evolutionary Computation*, IEEE, 2004.
- [249] T. Geisler and T. W. Manikas, 'Autonomous Robot Navigation System Using A Novel Value Encoded Genetic Algorithm', 2002.

- [250] A. Ghorbani, S. Shiry, and A. Nodehi, 'Using genetic algorithm for a mobile robot path planning', in *Proceedings - 2009 International Conference on Future Computer and Communication, ICFCC 2009*, 2009, pp. 164–166. doi: 10.1109/ICFCC.2009.28.
- [251] I. Al-Taharwa, A. Sheta, and M. Al-Weshah, 'A Mobile Robot Path Planning Using Genetic Algorithm in Static Environment', *Journal of Computer Science*, vol. 4, no. 4, pp. 341–344, 2008.
- [252] J. Tu and S. X. Yangt, 'Genetic Algorithm Based Path Planning for a Mobile Robot*', in *2003 IEEE Internatioarl Conference on Robotics Y AulOrnatioO*, 2003.
- [253] A. Tuncer and M. Yildirim, 'Dynamic path planning of mobile robots with improved genetic algorithm', *Computers and Electrical Engineering*, vol. 38, no. 6, pp. 1564–1572, Nov. 2012, doi: 10.1016/j.compeleceng.2012.06.016.
- [254] G. Nagib and W. Gharieb, 'Path Planning For A Mobile Robot Using Genetic Algorithms', 2004.
- [255] P. Shi and Y. Cui, 'Dynamic Path Planning for Mobile Robot Based on Genetic Algorithm in Unknown Environment', in *2010 Chinese Control and Decision Conference*, IEEE Industrial Electronics (IE) Chapter, 2010, p. 4503.
- [256] X. Du, H. H. Chen, and W. K. Gu, 'Neural network and genetic algorithm based global path planning in a static environment', *J Zhejiang Univ Sci*, vol. 6 A, no. 6, pp. 549–554, Jun. 2005, doi: 10.1631/jzus.2005.A0549.
- [257] S. Arrigoni, F. Braghin, and F. Cheli, 'MPC trajectory planner for autonomous driving solved by genetic algorithm technique', *Vehicle System Dynamics*, vol. 60, no. 12, pp. 4118–4143, 2022, doi: 10.1080/00423114.2021.1999991.
- [258] C. C. Tsai, H. C. Huang, and C. K. Chan, 'Parallel elite genetic algorithm and its application to global path planning for autonomous robot navigation', *IEEE Transactions on Industrial Electronics*, vol. 58, no. 10, pp. 4813–4821, Oct. 2011, doi: 10.1109/TIE.2011.2109332.
- [259] C. Wang, Y. C. Soh, H. Wang, and H. Wang, 'A Hierarchical Genetic Algorithm for Path Planning in a Static Environment with Obstacles', in *2002 IEEE Canadian Conference on Electrical & Computer Engineering*, 2002.
- [260] R. K. Panda and B. B. Choudhury, 'An effective path planning of mobile robot using genetic algorithm', in *Proceedings - 2015 IEEE International Conference on Computational Intelligence and Communication Technology, CICT 2015*, Institute of Electrical and Electronics Engineers Inc., Apr. 2015, pp. 287–291. doi: 10.1109/CICT.2015.145.
- [261] A. Kumar, P. B. Kumar, and D. R. Parhi, 'Intelligent Navigation of Humanoids in Cluttered Environments Using Regression Analysis and Genetic Algorithm', *Arab J Sci Eng*, vol. 43, no. 12, pp. 7655–7678, Dec. 2018, doi: 10.1007/s13369-018-3157-7.
- [262] M. Yoshikawa and H. Terai, 'Car navigation system based on hybrid genetic algorithm', in *2009 WRI World Congress on Computer Science and Information Engineering, CSIE 2009*, 2009, pp. 62–65. doi: 10.1109/CSIE.2009.558.

- [263] M. Mousavi, H. J. Yap, S. N. Musa, F. Tahriri, and S. Z. Md Dawal, 'Multi-objective AGV scheduling in an FMS using a hybrid of genetic algorithm and particle swarm optimization', *PLoS One*, vol. 12, no. 3, Mar. 2017, doi: 10.1371/journal.pone.0169817.
- [264] S. M. Jafar Jalali, P. M. Kebria, A. Khosravi, K. Saleh, D. Nahavandi, and S. Nahavandi, 'Optimal Autonomous Driving Through Deep Imitation Learning and Neuroevolution', in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2019.
- [265] Z. Gyenes, L. Bölöni, and E. G. Szádeczky-Kardoss, 'Can Genetic Algorithms Be Used for Real-Time Obstacle Avoidance for LiDAR-Equipped Mobile Robots?', *Sensors*, vol. 23, no. 6, p. 3039, Mar. 2023, doi: 10.3390/s23063039.
- [266] F. Yan, M. Dridi, and A. El Moudni, 'An autonomous vehicle sequencing problem at intersections: A genetic algorithm approach', *International Journal of Applied Mathematics and Computer Science*, vol. 23, no. 1, pp. 183–200, Mar. 2013, doi: 10.2478/amcs-2013-0015.
- [267] L. Cruz-Piris, M. A. Lopez-Carmona, and I. Marsa-Maestre, 'Automated Optimization of Intersections Using a Genetic Algorithm', *IEEE Access*, vol. 7, pp. 15452–15468, 2019, doi: 10.1109/ACCESS.2019.2895370.
- [268] 'Introducción a la Lógica Difusa - Fernando Sancho Caparrini'. Accessed: Jun. 03, 2023. [Online]. Available: <http://www.cs.us.es/~fsancho/?e=97>
- [269] M. Nour, S. M. Said, H. Ramadan, A. Ali, and C. Farkas, 'Control of Electric Vehicles Charging Without Communication Infrastructure', in *2018 20th International Middle East Power Systems Conference, MEPCON 2018 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., Feb. 2019, pp. 773–778. doi: 10.1109/MEPCON.2018.8635277.
- [270] A. Saffiotti *et al.*, 'The uses of fuzzy logic in autonomous robot navigation', *Soft Computing 1997 1:4*, vol. 1, no. 4, pp. 180–197, Dec. 1997, doi: 10.1007/S005000050020.
- [271] J. Gasós and A. Martin, 'Mobile Robot Localization Using Fuzzy Maps', 2005.
- [272] M. Alakhras, M. Oussalah, and M. Hussein, 'A survey of fuzzy logic in wireless localization', *Eurasip Journal on Wireless Communications and Networking*, vol. 2020, no. 1. Springer, Dec. 01, 2020. doi: 10.1186/s13638-020-01703-7.
- [273] K. Demirli and I. B. Türk, türk, sen, 'Sonar based mobile robot localization by using fuzzy triangulation', 2000.
- [274] L. Caroprese, C. Comito, D. Talia, and E. Zumpano, 'A fuzzy logic technique for virtual sensor networks', *Future Generation Computer Systems*, vol. 137, pp. 302–322, Dec. 2022, doi: 10.1016/j.future.2022.07.022.
- [275] H. Y. Chung, C. C. Hou, and Y. S. Chen, 'Indoor Intelligent Mobile Robot Localization Using Fuzzy Compensation and Kalman Filter to Fuse the Data of Gyroscope and Magnetometer', *IEEE Transactions on Industrial Electronics*, vol. 62, no. 10, pp. 6436–6447, Oct. 2015, doi: 10.1109/TIE.2015.2416692.

- [276] H. Lee, J. Jung, K. Choi, J. Park, and H. Myung, 'Fuzzy-logic-assisted interacting multiple model (FLAIMM) for mobile robot localization', *Rob Auton Syst*, vol. 60, no. 12, pp. 1592–1606, Dec. 2012, doi: 10.1016/j.robot.2012.09.018.
- [277] K. Jajulwar and A. Y. Deshmukh, 'Design of SLAM based Adaptive Fuzzy Tracking Controller for Autonomous Navigation System', 2016.
- [278] C. Fu, M. A. Olivares-Mendez, R. Suarez-Fernandez, and P. Campoy, 'Monocular Visual-Inertial SLAM-based collision avoidance strategy for Fail-Safe UAV using Fuzzy Logic Controllers: Comparison of two Cross-Entropy Optimization approaches', *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 73, no. 1–4, pp. 513–533, 2014, doi: 10.1007/s10846-013-9918-3.
- [279] C. D. Pathiranage, K. Watanabe, and K. Izumi, 'A fuzzy logic based approach to the SLAM problem using pseudolinear models with multiframe data association', *Artif Life Robot*, vol. 13, no. 1, pp. 155–161, 2008, doi: 10.1007/s10015-008-0566-9.
- [280] K. Watanabe, C. Dedduwa Pathiranage, and K. Izumi, 'A fuzzy Kalman filter approach to the SLAM problem of nonholonomic mobile robots', in *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 2008. doi: 10.3182/20080706-5-KR-1001.0942.
- [281] S. Khanmohammadi and M. K. Mirnia, 'Multi AGV hybrid path planning using Fuzzy Inference Systems', 2010, doi: 10.1109/ICCAE.2010.5451236.
- [282] M. Fakoor, A. Kosari, and M. Jafarzadeh, 'Humanoid robot path planning with fuzzy Markov decision processes', *Journal of Applied Research and Technology*, vol. 14, no. 5, pp. 300–310, Oct. 2016, doi: 10.1016/J.JART.2016.06.006.
- [283] A. R. Soltani and T. Fernando, 'A fuzzy based multi-objective path planning of construction sites', *Autom Constr*, vol. 13, no. 6, pp. 717–734, Nov. 2004, doi: 10.1016/j.autcon.2004.04.012.
- [284] M. Wang and J. N. K. Liu, 'Fuzzy logic based robot path planning in unknown environment', in *2005 International Conference on Machine Learning and Cybernetics, ICMLC 2005*, 2005, pp. 813–818. doi: 10.1109/icmlc.2005.1527055.
- [285] X. huan Liu, D. Zhang, J. Zhang, T. Zhang, and H. Zhu, 'A path planning method based on the particle swarm optimization trained fuzzy neural network algorithm', *Cluster Comput*, vol. 24, no. 3, pp. 1901–1915, Sep. 2021, doi: 10.1007/S10586-021-03235-1.
- [286] B. Sun, D. Zhu, and S. X. Yang, 'An Optimized Fuzzy Control Algorithm for Three-Dimensional AUV Path Planning', *International Journal of Fuzzy Systems*, vol. 20, no. 2, pp. 597–610, Feb. 2018, doi: 10.1007/S40815-017-0403-1.
- [287] Q. Liu, 'Optimal Genetic Fuzzy Obstacle Avoidance Controller of Autonomous Mobile Robot Based on Ultrasonic Sensors', 2006, doi: 10.1109/ROBIO.2006.340327.
- [288] R. Kala *et al.*, 'Fusion of probabilistic A* algorithm and fuzzy inference system for robotic path planning', *Springer*, vol. 33, no. 4, pp. 307–327, 2010, doi: 10.1007/s10462-010-9157-y.
- [289] M. Sc Sarah Uttendorf and L. Overmeyer, 'Fuzzy-enhanced path-finding algorithm for AGV roadmaps', *Proceedings of the 2015 Conference of the International Fuzzy*

- Systems Association and the European Society for Fuzzy Logic and Technology*, vol. 89, pp. 675–681, Jun. 2015, doi: 10.2991/IFSA-EUSFLAT-15.2015.96.
- [290] X. Li, B. C.-I. J. of S. Home, and undefined 2013, 'Design of obstacle avoidance system for mobile robot using fuzzy logic systems', *gvpress.com*, vol. 7, no. 3, 2013, Accessed: Apr. 08, 2023. [Online]. Available: https://gvpress.com/journals/IJSH/vol7_no3/30.pdf
- [291] B. K. Patle, D. R. K. Parhi, A. Jagadeesh, and S. K. Kashyap, 'Application of probability to enhance the performance of fuzzy based mobile robot navigation', *Appl Soft Comput*, vol. 75, pp. 265–283, Feb. 2019, doi: 10.1016/J.ASOC.2018.11.026.
- [292] S. Ishikawa, 'A method of indoor mobile robot navigation by using fuzzy control', 1991, doi: 10.1109/IROS.1991.174624.
- [293] H. Omrane, M. S. Masmoudi, and M. Masmoudi, 'Fuzzy Logic Based Control for Autonomous Mobile Robot Navigation', 2016, doi: 10.1155/2016/9548482.
- [294] Y. Rubio *et al.*, 'Path following fuzzy system for a nonholonomic mobile robot based on frontal camera information', *Studies in Computational Intelligence*, vol. 749, pp. 223–240, 2018, doi: 10.1007/978-3-319-71008-2_18.
- [295] S. Kovács and L. T. Kóczy, 'Application of an approximate fuzzy logic controller in an AGV steering system, path tracking and collision avoidance strategy, Fuzzy Set Theory and Applications', Tatra Mountains Mathematical Publications, 1999.
- [296] S. K. Pradhan, D. R. Parhi, and A. K. Panda, 'Fuzzy logic techniques for navigation of several mobile robots', *Applied Soft Computing Journal*, vol. 9, no. 1, pp. 290–304, Jan. 2009, doi: 10.1016/j.asoc.2008.04.008.
- [297] D. R. Parhi, 'Navigation of Mobile Robots Using a Fuzzy Logic Controller', *J Intell Robot Syst*, vol. 42, pp. 253–273, 2005.
- [298] S. yun Kim and Y. Yang, 'A self-navigating robot using Fuzzy Petri nets', *Rob Auton Syst*, vol. 101, pp. 153–165, Mar. 2018, doi: 10.1016/j.robot.2017.11.008.
- [299] M. Faisal, R. Hedjar, ... M. A. S.-... J. of A., and undefined 2013, 'Fuzzy logic navigation and obstacle avoidance by a mobile robot in an unknown dynamic environment', *journals.sagepub.com*, vol. 10, Jan. 2013, doi: 10.5772/54427.
- [300] M. A. K. Jaradat, M. H. Garibeh, and E. A. Feilat, 'Autonomous mobile robot dynamic motion planning using hybrid fuzzy potential field', *Soft comput*, vol. 16, no. 1, pp. 153–164, Jan. 2012, doi: 10.1007/S00500-011-0742-Z.
- [301] Y. Dianyong and X. Hui, 'Application of Fuzzy Control Method to AGV', 2003, doi: 10.1109/RISSP.2003.1285682.
- [302] N. Kelkar, T. Samu, and E. L. Hall, 'Fuzzy logic control of an AGV', in *Intelligent Robots and Computer Vision XVI: Algorithms, Techniques, Active Vision, and Materials Handling*, SPIE, Sep. 1997, pp. 140–149. doi: 10.1117/12.290287.
- [303] A. Pandey and D. R. Parhi, 'Optimum path planning of mobile robot in unknown static and dynamic environments using Fuzzy-Wind Driven Optimization algorithm', *Defence Technology*, vol. 13, no. 1, pp. 47–58, Feb. 2017, doi: 10.1016/j.dt.2017.01.001.

- [304] X. Li, C. Luo, Y. Xu, and P. Li, 'A Fuzzy PID controller applied in AGV control system; A Fuzzy PID controller applied in AGV control system', 2016, doi: 10.1109/ICARM.2016.7606981.
- [305] K. R. S. Kodagoda, W. S. Wijesoma, and E. K. Teoh, 'Fuzzy speed and steering control of an AGV; Fuzzy speed and steering control of an AGV', *IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY*, vol. 10, no. 1, 2002, doi: 10.1109/87.974344.
- [306] W. Shi, K. Wang, and S. X. Yang, 'A fuzzy-neural network approach to multisensor integration for obstacle avoidance of a mobile robot', *Intelligent Automation and Soft Computing*, vol. 15, no. 2, pp. 289–301, 2009, doi: 10.1080/10798587.2009.10643032.
- [307] H. Maaref and C. Barret, 'Sensor-based fuzzy navigation of an autonomous mobile robot in an indoor environment', 2000.
- [308] S. Lee, T. M. Adams, and B. Y. Ryoo, 'A fuzzy navigation system for mobile construction robots', *Autom Constr*, vol. 6, no. 2, pp. 97–107, 1997, doi: 10.1016/S0926-5805(96)00185-9.
- [309] J. Lilly, 'Evolution of a negative-rule fuzzy obstacle avoidance controller for an autonomous vehicle', *IEEE Transactions on Fuzzy Systems*, vol. 15, 2007, Accessed: Apr. 08, 2023. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/4286972/>
- [310] S. H. A. Mohammad, M. Akmal Jeffril, and N. Sariff, 'Mobile Robot Obstacle Avoidance By Using Fuzzy Logic Technique', in *IEEE 3rd International Conference on System Engineering and Technology*, 2013, p. 417.
- [311] A. M. Rao, K. Ramji, B. S. K. Sundara Siva Rao, V. Vasu, and C. Puneeth, 'Navigation of non-holonomic mobile robot using neuro-fuzzy logic with integrated safe boundary algorithm', *International Journal of Automation and Computing*, vol. 14, no. 3, pp. 285–294, Jun. 2017, doi: 10.1007/s11633-016-1042-y.
- [312] J. Janai, F. Güney, A. Behl, and A. Geiger, 'Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art', Apr. 2017, [Online]. Available: <http://arxiv.org/abs/1704.05519>
- [313] '15 Best Open-Source Autonomous Driving Datasets | by Alex Nguyen | Analytics Vidhya | Medium'. Accessed: Jun. 19, 2023. [Online]. Available: <https://medium.com/analytics-vidhya/15-best-open-source-autonomous-driving-datasets-34324676c8d7>
- [314] 'Download | a2d2.audi'. Accessed: Jun. 20, 2023. [Online]. Available: <https://www.a2d2.audi/a2d2/en/download.html>
- [315] 'Apollo Scape'. Accessed: Jun. 20, 2023. [Online]. Available: <https://apolloscape.auto/scene.html>
- [316] 'Argoverse 1'. Accessed: Jun. 20, 2023. [Online]. Available: <https://www.argoverse.org/av1.html#download-link>
- [317] 'Login – Cityscapes Dataset'. Accessed: Jun. 20, 2023. [Online]. Available: <https://www.cityscapes-dataset.com/login/>

- [318] 'Google Landmarks Dataset v2 – Google Research'. Accessed: Jun. 20, 2023. [Online]. Available: <https://research.google/resources/datasets/google-landmarks-v2/>
- [319] 'nuScenes'. Accessed: Jun. 20, 2023. [Online]. Available: <https://www.nuscenes.org/nuscenes#download>
- [320] 'The Oxford Radar RobotCar Dataset'. Accessed: Jun. 20, 2023. [Online]. Available: <https://oxford-robotics-institute.github.io/radar-robotcar-dataset/downloads>
- [321] 'Pandaset by Hesai and Scale'. Accessed: Jun. 20, 2023. [Online]. Available: <https://pandaset.org/#data-collection>
- [322] D. Scharstein and R. Szeliski, 'A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms', 2002. [Online]. Available: www.middlebury.edu/stereo.
- [323] C. Hane, C. Zach, J. Lim, A. Ranganathan, and M. Pollefeys, 'Stereo Depth Map Fusion for Robot Navigation', in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems.*, IEEE, 2011.
- [324] D. Scharstein and R. Szeliski, 'High-Accuracy Stereo Depth Maps Using Structured Light', 2003. [Online]. Available: <http://www.middlebury.edu/stereo/>.
- [325] D. Scharstein *et al.*, 'High-Resolution Stereo Datasets with Subpixel-Accurate Ground Truth', in *Pattern recognition: 36th German Conference, GCPR 2014 Münster, Germany, September 2–5, 2014 proceedings*, Springer Verlag, 2014. doi: 10.1007/978-3-319-11752-2.
- [326] Y. Yao, Z. Luo, S. Li, T. Shen, T. Fang, and † Long Quan, 'Recurrent MVSNet for High-resolution Multi-view Stereo Depth Inference', 2019. [Online]. Available: <https://github.com/>
- [327] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, 'MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking', Apr. 2015, [Online]. Available: <http://arxiv.org/abs/1504.01942>
- [328] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski, 'A database and evaluation methodology for optical flow', *Int J Comput Vis*, vol. 92, no. 1, pp. 1–31, Mar. 2011, doi: 10.1007/s11263-010-0390-2.
- [329] J. Janai, F. Güney, J. Wulff, M. Black, and A. Geiger, 'Slow Flow: Exploiting High-Speed Cameras for Accurate and Diverse Optical Flow Reference Data', 2017. [Online]. Available: <http://www.cvlibs.net/projects/slow>
- [330] Y. Zhang *et al.*, 'ByteTrack: Multi-object Tracking by Associating Every Detection Box', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13682 LNCS, pp. 1–21, 2022, doi: 10.1007/978-3-031-20047-2_1/COVER.
- [331] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, 'MOT16: A Benchmark for Multi-Object Tracking', Mar. 2016, [Online]. Available: <http://arxiv.org/abs/1603.00831>
- [332] P. Dendorfer *et al.*, 'MOT20: A benchmark for multi object tracking in crowded scenes', Mar. 2020, [Online]. Available: <http://arxiv.org/abs/2003.09003>

- [333] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, 'FairMOT: On the Fairness of Detection and Re-identification in Multiple Object Tracking', *Int J Comput Vis*, vol. 129, no. 11, pp. 3069–3087, Nov. 2021, doi: 10.1007/s11263-021-01513-4.
- [334] X. Zhou, V. Koltun, and P. Krähenbühl, 'Tracking Objects as Points', in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., in Lecture Notes in Computer Science, vol. 12349. Cham: Springer International Publishing, 2020. doi: 10.1007/978-3-030-58548-8.
- [335] T. Meinhardt, A. Kirillov, L. Leal-Taixé, and C. Feichtenhofer, 'TrackFormer: Multi-Object Tracking with Transformers', 2021. [Online]. Available: <https://github.com>.
- [336] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, 'The pascal visual object classes (VOC) challenge', *Int J Comput Vis*, vol. 88, no. 2, pp. 303–338, Jun. 2010, doi: 10.1007/s11263-009-0275-4.
- [337] T.-Y. Lin *et al.*, 'Microsoft COCO: Common Objects in Context', 2014.
- [338] A. Srinivas, T.-Y. Lin, N. Parmar, J. Shlens, P. Abbeel, and A. Vaswani, 'Bottleneck Transformers for Visual Recognition', in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. [Online]. Available: <https://arxiv.org/abs/2101.11605>
- [339] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, 'BiSeNet: Bilateral Segmentation Network for Real-time Semantic Segmentation', in *European Conference on Computer Vision (ECCV)*, 2018.
- [340] R. Strudel, R. Garcia, I. Laptev Inria, and C. Schmid Inria, 'Segmenter: Transformer for Semantic Segmentation', in *International Conference on Computer Vision (ICCV)*, 2021. [Online]. Available: <https://github.com/rstrudel/segmenter>
- [341] M. Kalaitzakis, B. Cain, S. Carroll, A. Ambrosi, C. Whitehead, and N. Vitzilaios, 'Fiducial Markers for Pose Estimation: Overview, Applications and Experimental Comparison of the ARTag, AprilTag, ArUco and STag Markers', *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 101, no. 4, Apr. 2021, doi: 10.1007/s10846-020-01307-9.
- [342] E. Olson, 'AprilTag: A robust and flexible visual fiducial system', in *2011 IEEE International Conference on Robotics and Automation*, IEEE, 2011.
- [343] J. Wang and E. Olson, 'AprilTag 2: Efficient and robust fiducial detection', in *IROS 2016 : 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016.
- [344] M. Fiala, 'ARTag, a fiducial marker system using digital techniques', in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005. [Online]. Available: <http://www.photomodeler.com>
- [345] M. Fiala, 'Designing highly reliable fiducial markers', *IEEE Trans Pattern Anal Mach Intell*, vol. 32, no. 7, pp. 1317–1324, 2010, doi: 10.1109/TPAMI.2009.146.
- [346] H. Kato and M. Billinghurst, 'Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System', in *Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*, 1999.

- [347] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, 'Automatic generation and detection of highly reliable fiducial markers under occlusion', *Pattern Recognit*, vol. 47, no. 6, pp. 2280–2292, Jun. 2014, doi: 10.1016/j.patcog.2014.01.005.
- [348] D. Flohr and J. Fischer, 'A Lightweight ID-Based Extension for Marker Tracking Systems', in *IPT-EGVE Symposium*, 2007.
- [349] C. N. Klokrose, J. B. Kristensen, R. Bagge, and K. Halskov, 'Bullseye: High-precision fiducial tracking for table-based tangible interaction', in *ITS 2014 - Proceedings of the 2014 ACM International Conference on Interactive Tabletops and Surfaces*, Association for Computing Machinery, Nov. 2014, pp. 269–278. doi: 10.1145/2669485.2669503.
- [350] B. Atcheson, F. Heide, and W. Heidrich, 'CALTag: High precision fiducial markers for camera calibration', in *VMV 2010 - Vision, Modeling and Visualization*, 2010, pp. 41–48. doi: 10.2312/PE/VMV/VMV10/041-048.
- [351] L. B. Gatrell, W. A. Hoff, and C. W. Sklair, 'Robust image features: concentric contrasting circles and their image extraction', <https://doi.org/10.1117/12.56761>, vol. 1612, pp. 235–244, Mar. 1992, doi: 10.1117/12.56761.
- [352] L. Calvet, P. Gurdjos, C. Griwodz, and S. Gasparini, 'Detection and Accurate Localization of Circular Fiducials under Highly Challenging Conditions', in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [353] J. Degol, T. Bretl, and D. Hoiem, 'ChromaTag: A Colored Marker and Fast Detection Algorithm', in *IEEE International Conference on Computer Vision (ICCV)*, 2017. [Online]. Available: <http://degol2.web.engr.illinois.edu/pages/ChromaTag>
- [354] J. Sattar, E. Bourque, P. Gigù, and G. Dudek, 'Fourier tags: Smoothly degradable fiducial markers for use in human-robot interaction', in *Fourth Canadian Conference on Computer and Robot Vision(CRV'07)*, 2007.
- [355] A. Xu and G. Dudek, 'Fourier tag: A smoothly degradable fiducial marker system with configurable payload capacity', in *Proceedings - 2011 Canadian Conference on Computer and Robot Vision, CRV 2011*, 2011, pp. 40–47. doi: 10.1109/CRV.2011.13.
- [356] L. Naimark and E. Foxlin, 'Circular Data Matrix Fiducial System and Robust Image Processing for a Wearable Vision-Inertial Self-Tracker', in *Proceedings. International Symposium on Mixed and Augmented Reality*, 2002.
- [357] Y. Cho, J. Lee, and U. Neumann, 'A Multi-ring Color Fiducial System and An Intensity-invariant Detection Method for Scalable Fiducial-Tracking Augmented Reality', in *24th BCS Interaction Specialist Group Conference.*, 1970, pp. 269–277. [Online]. Available: <http://tracker.usc.edu/~ykcho/star.html>
- [358] F. Bergamasco, A. Albarelli, and A. Torsello, 'Pi-Tag: A fast image-space marker design based on projective invariants', *Mach Vis Appl*, vol. 24, no. 6, pp. 1295–1310, Aug. 2013, doi: 10.1007/s00138-012-0469-6.
- [359] F. Bergamasco, A. Albarelli, E. Rodò, and A. Torsello, 'RUNE-Tag: a High Accuracy Fiducial Marker with Strong Occlusion Resilience', in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. [Online]. Available: <http://www.dais.unive.it>

- [360] F. Bergamasco, A. Albarelli, L. Cosmo, E. Rodola, and A. Torsello, 'An Accurate and Robust Artificial Marker Based on Cyclic Codes', *IEEE Trans Pattern Anal Mach Intell*, vol. 38, no. 12, pp. 2359–2373, Dec. 2016, doi: 10.1109/TPAMI.2016.2519024.
- [361] F. Schweiger, B. Zeisl, P. Georgel, G. Schroth, E. Steinbach, and N. Navab, 'Maximum Detector Response Markers for SIFT and SURF', in *Proceedings of the Vision, Modeling, and Visualization Workshop 2009*, 2009.
- [362] B. Benligiray, C. Topal, and C. Akinlar, 'STag: A stable fiducial marker system', *Image Vis Comput*, vol. 89, pp. 158–169, Sep. 2019, doi: 10.1016/j.imavis.2019.06.007.
- [363] G. Yu, Y. Hu, and J. Dai, 'TopoTag: A Robust and Scalable Topological Fiducial Marker System', *IEEE Trans Vis Comput Graph*, vol. 27, no. 9, pp. 3769–3780, Sep. 2021, doi: 10.1109/TVCG.2020.2988466.
- [364] D. López De Ipiña, P. R. S. Mendonça, and A. Hopper, 'TRIP: A Low-Cost Vision-Based Location System for Ubiquitous Computing', *Personal and Ubiquitous Computing volume*, vol. 6, pp. 206–219, 2002.
- [365] M. Rohs, 'Real-World Interaction with Camera Phones', in *Second international conference on Ubiquitous Computing Systems*, 2004. [Online]. Available: www.highenergymagic.com
- [366] P. Lightbody, T. Krajník, and M. Hanheide, 'An Efficient Visual Fiducial Localisation System', in *The 32nd Annual ACM Symposium on Applied Computing*, 2017.
- [367] J. Zheng, S. Bi, B. Cao, and D. Yang, 'Visual Localization of Inspection Robot using Extended Kalman Filter and Aruco Markers', in *IEEE International Conference on Robotics and Biomimetics*, 2018.
- [368] L. Jayatilleke and N. Zhang, 'Landmark-Based Localization for Unmanned Aerial Vehicles', in *IEEE International Systems Conference (SYSCON)*, 2013.
- [369] F. Vanegas and F. Gonzalez, 'Enabling UAV navigation with sensor and environmental uncertainty in cluttered and GPS-denied environments', *Sensors (Switzerland)*, vol. 16, no. 5, May 2016, doi: 10.3390/s16050666.
- [370] M. Kalaitzakis, S. Rajan Kattil, N. Vitzilaios, D. Rizos, and M. Sutton, 'Dynamic Structural Health Monitoring using a DIC-enabled drone', in *International Conference on Unmanned Aircraft Systems : ICUAS'19*, 2019.
- [371] A. R. Cantieri, M. A. Wehrmeister, A. S. Oliveira, J. Lima, M. Ferraz, and G. Szekir, 'Proposal of an Augmented Reality Tag UAV Positioning System for Power Line Tower Inspection', in *Advances in Intelligent Systems and Computing*, Springer, 2020, pp. 87–98. doi: 10.1007/978-3-030-35990-4_8.
- [372] L. C. Klein *et al.*, 'A Machine Learning Approach to Robot Localization Using Fiducial Markers in RobotAtFactory 4.0 Competition', *Sensors*, vol. 23, no. 6, p. 3128, Mar. 2023, doi: 10.3390/s23063128.
- [373] G. Zhenglong, F. Qiang, and Q. Quan, 'Pose Estimation for Multicopters Based on Monocular Vision and AprilTag', in *37th Chinese Control Conference*, 2018.

- [374] A. Tsoukalas, A. Tzes, and F. Khorrami, 'Relative Pose Estimation of Unmanned Aerial Systems', in *26th Mediterranean Conference on Control and Automation (MED)*, IEEE, 2018.
- [375] D. Pickem *et al.*, 'The Robotarium: A remotely accessible swarm robotics research testbed', in *ICRA2017 : IEEE International Conference on Robotics and Automation*, 2017.
- [376] M. Bertoni, S. Michieletto, R. Oboe, and G. Michieletto, 'Indoor Visual-Based Localization System for Multi-Rotor UAVs', *Sensors*, vol. 22, no. 15, Aug. 2022, doi: 10.3390/s22155798.
- [377] R. Muñoz-Salinas, M. J. Marín-Jimenez, E. Yeguas-Bolivar, and R. Medina-Carnicer, 'Mapping and localization from planar markers', *Pattern Recognit*, vol. 73, pp. 158–171, Jan. 2018, doi: 10.1016/j.patcog.2017.08.010.
- [378] R. Muñoz-Salinas, M. J. Marín-Jimenez, and R. Medina-Carnicer, 'SPM-SLAM: Simultaneous localization and mapping with squared planar markers', *Pattern Recognit*, vol. 86, pp. 156–171, Feb. 2019, doi: 10.1016/j.patcog.2018.09.003.
- [379] H. Lim and Y. S. Lee, 'Real-Time Single Camera SLAM Using Fiducial Markers', in *ICROS-SICE International Joint Conference*, 2009, p. 177.
- [380] B. Pfrommer and K. Daniilidis, 'TagSLAM: Robust SLAM with Fiducial Markers', Oct. 2019, [Online]. Available: <http://arxiv.org/abs/1910.00679>
- [381] C. Kilic, N. Ohi, Y. Gu, and J. Gross, 'Slip-Based Autonomous ZUPT through Gaussian Process to Improve Planetary Rover Localization', *IEEE Robot Autom Lett*, vol. 6, no. 3, pp. 4782–4789, 2021, doi: 10.1109/LRA.2021.3068893.
- [382] S. Y. Chen, 'Kalman filter for robot vision: A survey', *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4409–4420, 2012, doi: 10.1109/TIE.2011.2162714.
- [383] H. Cho, E. K. Kim, and S. Kim, 'Indoor SLAM application using geometric and ICP matching methods based on line features', *Rob Auton Syst*, vol. 100, pp. 206–224, Feb. 2018, doi: 10.1016/J.ROBOT.2017.11.011.
- [384] F. Shamsfakhr and B. Sadeghi Bigham, 'GSR: geometrical scan registration algorithm for robust and fast robot pose estimation', *Assembly Automation*, vol. 40, no. 6, pp. 801–817, Dec. 2020, doi: 10.1108/AA-09-2017-119/FULL/PDF.
- [385] B. Tao, H. Wu, Z. Gong, Z. Yin, and H. Ding, 'An RFID-Based Mobile Robot Localization Method Combining Phase Difference and Readability', *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 3, pp. 1406–1416, Jul. 2021, doi: 10.1109/TASE.2020.3006724.
- [386] S. Lu, C. Xu, and R. Y. Zhong, 'An Active RFID Tag-Enabled Locating Approach with Multipath Effect Elimination in AGV', *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 3, pp. 1333–1342, Jul. 2016, doi: 10.1109/TASE.2016.2573595.
- [387] K. Naus and Ł. Marchel, 'Use of a weighted ICP algorithm to precisely determine USV movement parameters', *Applied Sciences (Switzerland)*, vol. 9, no. 17, 2019, doi: 10.3390/app9173530.

- [388] Y. T. Wang, C. C. Peng, A. A. Ravankar, and A. Ravankar, 'A single LiDAR-based feature fusion indoor localization algorithm', *Sensors (Switzerland)*, vol. 18, no. 4, 2018, doi: 10.3390/s18041294.
- [389] N. Senin, B. M. Colosimo, and M. Pacella, 'Point set augmentation through fitting for enhanced ICP registration of point clouds in multisensor coordinate metrology', *Robot Comput Integr Manuf*, vol. 29, no. 1, pp. 39–52, 2013, doi: 10.1016/j.rcim.2012.07.003.
- [390] Y. Gao, S. Liu, M. M. Atia, and A. Noureldin, 'INS/GPS/LiDAR integrated navigation system for urban and indoor environments using hybrid scan matching algorithm', *Sensors (Switzerland)*, vol. 15, no. 9, pp. 23286–23302, 2015, doi: 10.3390/s150923286.
- [391] H. Kim, S. Song, and H. Myung, 'GP-ICP: Ground Plane ICP for Mobile Robots', *IEEE Access*, vol. 7, pp. 76599–76610, 2019, doi: 10.1109/ACCESS.2019.2921676.
- [392] B. Sadeghi Bigham, S. Dolatikalan, and A. Khastan, 'Minimum landmarks for robot localization in orthogonal environments', *Evol Intell*, vol. 1, pp. 1–4, May 2021, doi: 10.1007/S12065-021-00616-8/FIGURES/4.
- [393] Y. Y. Yap and B. E. Khoo, 'Landmark-based Automated Guided Vehicle Localization Algorithm for Warehouse Application', *PervasiveHealth: Pervasive Computing Technologies for Healthcare*, pp. 47–54, Sep. 2019, doi: 10.1145/3362752.3365285.
- [394] C. Gentner, T. Jost, W. Wang, S. Zhang, A. Dammann, and U. C. Fiebig, 'Multipath Assisted Positioning with Simultaneous Localization and Mapping', *IEEE Trans Wirel Commun*, vol. 15, no. 9, pp. 6104–6117, 2016, doi: 10.1109/TWC.2016.2578336.
- [395] P. Yang and W. Wu, 'Efficient particle filter localization algorithm in dense passive RFID tag environment', *IEEE Transactions on Industrial Electronics*, vol. 61, no. 10, pp. 5641–5651, 2014, doi: 10.1109/TIE.2014.2301737.
- [396] J. Clemens, T. Kluth, and T. Reineking, ' β -SLAM: Simultaneous localization and grid mapping with beta distributions', *Information Fusion*, vol. 52, no. March 2018, pp. 62–75, 2019, doi: 10.1016/j.inffus.2018.11.005.
- [397] Q. bin Zhang, P. Wang, and Z. hai Chen, 'An improved particle filter for mobile robot localization based on particle swarm optimization', *Expert Syst Appl*, vol. 135, pp. 181–193, 2019, doi: 10.1016/j.eswa.2019.06.006.
- [398] W. Fei, W. Ziwei, and L. Meijin, 'Robot Path Planning Based on Improved Particle Swarm Optimization', *2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering, ICBAIE 2021*, vol. 21, no. 5, pp. 887–891, 2021, doi: 10.1109/ICBAIE52039.2021.9390071.
- [399] Z. Liu, H. Liu, Z. Lu, and Q. Zeng, 'A Dynamic Fusion Pathfinding Algorithm Using Delaunay Triangulation and Improved A-Star for Mobile Robots', *IEEE Access*, vol. 9, pp. 20602–20621, 2021. doi: 10.1109/ACCESS.2021.3055231.
- [400] L. Cheng, C. Liu, and B. Yan, 'Improved hierarchical A-star algorithm for optimal parking path planning of the large parking lot', *2014 IEEE International Conference on Information and Automation, ICIA 2014*, no. July, pp. 695–698, 2014, doi: 10.1109/ICInfA.2014.6932742.

- [401] L. da Silva Costa and F. Tonidandel, 'DVG+A* and RRT Path-Planners: A Comparison in a Highly Dynamic Environment', *Journal of Intelligent & Robotic Systems 2021* 101:3, vol. 101, no. 3, pp. 1–20, Mar. 2021, doi: 10.1007/S10846-021-01326-0.
- [402] J. Wang, B. Li, and M. Q. H. Meng, 'Kinematic Constrained Bi-directional RRT with Efficient Branch Pruning for robot path planning', *Expert Syst Appl*, vol. 170, no. December 2020, p. 114541, 2021, doi: 10.1016/j.eswa.2020.114541.
- [403] K. Wei and B. Ren, 'A method on dynamic path planning for robotic manipulator autonomous obstacle avoidance based on an improved RRT algorithm', *Sensors (Switzerland)*, vol. 18, no. 2, 2018, doi: 10.3390/s18020571.
- [404] C. Tang, R. Sun, S. Yu, L. Chen, and J. Zheng, 'Autonomous Indoor Mobile Robot Exploration Based on Wavefront Algorithm', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11744 LNAI, pp. 338–348, 2019. doi: 10.1007/978-3-030-27541-9_28.
- [405] S. Wu, Y. Du, and Y. Zhang, 'Mobile Robot Path Planning Based on a Generalized Wavefront Algorithm', *Math Probl Eng*, vol. 2020, 2020, doi: 10.1155/2020/6798798.
- [406] A. K. Kashyap, D. R. Parhi, M. K. Muni, and K. K. Pandey, 'A hybrid technique for path planning of humanoid robot NAO in static and dynamic terrains', *Applied Soft Computing Journal*, vol. 96, p. 106581, 2020, doi: 10.1016/j.asoc.2020.106581.
- [407] O. Brock and O. Khatib, 'High-speed navigation using the global dynamic window approach', *Proc IEEE Int Conf Robot Autom*, vol. 1, no. May, pp. 341–346, 1999, doi: 10.1109/robot.1999.770002.
- [408] L. S. Liu *et al.*, 'Path Planning for Smart Car Based on Dijkstra Algorithm and Dynamic Window Approach', *Wirel Commun Mob Comput*, vol. 2021, 2021, doi: 10.1155/2021/8881684.
- [409] D. Fox, W. Burgard, and S. Thrun, 'The dynamic window approach to collision avoidance', *IEEE Robot Autom Mag*, vol. 4, no. 1, pp. 23–33, 1997, doi: 10.1109/100.580977.
- [410] T. Wang, X. Yan, Y. Wang, and Q. Wu, 'A distributed model predictive control using virtual field force for multi-ship collision avoidance under COLREGs', *2017 4th International Conference on Transportation Information and Safety, ICTIS 2017 - Proceedings*, pp. 296–305, 2017, doi: 10.1109/ICTIS.2017.8047780.
- [411] E. Burgos and S. Bhandari, 'Potential flow field navigation with virtual force field for UAS collision avoidance', *2016 International Conference on Unmanned Aircraft Systems, ICUAS 2016*, pp. 505–513, 2016, doi: 10.1109/ICUAS.2016.7502641.
- [412] J. Borenstein and Y. Koren, 'the Vector Field Histogram - Fast obstacle avoidance for mobile robots', *IEEE Journal of Robotics and Automation*, vol. 7, no. 3, pp. 278–288, 1991.
- [413] L. Wang, 'Automatic control of mobile robot based on autonomous navigation algorithm', *Artif Life Robot*, vol. 24, no. 4, pp. 494–498, 2019, doi: 10.1007/s10015-019-00542-0.

- [414] C. Ye, 'Navigating a mobile robot by a traversability field histogram', *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, no. 2, pp. 361–372, 2007, doi: 10.1109/TSMCB.2006.883870.
- [415] A. Abdelmoniem, A. Osama, M. Abdelaziz, and S. A. Maged, 'A path-tracking algorithm using predictive Stanley lateral controller', *Int J Adv Robot Syst*, vol. 17, no. 6, pp. 1–11, 2020, doi: 10.1177/1729881420974852.
- [416] A. Adadi and M. Berrada, 'Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)', *IEEE Access*, vol. 6, pp. 52138–52160, Sep. 2018, doi: 10.1109/ACCESS.2018.2870052.
- [417] S. M. Abbas, S. Aslam, K. Berns, and A. Muhammad, 'Analysis and Improvements in AprilTag Based State Estimation', *Sensors 2019, Vol. 19, Page 5480*, vol. 19, no. 24, p. 5480, Dec. 2019, doi: 10.3390/S19245480.
- [418] K. O. Stanley and R. Miikkulainen, 'Evolving Neural Networks Through Augmenting Topologies'. pp. 99–127. Accessed: Nov. 23, 2023. [Online]. Available: <http://mitpress.mit.edu/e-mail>
- [419] 'Camera Calibration - MATLAB & Simulink - MathWorks España'. Accessed: Oct. 19, 2023. [Online]. Available: <https://es.mathworks.com/help/vision/camera-calibration.html>
- [420] 'Object Detection Using Faster R-CNN Deep Learning - MATLAB & Simulink - MathWorks España'. Accessed: Nov. 03, 2023. [Online]. Available: <https://es.mathworks.com/help/vision/ug/object-detection-using-faster-r-cnn-deep-learning.html>
- [421] 'Label images for computer vision applications - MATLAB - MathWorks España'. Accessed: Nov. 05, 2023. [Online]. Available: <https://es.mathworks.com/help/vision/ref/imagelabeler-app.html>
- [422] 'Getting Started with R-CNN, Fast R-CNN, and Faster R-CNN - MATLAB & Simulink - MathWorks España'. Accessed: Nov. 05, 2023. [Online]. Available: <https://es.mathworks.com/help/vision/ug/getting-started-with-r-cnn-fast-r-cnn-and-faster-r-cnn.html>
- [423] M. Cabezas-Olivenza, E. Zulueta, A. Sánchez-Chica, A. Teso-Fz-betoño, and U. Fernandez-Gamiz, 'Dynamical analysis of a navigation algorithm', *Mathematics*, vol. 9, no. 23, Dec. 2021, doi: 10.3390/math9233139.
- [424] M. Cabezas-Olivenza, E. Zulueta, A. Sanchez-Chica, U. Fernandez-Gamiz, and A. Teso-Fz-Betoño, 'Stability Analysis for Autonomous Vehicle Navigation Trained over Deep Deterministic Policy Gradient', *Mathematics*, vol. 11, no. 1, Jan. 2023, doi: 10.3390/math11010132.
- [425] A. Teso-Fz-Betoño, E. Zulueta, M. Cabezas-Olivenza, D. Teso-Fz-Betoño, and U. Fernandez-Gamiz, 'A Study of Learning Issues in Feedforward Neural Networks', *Mathematics*, vol. 10, no. 17, Sep. 2022, doi: 10.3390/math10173206.
- [426] A. Teso-Fz-Betoño, E. Zulueta, M. Cabezas-Olivenza, U. Fernandez-Gamiz, and C. Botana-M-Ibarreta, 'Modification of Learning Ratio and Drop-Out for Stochastic

Gradient Descendant Algorithm', *Mathematics*, vol. 11, no. 5, Mar. 2023, doi: 10.3390/math11051183.

- [427] M. Graña, 'Basque Conference on Cyber Physical Systems and Artificial Intelligence', May 2022, doi: 10.5281/ZENODO.7039276.

ERANSKINAK

GEHIGARRIAK

1. I. Eranskina: Baldintza-agiria

Universidad del País Vasco/Euskal Herriko Unibertsitatea (UPV/EHU) -ari, Sistemen Ingeniaritza eta Automatika sailari eta lan honen zuzendaria izan den Ekaitz Zulueta Guerrero irakasleari, baimena ematen zaie bai dokumentu honen, bai proiektua aurrera eramateko diseinatu den kodigoaren erabilera eta aldaketak egiteko.

2. II. Eranskina: Gidaliburua

Eranskin honetan kodigoen funtzionamendua azaltzen da. Hasteko, gomendagarria da atal bakoitzeko fitxategi guztiak direktorio berean gordeta izatea, baina hau ezin bada egin, kontsiderazio batzuk izan beharko dira hurrengo eranskinean erakutsitako kodigoekiko. Kodigo guztiak exekutatu ahal izateko gainera, MATLAB programa instalatuta beharrezkoa dela argi dago, 2022b bertsiotik gorakoa izanda, bestela ez delako bateragarria izango zenbait funtzioekin.

2.1. Errefortzu bidezko Ikaskuntzaren Optimizazioa

Errefortzu bidezko ikaskuntza optimizatzeko bi era aurkeztu dira, exekuzioa indibiduala izanda.

2.1.1. OPTIMIZAZIO ANALITIKOA

Optimizazio analitikoa aurrera eramateko, programa nagusia *GURERLosorikV7v4.mlx* exekutatu behar da. Kodigo honetan lehendabizi antzezlearen sarea definitzen da, entrenamendu prozesuari pausu emateko. Aldagaiak hasieratzen dira eta entrenamendu garaien begizta aurkitu daiteke. *hasieradef3.m*-ri deitzen zaio hasierako posizioa definitzeko. Posizio hau iterazio bakoitzean eguneratuko da eta mapako puntu guztietatik pasaraziko da entrenatzeko. Bigarren artxibo honetan mapa definituko da ere, helmugarekin batera. Begizta berri bat dago, iterazioko sarearen balioak kalkulatzeko. Honetan “*stepfunction*”-a exekutatzen da eta horrekin sentsibiltate ekuazioa kalkulatu da sarearen parametroak eguneratzeko. Ondoren, zenbait funtzio aurki daitezke sentsibiltate ekuazioaren taulen elkarketak egiteko.

Behin entrenamendua amaituta dagoela “*NNactor*” sarearen aldagaia gorde egingo da eta *bidekalkulua.m* artxiboan kargatuko da. Kodigo honek kalkulatuako sarea proban jartzen du ibilbide bat aurreikus dezan ingurune batean.

2.1.2. EBOLUZIO BIDEZKO OPTIMIZAZIOA

Sare neuronalaren eboluzioaren prozesua exekutatze bost fitxategi ezberdin erabili dira. Garrantzitsuena *main.mlx* izeneko da, bertan egiten delako sarearen eboluzioa eta entrenamenduaren deitzea. Beraz, artxibo hau exekutatzearekin nahikoa da prozesu osoari ekiteko. Hasteko aldagaiak definitzen dira eta zuzenean begizta batean sartzen da, non sare neuronalaren eboluzioa gertatzen den. Hau string-en bidez egiten da eta behin sarea sortuta *ReinforcementLearning.m*-ri deitzen zaio agentea entrena dezan. Bigarren honetan, hasteko ingurunea sortzen da eta horretarako *myResetFunction.m* eta *myStepFunction.m* erabiltzen dira. Hauetako lehengoa ingurunea berriztatu egiten du garai bakoitzeko. Bigarrenak ibilgailuaren dinamika, obserbazioen eguneraketa eta sari funtzioaren kalkulua egiten du. Ondoren, sare neuronalak sortzen dira, kritikorena hain zuzen ere, antzezlearena funtzioaren sarrera bat bait da. Agentearen parametroak definitzen dira eta entrenamendua abiarazten da. Entrenamendua gelditzerakoan baldintzak bete diren miatzen da, eboluzioarekin aurrera egiteko edo eten egiteko.

Guzti hau era egokian funtzionatu dezan *hasieradef.m* artxiboaren beharra dago. Bertan inguruneke mapa definituta dago eta hasierako eta amaierako posizioen balioak definitzen dira. Gainera hasierako posizioei dagokionez, begizta bat diseinatu da oztopoen gainean entrenamendua ez hasteko. Azkenik, exekuzio nagusian simulazio bat egiten da agenteak kalkulatzeko dituen ibilbideak bistartzeko.

Algoritmo genetiko bidezko optimizazioa

Nabigazioa gauzatzeko kodigoa algoritmo genetikoaren bidez ere egin da. Honetarako bi artxibo erabiltzen dira. Batetik *hasieradef3.m*, aurreko ataletatik berreskuratua eta posizioen hasieraketa egiten duena. Bestetik *BasqueNavigationFuntz.m*, eta hau da exekutatu behar den fitxategi nagusia. Honetan lehendabizi algoritmo genetikoaren opzioak finkatzen dira, hasieraketekin batera. Ondoren, K balioa optimizatzen da eta horretarako begizta bat dago. Horretan, metodologian azaldutako prozesua gauzatzen da. Azkenik, funtzio berdina abiarazten da bidearen kalkulua egiteko K balioarekin.

2.2.SLAM bisuala

SLAM bisuala lortzeko zenbait pausu jarraitu behar dira. Hasteko eta garrantzi handien duena, kameraren kalibrazioa da. Hau egiteko MATLAB-en eskuragarri dagoen kodigoa erabiltzen da, <https://es.mathworks.com/help/vision/ug/camera-calibration-using-apriltag-markers.html> webgunean eskuragarri dagoena. *CalibracionCamaraConAprilTag.mlx* izenarekin gorde da kasu honetan. Bertan txantiloia dago eskuragarri eta hau inprimatu beharra dago. Txantiloari irudiak atera behar zaizkio eta horretarako *guardarimagenes.m* artxiboa erabili da. Honetan irudiak non gorde nahi diren finkatu behar da eta 4 segundoko irudi bat eskuratzen du kalibrazioa egin ahal izateko. Behin nahi diren irudi guztiak aterata daudenean kalibratzeko kodigoari adierazi behar zaizkio non gordeta dauden eta prozesuari ekiten dio. Azkenean “*CamParamsAprilTagCalibration*” aldagaia lortzen da, eta hau gorde behar da kamera erabili nahi den aplikaziorako.

Aplikazio hori kasu honetan mapa eraikitzea da eta horretarako *V3AprilTagDeteccionMovimiento.mlx* erabiltzen da. Kodigo honetan lehendabizi kamera konfiguratu da eta ondoren AprilTag-ak gordeko dituen taula definitzen da. Ondoren etiketa absolutuak detektatuko dira eta .mlx-eko 3.atala exekutatu behar da. Irudi bat ateratzen da eta bertan dauden etiketen detekzioa egiten da. Etiketa bakoitzarekiko irakurri diren balioak kalkulatzeko dira eta datu taulan biltzen dira. Jarraitzeko ibilgailua mugitu behar da eta 4.atala exekutatu behar da mugitzen den aldi bakoitzeko, AprilTag berriak gordetzen joateko. Hemen absolutuen pausu berdina jarraitzen da baina metodologian azaldu den absolutu-erlatibo aldaketarekin. Horretarako lehendabizi erreferentzia absolutuko ezberdintasuna kalkulatzeko da ezaguna den AprilTag bat erabilia eta ondore berriei aplikatzen zaie. Azkenik, taulan gordeta dauden etiketa guztiak ikusarazten dira, maparen eraiketarekin amaitzeko. Antzeman daitekenez eskuzko prozesua da eta erabiltzailea exekutatu behar du pausoz-pauso eta ordena jarraituz.

2.3. Oztopen distantzien kalkulua

Oztopen distantzia lortu arte eta azaldu den bezala, zenbait pausu ezberdin jarraitu behar dira. Lehenengo datu basearekin erlazioa daukate. Kasu honetan KITTI datu basea erabili da eta webgune ofizialetik deskargatu da, <https://www.cvlibs.net/datasets/kitti/> helbidetik hain zuzen ere. Bertan “*depth*” erlaitzean “*Download projected raw LiDAR scans data set*” aukeratzen da eta deskarga egiten da. Honekin batera “*Download development kit*” ere jaitzi da. Lehenengo artxiboan daude erabili diren irudiak, barnean dagoen “*test_depth_completion_anonymous*” karpeta hain zuzen ere. Bertan dauden “*image*” eta “*velodyne_raw*” barruan dauden irudiak erabiliko dira.

Lehenengo eragiketa beraz irudien leuntzea da eta *suavizado.m* kodigoa erabiltzen da. Kodigo honetan “*velodyne_raw*”-ko irudien direkzioa finkatzen da eta irudi leunduak gordeko diren direktorioa. Honen ondoren leihoaren tamaina inkatzen da eta irudiz irudi leunketa egiten den begizta bat dago. Azkenean izen berdinarekin gordeko dira.

Hurrengo pausua bai *recorteDataAugmentation.m* exekutzea da. Lehenik eta behin “*image*” karpeta barruan dauden RGB irudiei aplikatuko zaie, beraz hauen direkzioa finkatu behar da. Datu basean 1000 irudi daude eta 4000 lortzeko pausu hau lau aldiz exekutatu behar da. Lehenengoan $N=0$ -ra finkatuz, bigarrenean $N=1000$, hirugarrenean $N=2000$ eta azkenik, $N=3000$. Hau irudiak ez zapaltzeko egiten da. Lehenengo exekuzioan irudia bakarrik moztu egiten da tamaina adaptatzeko beraz 23 eta 24 lerroak komentatuta geratzen dira. Bigarren exekuzioan irauli egiten dira beraz 23 lerroa aktibatzen da. Hurrengo exekuzioan berriz ere 23 lerroa komentatzen da eta 24-a aktibatzen da distira eta kontrastea moldatzeko. Azken exekuzioan guztia deskomentatzen da. Horrela irudi moldatuak lortzen dira, irudiko erdiko gunea aukeratuz. Orain direkzioa aldatzen da, “*velodyne_raw*”-ko LiDAR irudiekin eta pausua errepikatzen dira 24 lerroa beti desaktibatuta egonda.

Aurreko pausutik irudiak “*image_(zenbakia)*” izenarekin irtengo dira eta hau ez da baliagarria entrenamendurako, irudi nahasketa bat sortzen delako ez dagoelako karakter kopuru berdina izen bakoitzean beraz *nameAdapt.m* exekutatzen da. Irudien karpeta direkzioa adierazten da eta gorde nahi den direkzioa ere. Ondoren banan-banan irudiak hartzen dira eta zenbaki karakter berdinak finkatzen dira irudi guztientzako. Hau bai “*image*”-ko 4000 irudiekin bai “*velodyne_raw*”-ko 4000 irudiekin egin behar da beraz bi exekuzio behar ditu helbide ezberdinekin. Behin hau izanda, eskuz test, entrenamendu eta balidazio datuak banatuko dira 3 karpeta ezberdinetan.

Behin datu basea osaturik dagoela sareen entrenamenduetara pasatzen da. Lehenengo, hau da, sakontasunaren sarea *GureSareGureGeruzarekin.m* kodigoan entrenatzen da. Artxibo hoetako lehenengo parte LiDAR -eko irudiei, bai balidazio bai entrenamendukoei neurri ulergarria ematea da. Horretarako hauen helbideak finkatuko dira eta *depth_read.m*-tik pasako dira. Kodigo hau KITTI webgunetik deskargatutako bigarren artxiboan aurkitu daiteke. Ondoren *dataset*-a eratuko da RGB irudiak eta sakontasuneko irudiak konbinatuz. *GureGeruzakoFcn.m*-ri deitzen zaio, sarea sortzeko eta hau kontuan izateko. Esan beharra dago, 63 eta 80 lerroen artean dagoen kodigoa komentatu edo deskomentatuko dela sarearen eraldaketak egiteko, konparazioetarako erabili direnak. Entrenamendura pasatzen da eta galera ebaluatzeko *customLoss.m*-ri deitzen zaio iterazio bakoitzean. Horrela, distantziak aurreikusten dituen sare bat lortzen da *SareHandia* aldagaian, gorde egin beharko dena.

Jarraitzeko *objectdetection.m* kodigoa exekutatzen da. Kodigo hau esan bezala MATLAB-eko webgunetik lortzen da baina moldatu egin da nahi izan diren klaseak detektatzeko. Erabiltzen duen *heperSanitizeBoxes.m* ere MATLAB-etik deskargatzen da eta hoetan ez da aldaketarik egiten. Honetarako bakarrik RGB irudiak beharko dira beraz direkzio hori emango zaio. Gainera, *Image Labeler*-etik esportatutako datuak ere beharko dira, *bbox*-ak eta klaseak dituenak. Hasteko *dataset*-ak eratuko dira informazio guztia bilduz eta ondoren etiketarik ez dituen irudiak baztertuko dira. Sarea sortu eta entrenatu egiten da eta lortutako *detector* aldagaia gordeko da.

Azkenik, *depthcalculation.m* artxiboa dago. Honetarako bi sareen aldagaiak kargatuko dira eta testatzeko gorde diren irudiak. Begizta bat egiten da irudi bakoitza bi sareetatik pasatzeko eta objektuen detekzioa eta hauen distantzia estimazioa kalkulatzeko.

3. III. Eranskina: Kodigoa

Lan honen garapena egiteko kodigo ezberdinen erabileraren beharra existitu da, ibilgailu autonomo industrialen sistemen diseinuak aurrera eraman ahal izateko. Ondoren, kodigo hauek erantsita agertzen dira, erabilera ordenan sailkatuta. Gogoratzekoa da, MATLAB plataforma erabili dela guzti hauen garapenean.

3.1. Errefortzu bidezko Ikaskuntzaren Optimizazioa

Atal hau bi garapenez osatuta dago metodologian antzeman den bezala eta bakoitza aurrera eramateko kodigo ezberdinak erabili dira.

3.1.1. OPTIMIZAZIO ANALITIKOA

Lehendabizi, optimizazio analitikoa aurrera eramateko kodigoak aurkezten dira, bi artxiboz osatuta egonda.

GureRLosorikv7v4.mlx.

Basque RL algorithm

```
clear all
close all
clc
Parameters:
numObservations=6;%x y xh-x yh-y xo-x yo-y
Actor's neural network equations:
actorNetwork = [
    featureInputLayer(numObservations,'Normalization','none','Name','observation')
    fullyConnectedLayer(50,'Name','ActorFC1')
    reluLayer('Name','ActorRelu1')
    fullyConnectedLayer(50,'Name','ActorFC2')
    reluLayer('Name','ActorRelu2')
    fullyConnectedLayer(1,'Name','ActorFC3')
    tanhLayer('Name','ActorTanh')
    scalingLayer('Name','ActorScaling','Scale',pi/2)];

NNactor=dlnetwork(actorNetwork)
Training algorithm:
k is the kth training epoch. In a epoch we have a simulation, so the t time changes from 0 to  $T_{\text{horizont}}$ .
Training loop:
Niter=100; %How many epochs we are going to apply (1 epoch implies 1 simulation and 1 actualization)
Tsimul=100;%How many simulation steps we are going to apply
Vmax=1;
dt=0.1;
S=[];
velocity=[];
initialLearnRate = 1e-6;
decay = 0.75;
momentum = 0.95;
dGdw=[];

xvec=[];
yvec=[];
for epochs=1:Niter
    %Simulation step
```

```

aldagaiak=@hasieradef3;
[~,allpos,~,~] = aldagaiak();
for posizioak=1:3
    %reset function
    hasierapos=allpos(:,posizioak);
    initX=hasierapos(1);
    initY=hasierapos(2);

    [Measuresinit]=reset(initX,initY); %sei obserbazioak irtetzen dira

    Measures=dlarray(Measuresinit, 'C');
    pos=extractdata(Measures(1:2));
    index=1;
    xvec(index)=pos(1);
    yvec(index)=pos(2);
    u_ant=0;
    u_ant=dlarray(u_ant, 'C');
    for t=1:Tsimul
        [Snew,u,drdx,dGdw,Measuresnew,IsDone] =
dlfeval(@modelRewardSensibilityXw,NNactor,Measures,S,u_ant,Vmax,dt,dGdw);
        u_ant=u;
        Measures=Measuresnew;
        S=Snew;
        pos=extractdata(Measures(1:2));

        index=index+1;
        xvec(index)=pos(1);
        yvec(index)=pos(2);
        if IsDone ==1
            break
        end
    end
end

%Weigth actualization
gradients=ChangeSignTableofDL(dGdw);
% Determine learning rate for time-based decay learning rate schedule.
learnRate = initialLearnRate/(1 + decay*epochs);
pisuzaharrak=extractdata(NNactor.Learnables.Value{1});
[NNactor,velocity] = sgdupdate(NNactor,gradients,velocity,learnRate,momentum);%actualize weights
pisuBerriak=extractdata(NNactor.Learnables.Value{1});
end
simulazioa(xvec,yvec);
a= 100*epochs/Niter

end
Execution
simulation for each epoch
function []=simulazioa(xvec,yvec)

    figure
    plot(xvec,yvec, 'b')
    hold on
    grid on
    aldagaiak=@hasieradef3;
    [mapa,~,finX,finY] = aldagaiak();
    plot(mapa(1,:),mapa(2,:), 'xb')
    hold on
    plot(xvec(end),yvec(end), 'or')
    xlabel('Position x [m]')
    ylabel('Position y [m]')
    hold on
    plot(finX,finY, '-p', 'MarkerFaceColor', 'red', 'MarkerSize', 15)
    pause(1)

end
Dynamical system's equations:
t is the the time in a simulation time.
function [Measures_new]=StepFunction(Measures,u,Vmax,dt)
    pos=extractdata(Measures(1:2));
    deltaX=Vmax*[cos(u);sin(u)]*dt;
    pos_new=pos+deltaX;
    pos_new=extractdata(pos_new);

aldagaiak=@hasieradef3;

```

```

[mapa,~,finX,finY] = aldagaiak();
DistMap=zeros(1,length(mapa));
Xhelmuga=[finX;finY];
for k=1:length(mapa)
    DistMap(k)=norm(mapa(:,k)-pos_new);
end
[gertukoena,rfmin]=min(DistMap);
xh=finX-pos_new(1);
yh=finY-pos_new(2);
xo=(mapa(1,rfmin))-pos_new(1);
yo=(mapa(2,rfmin))-pos_new(2);

Measures_new=dlarray([pos_new;xh;yh;xo;yo], 'C');
end
Critic's equation:
function
[Snew,u,drdx,dGdw,Measures_new,IsDone]=modelRewardSensibilityXw(NNactor,Measures,S,u_ant,Vmax,dt,dGdw)
u=predict(NNactor,Measures);
[Measures_new]=StepFunction(Measures,u,Vmax,dt);

[drdx,IsDone]=reward(Measures_new);

dfstepdu=[-sin(u);cos(u);sin(u);-cos(u);sin(u);-cos(u)];%obserbazio berriak zela sartu???
dfdx=[1 0 0 0 0 0;
      0 1 0 0 0 0;
      -1 0 0 0 0 0;
      0 -1 0 0 0 0;
      -1 0 0 0 0 0;
      0 -1 0 0 0 0];

dudw=dlgradient(u,NNactor.Learnables);%asi no da 0
[filaS,colS]=size(S);

if filaS*colS==0
    Snew=dxdwCalculation(dfstepdu,dudw);
    dfdw=dfdwCalculation(dfdx,Snew,dudw);
    dGdw=dGdwCalculation(drdx,dfdw,dudw);
else
    Snew=AddTablesOfDL(dxdwCalculation(dfstepdu,dudw),S);
    dfdw=dfdwCalculation(dfdx,Snew,dudw);
    dGdw=AddTablesOfDL(dGdw,dGdwCalculation(drdx,dfdw,dudw));
end
end
Reward function:
function [drdx,IsDone]=reward(Measures)
pos=extractdata(Measures(1:2));
aldagaiak=@hasieradef3;
[mapa,~,finX,finY] = aldagaiak();
DistMap=zeros(1,length(mapa));

for k=1:length(mapa)
    DistMap(k)=norm(mapa(:,k)-pos);
end
[gertukoena,rfmin]=min(DistMap);

%-----
%REWARD
a1=20;
a2=100;

%drdx=drdmeasures*dmeasuresdx
finmod=((finX-pos(1))^2)+(finY-pos(2))^2;
xo=(mapa(1,rfmin));%-pos(1);
yo=(mapa(2,rfmin));%-pos(2);

drdposx=(-a1*2*(finX-pos(1)))/(finmod^2)+(-a2*2*(xo-pos(1)));
drdposy=(-a1*2*(finY-pos(2)))/(finmod^2)+(-a2*2*(yo-pos(2)));
drdxhx=(a1*2*(finX-pos(1)))/(finmod^2);
drdyhy=(a1*2*(finY-pos(2)))/(finmod^2);
drdxox=a2*2*(xo-pos(1));
drdyoy=a2*2*(yo-pos(2));
drdx=[drdposx,drdposy,-drdxhx,-drdyhy,-drdxox,-drdyoy];

IsDone=0;

```



```

if pos(1)<=0 || pos(1)>=14 %dependen de la zona del mapa
    IsDone=1;
end
if pos(2)<=0 || pos(2)>=14
    IsDone=1;
end
if pos(1)>=10 && pos(2)<=10
    IsDone=1;
end
    DistmaxGertuEgoteko=0.3;
if gertukoena<DistmaxGertuEgoteko
    IsDone=1;
end
if pos(1)==12 && pos(2)==12
    IsDone=1;
end
end
Neurketak hasieraketa funtzioa
function [Measuresinit]=reset(initX,initY)
    aldagaiak=@hasieradef3;
    [mapa,~,finX,finY] = aldagaiak();
    DistMap=zeros(1,length(mapa));
    Xhelmuga=[finX;finY];
    for k=1:length(mapa)
        DistMap(k)=norm(mapa(:,k)-[initX;initY]);
    end
    [gertukoena,rfmin]=min(DistMap);
    xh=finX-initX;
    yh=finY-initY;
    xo=(mapa(1,rfmin))-initX;
    yo=(mapa(2,rfmin))-initY;

    Measuresinit=[initX;initY;xh;yh;xo;yo];
end
support functions
function [dxdw]=dxdwCalculation(dfstepdu,dudw)
    dxdw=dudw;
    NumberofBlocks=length(dudw.Value);
    for i=1:NumberofBlocks
        M=extractdata(dudw.Value{i});
        M1=extractdata(dfstepdu*M(:)');
        dxdw.Value{i}=dlarray(M1);
    end
end

function [dfdww]=dfdwwCalculation(dfdx,Snew,dudw)
    dfdw=dudw;
    NumberofBlocks=length(dudw.Value);
    for i=1:NumberofBlocks
        M=extractdata(Snew.Value{i});
        M1=dfdx*M;
        dfdw.Value{i}=dlarray(M1);
    end
end

function [ActualizationdGdw]=dGdwCalculation(drdx,Snew,dudw)
    ActualizationdGdw=dudw;
    NumberofBlocks=length(dudw.Value);
    for i=1:NumberofBlocks
        [f,c]=size(dudw.Value{i});
        M=extractdata(Snew.Value{i});
        M1=drdx*M;
        M1=reshape(M1',[f,c]);
        ActualizationdGdw.Value{i}=dlarray(M1);
    end
end

function [T1plusT2]=AddTablesOfDL(T1,T2)
    T1plusT2=T1;
    NumberofBlocks=length(T1.Value);
    for i=1:NumberofBlocks
        M1=extractdata(T1.Value{i});
        M2=extractdata(T2.Value{i});
    end
end

```

```

        M1plus2=M1+M2;
        T1plusT2.Value{i}=dlarray(M1plus2);
    end
end

function [Tschanged]=ChangeSignTableofDL(T)
    Tschanged=T;
    NumberofBlocks=length(T.Value);
    for i=1:NumberofBlocks
        M=-extractdata(T.Value{i});
        Tschanged.Value{i}=dlarray(M);
    end
end
end

```

hasieradef3.m

```

function [mapa,allpos,finX,finY]=hasieradef3()

%Mapa eraikitzen da ingurunearen limiteak eta oztupoak definituz
%limiteak
ParedInf1=(0:0.5:10);
ParedInf1(2,:)=zeros(1,length(ParedInf1));
ParedInf2=(10.5:0.5:14);
ParedInf2(2,:)=ones(1,length(ParedInf2))*10;
ParedInf=[ParedInf1,ParedInf2];
ParedSup=(0:0.5:14);
ParedSup(2,:)=ones(1,length(ParedSup))*14;
ParedIzq(2,:)=(0:0.5:14);
ParedIzq(1,:)=zeros(1,length(ParedIzq));
ParedDer1(2,:)=(0:0.5:10);
ParedDer1(1,:)=ones(1,length(ParedDer1))*10;
ParedDer2(2,:)=(10.5:0.5:14);
ParedDer2(1,:)=ones(1,length(ParedDer2))*14;
ParedDer=[ParedDer1,ParedDer2];
Paredes=[ParedInf,ParedSup,ParedIzq,ParedDer];
mapa=[Paredes];

finX=12;
finY=12;

postotal1gunea=[];
x=double(0.5):0.5:double(9.5);
for i=1:length(x)
    y=double(0.5):0.5:double(13.5);
    pos=x(i)*ones(1,length(y));
    hasierapos=[pos;y];
    postotal1gunea=[postotal1gunea,hasierapos];
end
postotal2gunea=[];
x=double(10):0.5:double(13.5);
for i=1:length(x)
    y=double(10.5):0.5:double(13.5);
    pos=x(i)*ones(1,length(y));
    hasierapos=[pos;y];
    postotal2gunea=[postotal2gunea,hasierapos];
end
allpositions=[postotal1gunea,postotal2gunea];
indx=find(allpositions(1,:)==finX);
indy=find(allpositions(2,:)==finY);
val=intersect(indx,indy);
allpositions(:,val)=[];

%mix
mix=randperm(size(allpositions,2));
allpos=allpositions(:,mix);
end

```

bidekalkulua.m

```

load("NNactor.mat")
initX=1;
initY=1;
[Measuresinit]=reset(initX,initY);
xvec=[];
yvec=[];
Measures=dllarray(Measuresinit,'C');
pos=extractdata(Measures(1:2));
Vmax=1;
dt=0.1;
index=1;
xvec(index)=pos(1);
yvec(index)=pos(2);

for t=1:100
    u=predict(NNactor,Measures);
    pos=extractdata(Measures(1:2));
    delta=Vmax*[cos(u);sin(u)]*dt;
    pos_new=pos+delta;
    pos_new=extractdata(pos_new);

    aldagaiak=@hasieradef2;
    [mapa,~,~,finX,finY] = aldagaiak();
    DistMap=zeros(1,length(mapa));
    Xhelmuga=[finX;finY];
    for k=1:length(mapa)
        DistMap(k)=norm(mapa(:,k)-pos_new);
    end
    [gertukoena,rfmin]=min(DistMap);
    xh=finX-pos_new(1);
    yh=finY-pos_new(2);
    xo=(mapa(1,rfmin))-pos_new(1);
    yo=(mapa(2,rfmin))-pos_new(2);

    Measures_new=dllarray([pos_new;xh;yh;xo;yo],'C');
    Measures=Measures_new;

    index=index+1;
    xvec(index)=pos_new(1);
    yvec(index)=pos_new(2);
end

figure
plot(xvec,yvec,'b')
hold on
grid on
aldagaiak=@hasieradef2;
[mapa,~,~,finX,finY] = aldagaiak();
plot(mapa(1,:),mapa(2:,:),'xb')
hold on
plot(xvec(end),yvec(end),'or')

function [Measuresinit]=reset(initX,initY)
    aldagaiak=@hasieradef2;
    [mapa,~,~,finX,finY] = aldagaiak();
    DistMap=zeros(1,length(mapa));
    Xhelmuga=[finX;finY];
    for k=1:length(mapa)
        DistMap(k)=norm(mapa(:,k)-[initX;initY]);
    end
    [gertukoena,rfmin]=min(DistMap);
    xh=finX-initX;
    yh=finY-initY;
    xo=(mapa(1,rfmin))-initX;
    yo=(mapa(2,rfmin))-initY;

    Measuresinit=[initX;initY;xh;yh;xo;yo];
end

```

3.1.2. EBOLUZIO BIDEZKO OPTIMIZAZIOA

Bigarren prozesua sare neuronala eboluzionatzea izan da eta horretako zenbait kodigo erabili dira.

main.mlx

REINFORCEMENT LEARNING (DDPG) OPTIMIZATUA GENETIK ALGORITM

```

clc
clear all
close all

isDone=0;
isOk=0;
aldagaiak=@hasieradef;
[mapa,~,~,finX,finY] = aldagaiak();

LOOP
kapakopuru=8;
numObservations=6;
while isOk==0
    %sarea sortzen da
    comando="NNactorlayers =
[featureInputLayer(numObservations,'Normalization','none','Name','observation')";
    capanueva=" fullyConnectedLayer(50) reluLayer";
    % fullyconnected+Relu bikoteak gehitzen zaizkio NNactorlayers eri
    for i=1:kapakopuru
        comando=strcat(comando,capanueva);
    end
    terminacion=" fullyConnectedLayer(1,'Name','ActorFC3') tanhLayer('Name','ActorTanh')
scalingLayer('Name','ActorScaling','Scale',max(pi));";
    comando=strcat(comando,terminacion);
    eval(comando);
    % pause
    [isOk,env,agent]=ReinforcementLearning(NNactorlayers);
    close all
end

%simulazioa
simOptions = rlSimulationOptions('MaxSteps',5000);
experience = sim(env,agent,simOptions);
TT=ts2timetable(experience.Observation.obs1);
X=TT.obs1(:,1);
Y=TT.obs1(:,2);
plot(X,Y,'b')
hold on
grid on
plot(mapa(1,:),mapa(2,),'xb')
hold on
plot(X(end),Y(end),'or')
hold on
plot(finX,finY,'-p','MarkerFaceColor','red','MarkerSize',15)

```

ReinforcementLearning.m

```

function [isOk,env,agent] = ReinforcementLearning(NNactorlayers)
%% ingurunea sortu
Tf=100;
Ts=0.1;

%ze parametro konprobatu nahi diren
obsInfo = rlNumericSpec([6 1]);% (x y xh yh xo yo)
numObservations = obsInfo.Dimension(1);

%zein parametrotan eragingo den
numActions=1; %thita abiadura
actInfo = rlNumericSpec([1 1],...
    "LowerLimit",-pi/2,...
    "UpperLimit",pi/2);

%Ingurunea sortzen da

```

```

[InitialObservation,LoggedSignals] = myResetFunction()
env = rlFunctionEnv(obsInfo,actInfo,'myStepFunction','myResetFunction');

%% kritikoa sortu
hiddenLayerSize = 10;

observationPath = [
    featureInputLayer(numObservations,'Normalization','none','Name','observation')
    fullyConnectedLayer(hiddenLayerSize,'Name','fc1')
    reluLayer('Name','relu1')
    fullyConnectedLayer(hiddenLayerSize,'Name','fc2')
    additionLayer(2,'Name','add')
    reluLayer('Name','relu2')
    fullyConnectedLayer(hiddenLayerSize,'Name','fc3')
    reluLayer('Name','relu3')
    fullyConnectedLayer(1,'Name','fc4')];
actionPath = [
    featureInputLayer(numActions,'Normalization','none','Name','action')
    fullyConnectedLayer(hiddenLayerSize,'Name','fc5')];

% Create the layer graph.
criticNetwork = layerGraph(observationPath);
criticNetwork = addLayers(criticNetwork,actionPath);

% Connect actionPath to observationPath.
criticNetwork = connectLayers(criticNetwork,'fc5','add/in2');
criticOpts = rlOptimizerOptions('LearnRate',0.001,'GradientThreshold',1,'L2RegularizationFactor',0.01);
critic = rlQValueFunction(criticNetwork,obsInfo,actInfo,"Observation",{ 'observation'},"Action",{ 'action'});

%% Antzezlea kargatu eta sortu
actorNetwork=NNactorlayers;%sarrera, GA-etik etortzen dena
actorOpts = rlOptimizerOptions('LearnRate',0.001,'GradientThreshold',1,'L2RegularizationFactor',0.01);
actor = rlContinuousDeterministicActor(actorNetwork,obsInfo,actInfo);

%% Agentea sortu
agentOpts = rlDDPGAgentOptions(...
    "SampleTime",Ts,...
    "TargetSmoothFactor",1e-3,...
    "DiscountFactor",0.995, ...
    "MiniBatchSize",128, ...
    "ExperienceBufferLength",1e6);

agentOpts.NoiseOptions.Variance = 0.1;
agentOpts.NoiseOptions.VarianceDecayRate = 1e-5;

agent = rlDDPGAgent(actor,critic,agentOpts);

%% Entrenamendua egin
maxEpisodes = 1000;
maxSteps = ceil(Tf/Ts);
trainOpts = rlTrainingOptions(...
    "MaxEpisodes",maxEpisodes, ...
    "MaxStepsPerEpisode",maxSteps, ...
    "ScoreAveragingWindowLength",10, ...
    "StopTrainingCriteria","AverageReward", ...
    "StopTrainingValue",50000, ...
    "Verbose", true, ...
    "Plots","training-progress");%,...
% 'SaveAgentCriteria','AverageReward',...
% 'SaveAgentValue',170,...%hau aldatu egin beharko da ikusten denaren arabera
% 'SaveAgentDirectory', pwd + "\run\Agents");
trainingStats = train(agent,env,trainOpts);

%condiciones de parada que se cosidere buen agente
if mean(trainingStats.EpisodeQ0(end-10:end))>=trainingStats.AverageReward(end)
    isOk=1;
else
    %neural network is incorrect
    isOk=0;
end
end

```

myStepFunction.m

```

function [NextObs,Reward,IsDone,LoggedSignals] = myStepFunction(Action,LoggedSignals)

aldagaiak=@hasieradef;
[mapa,~,~,finX,finY] = aldagaiak();

%se extrae la accion correspondiente
ThitaVel=Action;
% Unpack the state vector from the logged signals.
State = LoggedSignals.State;%x, y, xh, yh, xo, yo
IbilParam.dt=0.1;
IbilParam.Vmax=1;
%helburua
Xhelmuga=[finX;finY];

%Aquí se define el modelo de vehiculo que tenemos
%siguiente obserbacion
x_next=State(1)+IbilParam.Vmax*cos(ThitaVel)*IbilParam.dt;
y_next=State(2)+IbilParam.Vmax*sin(ThitaVel)*IbilParam.dt;

% SE ACTUALIZA LA OBSERVACION
LoggedSignals.State(1) = x_next;
LoggedSignals.State(2) = y_next;
% Return initial environment state variables as logged signals.

%EVALUAR DONDE VOY A ESTAR
DistMap=zeros*length(mapa);
for k=1:length(mapa)
    DistMap(k)=norm(mapa(:,k)-[x_next;y_next]);
end
[gertukoena,rfmin]=min(DistMap);
LoggedSignals.State(3)=finX-x_next;%xh
LoggedSignals.State(4)=finY-y_next;%yh
LoggedSignals.State(5)=(mapa(1,rfmin))-x_next;%xo
LoggedSignals.State(6)=(mapa(2,rfmin))-y_next;%yo

%-----
%IS DONE
IsDone=0;
if x_next<=0 || x_next>=14 %dependen de la zona del mapa
    IsDone=1;
end
if y_next<=0 || y_next>=14
    IsDone=1;
end
if x_next<=10 && y_next>=10
    IsDone=1;
end
DistmaxGertuEgoteko=0.51;
if gertukoena<DistmaxGertuEgoteko
    IsDone=1;
end

fin=norm(Xhelmuga-([x_next;y_next]));
%-----
%REWARD
Reward=(((1/fin)^2)*20)+((gertukoena^2)*100);
NextObs = [LoggedSignals.State];
end

```

myResetFunction.m

```

%reset function ddpq+neat
function [InitialObservation, LoggedSignal] = myResetFunction()

aldagaiak=@hasieradef;
[mapa,initX,initY,finX,finY] = aldagaiak();

% Reset function to place custom cart-pole environment into a random
hasiera=[initX;initY];

```

```

% Return initial environment state variables as logged signals.
DistMap=zeros*length(mapa);
for k=1:length(mapa)
    DistMap(k)=norm(mapa(:,k)-hasiera);
end
[gertukoena,rfmin]=min(DistMap);
xh=finX-initX;
yh=finY-initY;
xo=(mapa(1,rfmin))-initX;
yo=(mapa(2,rfmin))-initY;

LoggedSignal.State = [hasiera;xh;yh;xo;yo];

InitialObservation = [LoggedSignal.State];
end

```

hasieradef.m

```

function [mapa,initX,initY,finX,finY]=hasieradef()

%Mapa eraikitzen da ingurunearen limiteak eta oztopoak definituz
%limiteak
ParedInf1=(0:0.5:10);
ParedInf1(2,:)=zeros(1,length(ParedInf1));
ParedInf2=(10.5:0.5:14);
ParedInf2(2,:)=ones(1,length(ParedInf2))*10;
ParedInf=[ParedInf1,ParedInf2];
ParedSup=(0:0.5:14);
ParedSup(2,:)=ones(1,length(ParedSup))*14;
ParedIzq(2,:)=(0:0.5:14);
ParedIzq(1,:)=zeros(1,length(ParedIzq));
ParedDer1(2,:)=(0:0.5:10);
ParedDer1(1,:)=ones(1,length(ParedDer1))*10;
ParedDer2(2,:)=(10.5:0.5:14);
ParedDer2(1,:)=ones(1,length(ParedDer2))*14;
ParedDer=[ParedDer1,ParedDer2];
Paredes=[ParedInf,ParedSup,ParedIzq,ParedDer];
mapa=[Paredes];

%aldagaiak ere definitzen dira
%Posicion inicial
initX=randi(10)-1; %entrenamendurako ikusi da hobe zela aleatorioki hasiera leku baten
initY=randi(14)-1;
mapax=mapa(1,:);%x
mapay=mapa(2,:);%y
coinciden=1;
while coinciden==1
linearIndices=find(mapax==initX);
    if isempty(linearIndices)
        coinciden=0;
    else
        for i=1:length(linearIndices)
            value=mapay(linearIndices(i));
            if value==initY
                coinciden=1;
                initX=randi(10)-1;
                initY=randi(14)-1;
            else
                coinciden=0;
            end
        end
    end
end
end
%posicion objetivo
finX=12;
finY=12;
end

```

Algoritmo genetiko bidezko optimizazioa

Sarearen eboluzioa egin beharrea akzioa kalkulatzeko doitze bat ere proposatu da eta kodigo honen bidez egin da. Aipatu beharra dago, kodigo honekin batera *hasieradef3.m* ere erabili dela.

BasqueNavigationFuntz.m

```

clc
close all
clear all

[mapa,allpos,finX,finY]=hasieradef3();
rng default % For reproducibility
options = optimoptions('ga','ConstraintTolerance',1e-6,'PlotFcn', @gaplotbestf);
alpha1=20;
alpha2=100;
PlotBaiEz=0;

Kopt =
ga(@(K)BasqueNavFuntzioa(K,finX,finY,mapa,allpos,alpha1,alpha2,PlotBaiEz),1,[],[],[],[],0.01,0.1,[],options);
PlotBaiEz=1;
[Gken]=BasqueNavFuntzioa(Kopt,finX,finY,mapa,allpos,alpha1,alpha2,PlotBaiEz)

%% Basque Reinforcement Learning
% Execution Loop:
function [Gken]=BasqueNavFuntzioa(K,finX,finY,mapa,allpos,alpha1,alpha2,PlotBaiEz)
    if PlotBaiEz==1
        figure
    end
    % for posizioak=1:length(allpos)
    % State=allpos(:,posizioak);
    State=[1;1];
    Niter=1000;
    V=1;
    DeltaT=0.5;
    rvec=zeros(1,Niter);
    for t=1:Niter
        %Lerrokadura
        Helmuga=[finX;finY];
        XrelH0=Helmuga-State(:,end);
        distvec=zeros(1,size(mapa,2));
        for i=1:length(distvec)
            distvec(i)=norm(mapa(:,i)-State(:,end));
        end
        [distmin,refmin]=min(distvec);
        XrelH0=mapa(:,refmin)-State(:,end);
        thitaH=angle(XrelH0(1)+sqrt(-1)*XrelH0(2));
        M10=[cos(thitaH), -sin(thitaH);sin(thitaH) cos(thitaH)];
        M01=M10';
        XrelH1=M01*XrelH0;
        XrelO1=M01*XrelO0;
        if XrelO1(1)-XrelH1(1)>=0 || XrelO1(1)<=0
            u=atan2(XrelH0(2),XrelH0(1));
        else
            u=atan2(XrelH0(2),XrelH0(1))+atan(K*(XrelO1(1)-XrelH1(1))*XrelO1(1)/XrelO1(2));
        end
        if norm(XrelH0)<0.1
            V=0;
            break
        else
            V=1;
        end
        State_tplus1=fstep(State(:,end),u,V,DeltaT);
        State=[State,State_tplus1];
        rvec(t)=alpha1*(1/norm(XrelH1)^2)+alpha2*(norm(XrelO1)^2);
        if PlotBaiEz==1
            plot(mapa(1,:),mapa(2:,:), 'xb')
            grid on
            hold on
            hold on
            plot(finX,finY, '-p', 'MarkerFaceColor', 'red', 'MarkerSize', 15)
            plot(State(1,end),State(2,end), 'gx');
        end
    end
end

```



```

        quiver(State(1,end),State(2,end),cos(u),sin(u))
        plot(mapa(1,refmin),mapa(2,refmin),'xr')
    end
end
% end
%Kalkulatzea r eta G
G=sum(rvec);
Gken=-G;
end
%% Code:

function [State_tplus1]=fstep(State_t,u_t,V,DeltaT)
    State_tplus1=State_t+DeltaT*V*[cos(u_t);sin(u_t)];
end

```

3.2.SLAM bisuala

Ondoren agertzen diren kodigoan aldibereko lokalizazioa eta mapatzea egiteko erabilitakoak izan dira. Indibidualak dira eta banan-banan exekutatu behar dira.

guardarimagenes.m

```

clc
close all
clear all

%% Loop
run=true;
count=0;
mostrar=true;
%Directorio donde se quieren guardar
folder='C:\Users\CarpetaDestinoImagenesCalibracion';

divice_number=2; %Identificar la web cam, puede haber varias
cam = webcam(divice_number);
cam.Resolution='1920x1080';
cam.FocusMode='Auto';

% Imagen de prueba
Im_original=snapshot(cam);
Im_original_size=size(Im_original,(1:2));

while(run)
    Im_CalibApil=snapshot(cam);
    imshow(Im_CalibApil)
    if mostrar
        Tablero=Im_CalibApil;
        Tablero_name=sprintf('Tablero_Image_%i.jpg',count);%nombre de imagenes
        %Tablero_path=fullfile(folder,Tablero_name);
        imwrite(Tablero,Tablero_name);
    end
    count=count+1;
    pause(4)
end
end

```

CalibracionCamaraConAprilTag.mlx

Calibrar camara utilizando AprilTags

Step 1: Generate the calibration pattern

Download and prepare tag images

```

downloadURL = 'https://github.com/AprilRobotics/apriltag-imgs/archive/master.zip';
dataFolder = fullfile(tempdir, 'apriltag-imgs', filesep);
options = weboptions('Timeout', Inf);
zipFileName = [dataFolder, 'apriltag-imgs-master.zip'];

```

```

folderExists = exist(dataFolder, 'dir');

% Create a folder in a temporary directory to save the downloaded file
if ~folderExists
    mkdir(dataFolder);
    disp('Downloading apriltag-imgs-master.zip (60.1 MB)...')
    websave(zipFileName, downloadURL, options);

    % Extract contents of the downloaded file
    disp('Extracting apriltag-imgs-master.zip...')
    unzip(zipFileName, dataFolder);
end

% Set the properties of the calibration pattern.
tagArrangement = [5,8];
tagFamily = 'tag36h11';

% Generate the calibration pattern using AprilTags.
tagImageFolder = [dataFolder 'apriltag-imgs-master/' tagFamily];
imdsTags = imageDatastore(tagImageFolder);
calibPattern = helperGenerateAprilTagPattern(imdsTags, tagArrangement, tagFamily);

% Read and localize the tags in the calibration pattern.
[tagIds, tagLocs] = readAprilTag(calibPattern, tagFamily);

% Sort the tags based on their ID values.
[~, sortIdx] = sort(tagIds);
tagLocs = tagLocs(:, :, sortIdx);

% Reshape the tag corner locations into an M-by-2 array.
tagLocs = reshape(permute(tagLocs, [1,3,2]), [], 2);

% Convert the AprilTag corner locations to checkerboard corner locations.
checkerIdx = helperAprilTagToCheckerLocations(tagArrangement);
imagePoints = tagLocs(checkerIdx(:), :);

% Display corner locations.
figure; imshow(calibPattern); hold on
plot(imagePoints(:,1), imagePoints(:,2), 'ro-', 'MarkerSize', 15)

Step 2: Detect and localize the AprilTags
% Create an imageDatastore object to store the captured images.
imdsCalib = imageDatastore("aprilTagCalibImages/");

% Detect the calibration pattern from the images.
[imagePoints, boardSize] = helperDetectAprilTagCorners(imdsCalib, tagArrangement, tagFamily);

Step 3: Generate world points for the calibration pattern
% Generate world point coordinates for the pattern.
tagSize = 19; % in millimeters
worldPoints = generateCheckerboardPoints(boardSize, tagSize);

Step 4: Estimate camera parameters
% Determine the size of the images.
I = readimage(imdsCalib, 1);
imageSize = [size(I,1), size(I,2)];

% Estimate the camera parameters.
params = estimateCameraParameters(imagePoints, worldPoints, 'ImageSize', imageSize);

% Display the reprojection errors.
figure
showReprojectionErrors(params)

% Display the extrinsics.
figure
showExtrinsics(params)

% Read a calibration image.
I = readimage(imdsCalib, 5);

% Insert markers for the detected and reprojected points.
I = insertMarker(I, imagePoints(:, :, 5), 'o', 'Color', 'g', 'Size', 5);

```

```

I = insertMarker(I, params.ReprojectedPoints(:, :, 5), 'x', 'Color', 'r', 'Size', 5);

% Display the image.
figure
imshow(I)

Supporting functions
helperGenerateAprilTagPattern generates an AprilTag based calibration pattern.
function calibPattern = helperGenerateAprilTagPattern(imdsTags, tagArrangement, tagFamily)

numTags = tagArrangement(1)*tagArrangement(2);
tagIds = zeros(1,numTags);

% Read the first image.
I = readimage(imdsTags, 3);
Igray = im2gray(I);

% Scale up the thumbnail tag image.
Ires = imresize(Igray, 15, 'nearest');

% Detect the tag ID and location (in image coordinates).
[tagIds(1), tagLoc] = readAprilTag(Ires, tagFamily);

% Pad image with white boundaries (ensures the tags replace the black
% portions of the checkerboard).
tagSize = round(max(tagLoc(:,2)) - min(tagLoc(:,2)));
padSize = round(tagSize/2 - (size(Ires,2) - tagSize)/2);
Ires = padarray(Ires, [padSize,padSize], 255);

% Initialize tagImages array to hold the scaled tags.
tagImages = zeros(size(Ires,1), size(Ires,2), numTags);
tagImages(:, :, 1) = Ires;

for idx = 2:numTags

    I = readimage(imdsTags, idx + 2);
    Igray = im2gray(I);
    Ires = imresize(Igray, 15, 'nearest');
    Ires = padarray(Ires, [padSize,padSize], 255);

    tagIds(idx) = readAprilTag(Ires, tagFamily);

    % Store the tag images.
    tagImages(:, :, idx) = Ires;

end

% Sort the tag images based on their IDs.
[~, sortIdx] = sort(tagIds);
tagImages = tagImages(:, :, sortIdx);

% Reshape the tag images to ensure that they appear in column-major order
% (montage function places image in row-major order).
columnMajIdx = reshape(1:numTags, tagArrangement)';
tagImages = tagImages(:, :, columnMajIdx(:));

% Create the pattern using 'montage'.
imgData = montage(tagImages, 'Size', tagArrangement);
calibPattern = imgData.CData;

end
helperDetectAprilTagCorners detects AprilTag calibration pattern in images.
function [imagePoints, boardSize, imagesUsed] = helperDetectAprilTagCorners(imdsCalib, tagArrangement,
tagFamily)

% Get the pattern size from tagArrangement.
boardSize = tagArrangement*2 + 1;

% Initialize number of images and tags.
numImages = length(imdsCalib.Files);
numTags = tagArrangement(1)*tagArrangement(2);

% Initialize number of corners in AprilTag pattern.
imagePoints = zeros(numTags*4,2,numImages);

```

```

imagesUsed = zeros(1, numImages);

% Get checkerboard corner indices from AprilTag corners.
checkerIdx = helperAprilTagToCheckerLocations(tagArrangement);

for idx = 1:numImages

    % Read and detect AprilTags in image.
    I = readimage(imdsCalib, idx);
    [tagIds, tagLocs] = readAprilTag(I, tagFamily);

    % Accept images if all tags are detected.
    if numel(tagIds) == numTags
        % Sort detected tags using ID values.
        [~, sortIdx] = sort(tagIds);
        tagLocs = tagLocs(:, :, sortIdx);

        % Reshape tag corner locations into a M-by-2 array.
        tagLocs = reshape(permute(tagLocs, [1,3,2]), [], 2);

        % Populate imagePoints using checkerboard corner indices.
        imagePoints(:, :, idx) = tagLocs(checkerIdx(:, :), :);
        imagesUsed(idx) = true;
    else
        imagePoints(:, :, idx) = [];
    end
end

end

end
helperAprilTagToCheckerLocations converts AprilTag corners to checkerboard corners.
function checkerIdx = helperAprilTagToCheckerLocations(tagArrangement)

numTagRows = tagArrangement(1);
numTagCols = tagArrangement(2);
numTags = numTagRows * numTagCols;

% Row index offsets.
rowIdxOffset = [0:numTagRows - 1; 0:numTagRows - 1];

% Row indices for first and second columns in board.
col1Idx = repmat([4 1]', numTagRows, 1);
col2Idx = repmat([3 2]', numTagRows, 1);
col1Idx = col1Idx + rowIdxOffset(:)*4;
col2Idx = col2Idx + rowIdxOffset(:)*4;

% Column index offsets
colIdxOffset = 0:4*numTagRows:numTags*4 - 1;

% Implicit expansion to get all indices in order.
checkerIdx = [col1Idx;col2Idx] + colIdxOffset;

end

```

V3AprilTagDeteccionMovimiento.mlx

Codigo para detectar posicion y medidas de AprilTag MOVIENDO LA CAMARA

RASTREO DE APRILTAGS

Codigo para conocer en que posicion esta el AprilTag en referencia al la camara, esto es, centro de la imagen.

Tambien se dan las medidas en la imagen del Apriltag

POSICION Y MEDIDAS REALES DE LOS APRILTAGS

Medidas cuadrado= 9,9 x 9,9 cm

```
clear all
```

```
close all
```

```
clc
```

Configuracion de camara y definicion de aprilTag

```
device_number=2;
```

```
cam=webcam(device_number);
```

```
cam.Resolution='1920x1080';
```

```
cam.FocusMode='Auto';
```

```

%Parametros que se han calculado de la calibracion
load CamParamsAprilTagCalibration.mat %se ha calibrado en milimetros

%Datos de apriltags
tagFamily='tag36h11';
tagSize=99; %en milimetros, por la calibracion

Image_center=[1920/2,1080/2]; %cam resolution entre 2

Definicion de variables generales
Data_Table=table('Size',[1 13], 'VariableTypes',{ 'double','cell','cell','cell','double',
'double','double','double','double','double','double','double'},...
'VariableNames',{ 'ID','Loc','Pose.Rotation','Pose.Translation','Thita','Betha','x','y','z','distApril-
Cam','Xabs','Zabs','IDangle'});%tabla almacenamiento datos april
worldPoints = [0 0 0; tagSize/2 0 0; 0 tagSize/2 0; 0 0 tagSize/2];
%Los ejes del apriltag salen en el centro de la imagen hasta borde del april tag
TablePos=1;

I = snapshot(cam);
%PRIMERA DETECCION --> DEFINICION DE MEDIDAS ABSOLUTAS
[id,loc,pose,detectedFamily] = readAprilTag(I,tagFamily,params.Intrinsics,tagSize);
distAplilCam=zeros(1,length(id));

for i=1:length(id)
imagePoints = worldToImage(params.Intrinsics,pose(i).Rotation,pose(i).Translation,worldPoints);
I = insertShape(I,"Line",[imagePoints(1,:) imagePoints(2,:); ...
imagePoints(1,:) imagePoints(3,:); imagePoints(1,:) imagePoints(4,:)], ...
"Color",["red","green","blue"],"LineWidth",7);
Data_Table(TablePos,"ID")={id(i)};
Data_Table(TablePos,"Loc")={loc(:, :, i)};
Data_Table(TablePos,"Pose.Rotation")={pose(i).Rotation};
Data_Table(TablePos,"Pose.Translation")={{pose(i).Translation}};

% Marcas en las esquinas
markerRadius = 8;
numCorners = size(loc,1);
markerPosition = [loc(:, :, i), repmat(markerRadius,numCorners,1)];
I = insertShape(I,"FilledCircle",markerPosition,Color="red",Opacity=1);
hold on

%recuadro de los april
poligon=[markerPosition(1,1:2) markerPosition(2,1:2); markerPosition(2,1:2) markerPosition(3,1:2);...
markerPosition(3,1:2) markerPosition(4,1:2);markerPosition(4,1:2) markerPosition(1,1:2)];
I = insertShape(I,"Line",poligon,"Color","magenta","LineWidth",5);
I = insertText(I,loc(1, :, i),id(i), "BoxOpacity",1, "FontSize",25);
hold on

%datos de april
%los april van a estar pegados rectos en la pared
eulers=rad2deg(rotm2eul(pose(i).Rotation));
thita=eulers(2);
betha=eulers(3);
x=pose(i).Translation(1);%desplazamiento
y=pose(i).Translation(2);%altura
z=pose(i).Translation(3);%profundidad
Data_Table(TablePos,"Thita")={thita};
Data_Table(TablePos,"Betha")={betha};
Data_Table(TablePos,"x")={x};
Data_Table(TablePos,"y")={y};
Data_Table(TablePos,"z")={z};

%Medidas de profundidad real de camara a apriltag
distAplilCam=sqrt(x^2+y^2+z^2); %en mm
distAplilCam=distAplilCam/10; %en cm
Data_Table(TablePos,"distApril-Cam")={distAplilCam};

if mod(id(i),2)==0
x_abs=(z*sin(deg2rad(thita))+x*cos(deg2rad(thita)));%le he quitado el abs
z_abs=z*cos(deg2rad(thita))-x*sin(deg2rad(thita));
Data_Table(TablePos,"IDangle")={0};
else
x_abs=-(z*sin(deg2rad(thita-90))+x*cos(deg2rad(thita-90)));%le he quitado el abs
z_abs=-(z*cos(deg2rad(thita-90))-x*sin(deg2rad(thita-90)));

```

```

Data_Table(TablePos,"IDangle")={1};
if thita>=0
    x_abs=-x_abs;
    z_abs=-z_abs;
end
end

Data_Table(TablePos,"Xabs")={x_abs};
Data_Table(TablePos,"Zabs")={z_abs};
TablePos=TablePos+1;
end
imshow(I)

Bucle de deteccion de apriltags RELATIVOS
I = snapshot(cam);
%BUCLE DE DETECCION --> DEFINICION DE MEDIDAS EN RELATIVO --> PASAR A ABSOLUTO
[id,loc,pose,detectedFamily] = readAprilTag(I,tagFamily,params.Intrinsics,tagSize);
if ~isempty(id)

    %comprobar si hay nuevo april
    a=intersect(Data_Table(:, "ID").Variables,id);
    if isempty(a)%no se detecta ningun april nuevo
        fprintf('No hay un apriltag ya escaneado en la imagen, gira o retrocede')
        imshow(I)
    elseif length(a)==length(id)%los april detectados ya se conocian
        fprintf('Los aprilTag escaneados son conocidos, gira o avanza')
        imshow(I)
    elseif length(a)<length(id) %&& ~isempty(a)%hay al menos un nuevo april y al menos uno conocido

        %posicion relativa del april conocido
        for g=1:length(id)
            if id(g)==a(1)
                imagePoints = worldToImage(params.Intrinsics,pose(g).Rotation,
                    pose(g).Translation,worldPoints);
                %datos relativos de april conocido
                eulers=rad2deg(rotm2eul(pose(g).Rotation));
                x=pose(g).Translation(1);%desplazamiento
                z=pose(g).Translation(3);%profundidad
                thita=eulers(2);
                if mod(id(g),2)==0
                    x_relOLD=(z*sin(deg2rad(thita))+x*cos(deg2rad(thita)));%le he quitado el abs
                    z_relOLD=(z*cos(deg2rad(thita))-x*sin(deg2rad(thita)));
                    Data_Table(TablePos, "IDangle")={0};
                else
                    x_relOLD=-(z*sin(deg2rad(thita-90))+x*cos(deg2rad(thita-90)));%le he quitado el abs
                    z_relOLD=-(z*cos(deg2rad(thita-90))-x*sin(deg2rad(thita-90)));
                    Data_Table(TablePos, "IDangle")={1};
                    if thita>=0
                        x_relOLD=-x_relOLD;
                        z_relOLD=-z_relOLD;
                    end
                end
                idOLD=Data_Table.ID==id(g);
                DifX_RelAbs=(Data_Table.Xabs(idOLD))-(x_relOLD);
                DifZ_RelAbs=(Data_Table.Zabs(idOLD))-(z_relOLD);
            end
        end

        %Calculo las posiciones absolutas de los nuevos apriltags
        for i=1:length(id)
            imagePoints = worldToImage(params.Intrinsics,pose(i).Rotation,
                pose(i).Translation,worldPoints);
            I = insertShape(I,"Line",[imagePoints(1,:) imagePoints(2,:); ...
                imagePoints(1,:) imagePoints(3,:); imagePoints(1,:) imagePoints(4,:)], ...
                "Color",["red","green","blue"],"LineWidth",7);

            % Marcas en las esquinas
            markerRadius = 8;
            numCorners = size(loc,1);
            markerPosition = [loc(:, :, i), repmat(markerRadius,numCorners,1)];
            I = insertShape(I,"FilledCircle",markerPosition,Color="red",Opacity=1);
            hold on
        end

        %recuadro de los april

```

```

poligon=[markerPosition(1,1:2) markerPosition(2,1:2);
          markerPosition(2,1:2) markerPosition(3,1:2);...
          markerPosition(3,1:2) markerPosition(4,1:2);
          markerPosition(4,1:2) markerPosition(1,1:2)];
I = insertShape(I,"Line",poligon,"Color","magenta","LineWidth",5);
I = insertText(I,loc(1,:,i),id(i),"BoxOpacity",1,"FontSize",25);
hold on

%datos de april
eulers=rad2deg(rotm2eul(pose(i).Rotation));
thita=eulers(2);
betha=eulers(3);
x=pose(i).Translation(1);%desplazamiento
y=pose(i).Translation(2);%altura
z=pose(i).Translation(3);%profundidad

%Medidas de profundidad real de camara a apriltag
distAplilCam=sqrt(x^2+y^2+z^2); %en mm
distAplilCam=distAplilCam/10; %en cm
if mod(id(i),2)==0
    x_relNew=(z*sin(deg2rad(thita))+x*cos(deg2rad(thita)));%le he quitado el abs
    z_relNew=z*cos(deg2rad(thita))-x*sin(deg2rad(thita));
    idangle=0;
else
    x_relNew=-(z*sin(deg2rad(thita-90))+x*cos(deg2rad(thita-90)));%le he quitado el abs
    z_relNew=-(z*cos(deg2rad(thita-90))-x*sin(deg2rad(thita-90)));
    idangle=1;
    if thita>=0
        x_relNew=-x_relNew;
        z_relNew=-z_relNew;
    end
end

if ~(any(a(:)==id(i)))
%Conversion de medidas relativas a absolutas
x_abs=x_relNew+DifX_RelAbs;
z_abs=z_relNew+DifZ_RelAbs;

Data_Table(TablePos,"ID")={id(i)};
Data_Table(TablePos,"Loc")={loc(:, :, i)};
Data_Table(TablePos,"Pose.Rotation")={pose(i).Rotation};
Data_Table(TablePos,"Pose.Translation")={{pose(i).Translation}};
Data_Table(TablePos,"Thita")={thita};
Data_Table(TablePos,"Betha")={betha};
Data_Table(TablePos,"x")={x};
Data_Table(TablePos,"y")={y};
Data_Table(TablePos,"z")={z};
Data_Table(TablePos,"distApril-Cam")={distAplilCam};
Data_Table(TablePos,"Xabs")={x_abs};
Data_Table(TablePos,"Zabs")={z_abs};
Data_Table(TablePos,"IDangle")={idangle};
TablePos=TablePos+1;
end
imshow(I)
end
imshow(I)
end
else
fprint('No se estan detectando apriltags')
imshow(I)
end

COLOCACION DE APRILS EN ESPACIO
%Plano Camara con resolusion de camara
% x distancia lateral, y altura, z profundidad
PointA=[-1,-1,0];
PointB=[1,-1,0];
PointC=[1,1,0];
PointD=[-1,1,0];
PointsCam=[PointA' PointB' PointC' PointD'];
figure
fill3(PointsCam(1,:),PointsCam(3,:),PointsCam(2,:), 'r')
grid on
alpha(1)

```

```

text(PointB(1),PointB(2),PointB(3),'\leftarrow cam')
hold on

xlabel('xabs [cm]')
ylabel('zabs [cm]')

[x,y]=meshgrid(0:1:150);
Todos=size(Data_Table);

plano=table('Size',[1 5],'VariableTypes',{'cell','double','double','double','double'},'VariableNames',
    {'puntosdelplano','a','b','c','d'});

for i=1:(Todos(1))
    y_altura=300;
    %Plano Aplitag en cm 5cm es medio aprilTag
    if Data_Table.IDangle(i)==0
        c=-5*sin(deg2rad(0));
        d=5*cos(deg2rad(0));
    elseif Data_Table.IDangle(i)==1
        c=-5*sin(deg2rad(90));
        d=5*cos(deg2rad(90));
    end
    %este es giro thita
    PointA=[-d,-c,-5];
    PointB=[d,c,-5];
    PointC=[d,c,5];
    PointD=[-d,-c,5];

    PointsCam=[PointA' PointB' PointC' PointD'];
    PointsCam(1,:)=PointsCam(1,:)+((Data_Table.Xabs(i))/10);%colocamos el aprilTag en su posicion
    PointsCam(2,:)=PointsCam(2,:)+((Data_Table.Zabs(i))/10);
    PointsCam(3,:)=PointsCam(3,:)+(y_altura/10);
    fill3(PointsCam(1,:),PointsCam(2,:),PointsCam(3,),'g')
    alpha(1)
    indice=Data_Table.ID(i);
    text(PointsCam(1,1),PointsCam(2,1),PointsCam(3,1),num2str(indice))
    hold on

end
zlim([0 100])

MAPA
vec_april=[5 0 3 2 7 6 10 1 4];%orden de los aprils
%figure
grid on
altura=100;
%CALCULO DE PLANOS
for i=1:(length(vec_april))
    april=vec_april(i);
    [num,indx]=intersect(Data_Table(:, "ID").Variables,april);

    if Data_Table.IDangle(indx)==0 && i~=1 && i~=length(vec_april)
        aprilnext=vec_april(i+1);
        aprilant=vec_april(i-1);
        [numant,indxant]=intersect(Data_Table(:, "ID").Variables,aprilant);
        [numnext,indxnext]=intersect(Data_Table(:, "ID").Variables,aprilnext);
        z=(Data_Table.Zabs(indx));
        xant=(Data_Table.Xabs(indxant));
        xnext=(Data_Table.Xabs(indxnext));
        PointA=[xant,z,0];
        PointB=[xnext,z,0];
        PointC=[xnext,z,altura];
        PointD=[xant,z,altura];
    elseif Data_Table.IDangle(indx)==1 && i~=1 && i~=length(vec_april)
        aprilnext=vec_april(i+1);
        aprilant=vec_april(i-1);
        [numant,indxant]=intersect(Data_Table(:, "ID").Variables,aprilant);
        [numnext,indxnext]=intersect(Data_Table(:, "ID").Variables,aprilnext);
        x=(Data_Table.Xabs(indx));
        zant=(Data_Table.Zabs(indxant));
        znext=(Data_Table.Zabs(indxnext));
        PointA=[x,znext,0];
        PointB=[x,zant,0];
        PointC=[x,zant,altura];
    end
end

```



```

    PointD=[x,znext,altura];
end
%primer april
if i==1
    aprilnext=vec_april(i+1);
    [numnext,indxnext]=intersect(Data_Table(:, "ID").Variables,aprilnext);
    x=(Data_Table.Xabs(indx));
    z=(Data_Table.Zabs(indxnext));
    PointA=[x,z,0];
    PointB=[x,0,0];
    PointC=[x,0,altura];
    PointD=[x,z,altura];
end
%ultimo april
if i==length(vec_april)
    aprilant=vec_april(i-1);
    [numant,indxant]=intersect(Data_Table(:, "ID").Variables,aprilant);
    x=(Data_Table.Xabs(indxant));
    z=(Data_Table.Zabs(indx));
    fin=-1730;%no tenemos otro april en vertical para cerrar en este caso
    PointA=[fin,z,0];
    PointB=[x,z,0];
    PointC=[x,z,altura];
    PointD=[fin,z,altura];
end

PointsCam=[PointA' PointB' PointC' PointD'];
PointsCam(1,:)=PointsCam(1,+)/10;%colocamos el apriltag en su posicion
PointsCam(2,:)=PointsCam(2,+)/10;

fill3(PointsCam(1,:),PointsCam(2,:),PointsCam(3,:), 'b')
alpha(0.3)
hold on
end
zlim([0 100])

```

3.3.Oztopoen distantzien kalkulua

Hurrengo kodigoak oztopoen distantziak estimatzeko lanetan aplikatutakoak izan dira. Aurreko atalean azalduta dago hauen erabilera ordena eta ataza bakoitzerako beharrezkoak direnak.

suavizado.m

```

clc
clear all
close all

dirsuavizados="C:\Users\CarpetaDestinoImagenesSuavizadas"; %carpeta destino
qq=dir("C:\Users\CarpetaImagenesDepthKITTI"); %carpeta dataset lidar
nwindow=4;
for i=1:length(qq)
    if length(strfind(qq(i).name, '.png'))>0
        I=imread(strcat("C:\Users\CarpetaImagenesDepthKITTI", "\", qq(i).name));
        Inew=I;
        [filas, columnas]=size(I);
        for k=nwindow+1:filas-nwindow
            for j=nwindow+1:columnas-nwindow
                window=I(k-nwindow:k+nwindow, j-nwindow:j+nwindow);
                Inew(k, j)=max(max(window));
            end
        end
        subplot(2,1,1)
        mesh(I)
        subplot(2,1,2)
        mesh(Inew)
    end
end

```

```

        Filename=[qq(i).name];
        Destination=fullfile(dirsuavizados,Filename);
        imwrite(Inew,Destination);
    end
end

```

recorteDataAugmentation.m

```

clc
close all
clear all

%% reducir tamaño de imagenes 352*1216 a 240*300 para coger la zona central
N = 999 ; %cantidad de imagenes
j=3000; %numero de imagen para guardar
br=0.2; %brillo
co=0.5; %contraste
dir=" C:\Users\CarpetaImagenesKITTI\";%carpeta dataset KITTI
% loop for each file
for i = 0:N
    if length(num2str(i))==1
        name=append("00000000",num2str(i),".png");
    elseif length(num2str(i))==2
        name=append("0000000",num2str(i),".png");
    else
        name=append("000000",num2str(i),".png");
    end

    Im = imread(name);
    I1 = imcrop(Im,[458 113 319 352]);
    %I1=flip(I1,2);
    %I1 = imadjust(I1, [br, 1.0], [0.0, co]);

    Filename=[append("image_",num2str(j),".png")];
    Destination=fullfile(dir,Filename);
    imwrite(I1,Destination);
    j=j+1;
end

```

nameAdapt.m

```

clear all
close all
clc

j=0;
qq=dir("C:\Users\CarpetaImagenesDataAugmentation");%carpeta dataset final
dirimages="C:\Users\CarpetaDestinoImagenesNombreCorrecto";
for i=1:length(qq)
    name=append("image_",num2str(j),".png");
    Im = imread(name);
    if j>=0 && j<=9
        Filename=[append("image_00000",num2str(j),".png")];
    elseif j>=10 && j<=99
        Filename=[append("image_0000",num2str(j),".png")];
    elseif j>=100 && j<=999
        Filename=[append("image_000",num2str(j),".png")];
    elseif j>=1000 && j<=9999
        Filename=[append("image_00",num2str(j),".png")];
    end

    Destination=fullfile(dirimages,Filename);
    imwrite(Im,Destination);
    j=j+1;
end

```

GureSareGureGeruzarekin.m

```

clc
close all
clear all

%% KITTI DATU BASEAK EMANDAKO FUNTZIOA NEURRIAK LORTZEKO
%entrenamendurako distantzia neurriak
file_path_train=dir("C:\Users\EntrenamenduLidarIrudienKarpeta\*.png");
N=length(file_path_train);
vector_train=zeros(240,320,N);%irudiaren tamaina

for i=1:N
    currentfilename=append(file_path_train(i).folder, "\", file_path_train(i).name);
    depthData = depth_read(currentfilename);
    vector_train(:, :, i)=depthData;
    disp(i)
end

%balidaziorako distantzia neurriak
file_path_val=dir("C:\Users\BalidazioLidarIrudienKarpeta\*.png");
J=length(file_path_val);
vector_val=zeros(240,320,J);

for k=1:J
    currentfilename=append(file_path_val(k).folder, "\", file_path_val(k).name);
    depthData = depth_read(currentfilename);
    vector_val(:, :, k)=depthData;
    disp(k)
end

%% KITTI DATASET-eko irudiak
dataDir = fullfile("C:\Users\EntrenamenduRGBIrudienKarpeta\*.png");
EntrenamenduIrudiak = imageDatastore(fullfile(dataDir));
arrds = arrayDatastore(vector_train,"IterationDimension",3,"OutputType","cell","ReadSize",1);
dsTrain=combine(EntrenamenduIrudiak, arrds);

dataDir2 = fullfile("C:\Users\BalidazioRGBIrudienKarpeta\*.png");
BalidazioIrudiak = imageDatastore(fullfile(dataDir2));
arrds = arrayDatastore(vector_val,"IterationDimension",3,"OutputType","cell","ReadSize",1);
dsVal=combine(BalidazioIrudiak, arrds);

%%
%Gure geruza sortu behar dugu:
GureGeruza=functionLayer(@(X) GureGeruzakoFcn(X),Formattable=true,Description="OurLayer");

layers = [
    imageInputLayer([240 320 3], 'Normalization', 'zerocenter', 'Mean', 0.5)

    convolution2dLayer([3,3],8, 'Padding', 'same')
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(2, 'Stride', 2)

    convolution2dLayer(3,16, 'Padding', 'same')
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(2, 'Stride', 2)

    convolution2dLayer(3,32, 'Padding', 'same')
    batchNormalizationLayer
    reluLayer

    %lehenengo modif
    % transposedConv2dLayer(3,32, 'Stride', 2);
    % batchNormalizationLayer
    % reluLayer
    %bigarren modif
    % convolution2dLayer(3,64, 'Padding', 'same')
    % batchNormalizationLayer
    % reluLayer
    %hirugarren modif

```

```

% fullyConnectedLayer(2000)
% reluLayer
% fullyConnectedLayer(2000)
% reluLayer
%laugarren modif
% fullyConnectedLayer(10000)
% reluLayer
%bostgarren modif
% Azken convolution+batch+relu kendu

fullyConnectedLayer(500)
reluLayer
fullyConnectedLayer(1000)
reluLayer
fullyConnectedLayer(76800)
reluLayer
GureGeruza];

%sare handia sortzen dugu
SareHandia=dlnetwork(layers)
figure;
plot(SareHandia)

X=dllarray(zeros(240,320,3,100),"SSCB");
Irudia=predict(SareHandia,X);
imshow(extractdata(Irudia(:,:,1)))

%% ENTRENAMENDUA
numEpochs = 300;
miniBatchSize = 100;
learningratio=1e-3;
plots = "training-progress";

mbqTrain = minibatchqueue(dsTrain,2,...
    'MiniBatchSize',miniBatchSize,...
    'PartialMiniBatch',"discard",...
    'MiniBatchFormat',{'SSCB','SSB'});

mbqVal = minibatchqueue(dsVal,2,...
    'MiniBatchSize',miniBatchSize,...
    'PartialMiniBatch',"discard",...
    'MiniBatchFormat',{'SSCB','SSB'});

if plots == "training-progress"
    figure;
    lineLossTrain = animatedline('Color','r');
    lineLossVal = animatedline('Color','b','Marker','*');
    xlabel("Iteration")
    ylabel("Loss")
    grid on
end

trailingAvg = [];
trailingAvgSq = [];

iteration = 0;
start = tic;
ValIter=10;
ival=1;

for epoch = 1:numEpochs

    if mod(epoch,ValIter)==ValIter-1
        shuffle(mbqVal);
        GlossVal = [];
        while hasdata(mbqVal)
            [dlXval,dITarjetval] = next(mbqVal);
            [gradientsVal,stateVal,lossVal] = dlfeval(@customLoss, SareHandia, dlXval, dITarjetval);
            GlossVal = [GlossVal;lossVal];
            if plots == "training-progress"
                D = duration(0,0,toc(start),'Format','hh:mm:ss');
                title("Epoch: " + epoch + ", Iteration: " + iteration + ", Elapsed: " + string(D))
                drawnow
            end
        end
    end
end

```

```

end

if plots == "training-progress"
    D = duration(0,0,toc(start), 'Format', 'hh:mm:ss');
    addpoints(lineLossVal,iteration,mean(log10(double(gather(extractdata(GlossVal))))))
    title("Epoch: " + epoch + ", Iteration: " + iteration + ", Elapsed: " + string(D))
    drawnow
end

end

shuffle(mbqTrain);
while hasdata(mbqTrain)
    iteration = iteration + 1;
    [d1X,d1Tarjet] = next(mbqTrain);
    [gradients,state,loss] = dlfeval(@customLoss, SareHandia, d1X, d1Tarjet);

    [SareHandia,trailingAvg,trailingAvgSq] = adamupdate(SareHandia,gradients, ...
        trailingAvg,trailingAvgSq,iteration,learningratio);
    if plots == "training-progress"
        D = duration(0,0,toc(start), 'Format', 'hh:mm:ss');
        addpoints(lineLossTrain,iteration,log10(double(gather(extractdata(loss))))))
        title("Epoch: " + epoch + ", Iteration: " + iteration + ", Elapsed: " + string(D))
        drawnow
    end
end

end

Training = table(epoch,iteration,string(D),loss,lossVal);
fprintf('Training loss:\n');
disp(Training)

reset(mbqVal);
while hasdata(mbqVal)
    [d1Xval,d1Tarjetval] = next(mbqVal);
    d1Xval=double(d1Xval);
    d1YTeo=predict(SareHandia,d1Xval);
    hold on
    grid on
    plot(d1YTeo(:,1),-d1Tarjetval(:,1),'xb')
    pause
    close all
end
end

```

GureGeruzakoFcn.m

```

function [Output] = GureGeruzakoFcn(Features)
    F=d1array(Features);
    [col,fil]=size(F);
    Output1=reshape(F,[240,320,fil]);%irudiak moztuta
    Output=d1array(Output1,"SSB");
end

```

depth_read.m

```

function D = depth_read (filename)
% loads depth map D from png file

I = imread(filename);
D = double(I)/256;
D(I==0) = 100;
end

```

customLoss.m

```
function [gradients,state,loss] = customLoss(SareHandia,dlX,dlTarjet)
    dlX=double(dlX);
    dlY=predict(SareHandia,dlX);
    loss=mse(dlY,dlTarjet);
    gradients = dlgradient(loss,SareHandia.Learnables);
    state=0;
end
```

objectdetection.m

```
%% Object Detection Using Faster R-CNN Deep Learning
clear all
close all
clc
%% data
train=load('bbboxtrain.mat');
bbtrain=train.gTruth.LabelData;
dataDir = fullfile("C:\Users\RGBEntrenamenduIrudienKarpeta\*.png");
imdsTrain = imageDatastore(fullfile(dataDir));
bldsTrain = boxLabelDatastore(bbtrain);

val=load('bbboxval.mat');
bbval=val.gTruth.LabelData;
dataDir2 = fullfile("C:\Users\RGBBalidazioIrudienKarpeta\*.png");
imdsValidation = imageDatastore(fullfile(dataDir2));
bldsValidation = boxLabelDatastore(bbval);

test=load('bbboxtest.mat');
bbtest=test.gTruth.LabelData;
dataDir3 = fullfile("C:\Users\RGBTestIrudienKarpeta\*.png");
imdsTest= imageDatastore(fullfile(dataDir3));
bldsTest = boxLabelDatastore(bbtest);

trainingData = combine(imdsTrain,bldsTrain);
validationData = combine(imdsValidation,bldsValidation);
testData = combine(imdsTest,bldsTest);

data = read(trainingData);
I = data{1};
bbox = data{2};
annotatedImage = insertShape(I,'rectangle',bbox);
annotatedImage = imresize(annotatedImage,2);
figure
imshow(annotatedImage)
%% delete las que no tienen label7
N=length(bldsTrain.LabelData);
trainingIdx=[];
for i= 1:N
    a=bldsTrain.LabelData{i,1};
    if ~isempty(a)
        trainingIdx=[trainingIdx,i];
    end
end
N=length(bldsValidation.LabelData);
valIdx=[];
for i= 1:N
    a=bldsValidation.LabelData{i,1};
    if ~isempty(a)
        valIdx=[valIdx,i];
    end
end
N=length(bldsTest.LabelData);
testIdx=[];
for i= 1:N
    a=bldsTest.LabelData{i,1};
    if ~isempty(a)
        testIdx=[testIdx,i];
    end
end
end
```

```

trainingData=subset(trainingData,trainingIdx);
rng(0);
validationData=subset(validationData,valIdx);
rng(0);
testData=subset(testData,testIdx);
rng(0);

data = read(trainingData);
%% Create Faster R-CNN Detection Network
inputSize = [240 320 3];
preprocessedTrainingData = transform(trainingData, @(data)preprocessData(data,inputSize));

numAnchors = 3;
anchorBoxes = estimateAnchorBoxes(preprocessedTrainingData,numAnchors);

featureExtractionNetwork = resnet50;
featureLayer = 'activation_40_relu';
numClasses = 2;

lgraph = fasterRCNNLayers(inputSize,numClasses,anchorBoxes,featureExtractionNetwork,featureLayer);

%%
trainingData=preprocessedTrainingData;
validationData = transform(validationData,@(data)preprocessData(data,inputSize));
%% Train Faster R-CNN
options = trainingOptions('sgdm',...
    'MaxEpochs',10,...
    'MiniBatchSize',2,...
    'InitialLearnRate',1e-3,...
    'CheckpointPath',tempdir,...
    'ValidationData',validationData);

[detector, info] = trainFasterRCNNObjectDetector(trainingData,lgraph,options, ...
    'NegativeOverlapRange',[0 0.3], ...
    'PositiveOverlapRange',[0.6 1]);

I = imread(imsdTest.Files{3});
I = imresize(I,inputSize(1:2));
[bboxes,scores] = detect(detector,I);

I = insertObjectAnnotation(I,'rectangle',bboxes,scores);
figure
imshow(I)

%% Evaluate Detector Using Test Set

testData = transform(testData,@(data)preprocessData(data,inputSize));
detectionResults = detect(detector,testData,'MinibatchSize',4);
[ap, recall, precision] = evaluateDetectionPrecision(detectionResults,testData);

figure
plot(recall,precision)
xlabel('Recall')
ylabel('Precision')
grid on
title(sprintf('Average Precision = %.2f', ap))

%%
function data = preprocessData(data,targetSize)
% Resize image and bounding boxes to targetSize.
sz = size(data{1},[1 2]);
scale = targetSize(1:2)./sz;
data{1} = imresize(data{1},targetSize(1:2));

% Sanitize box data, if needed.
data{2} = helperSanitizeBoxes(data{2}, sz);

% Resize boxes.
data{2} = bboxresize(data{2},scale);
end

```

helperSanitizeBoxes.m

```

% Copyright 2020-2021 The Mathworks, Inc.

function boxes = helperSanitizeBoxes(boxes, ~)
persistent hasInvalidBoxes
valid = all(boxes > 0, 2);
if any(valid)
    if ~all(valid) && isempty(hasInvalidBoxes)
        % Issue one-time warning about removing invalid boxes.
        hasInvalidBoxes = true;
        warning('Removing ground truth bounding box data with values <= 0.')
    end
    boxes = boxes(valid,:);
end
end

```

depthcalculation.m

```

clc
close all
clear all
% OZTOPOEN DISTANTZIEN ESTIMAZIOA
% test irudiak kargatu
dataDir = fullfile('C:\Users\RGBTestIrudienKarpeta\*.png');
imdsTest= imageDatastore(fullfile(dataDir));
load('sareaentrenamenduosoarekin0311.mat')
load('detectorsareav2.mat')
N=length(imdsTest.Files);

for i=1:N
    im=imread(imdsTest.Files{i});

    %sakontasuna kalkulatzen da
    i1=double(im);
    i1=dlarray(i1,'SSC');
    idepth=predict(SareHandia,i1);

    %oztopoa lokalizatzen da
    [bboxes,scores] = detect(detector,im);

    Nobject=size(bboxes);
    Nobject=Nobject(1);
    for k=1:Nobject
        bboxes2=bboxes(k,:);
        %bbox-ean dauden pixelen %15a kalkulatzen da
        %idepth-eko 15% balio minimoak kalkulatzen dira
        bboxdepth = imcrop(idepth,bboxes2);%bakarrik depth iruko depth zatia
        x = extractdata(bboxdepth);
        B=x';
        B=B(:)';
        B(B==0) = [];
        percent=(length(B))*15/100;
        percent=round(percent);
        orden=sort(B);
        result=orden(1:percent);
        dist=mean(result);
        scores2=scores(k,:);

        figure
        ibox = insertObjectAnnotation(im,'rectangle',bboxes2,scores2);
        imshow(ibox)
        title(['Distance= ',num2str(dist),' [m]'])
    end
end
end

```


Ikaskuntza sakonak aurrerapauso handia suposatu du ibilgailu autonomoen munduan. Sare neuronalen garatzeak robot hauen sistemen hobekuntza bultzatu du. Beste alde batetik, ikusmen artifiziala, duen praktikotasun eta sinpletasunagatik nabarmentzen da, merkea izateaz gain, jasotako informazioa oso baliagarri bihurtzen duelako. Ondorioz, teknika asko garatu dira ikaskuntza sakona eta ikusmena konbinatzen dituztenak ibilgailu autonomoetan aplikatzeko. Alabaina, ikaskuntza sakoneko alor askotan optimizazioaren falta existitzen jarraitzen du. Barnealdeko ibilgailuen kontrolean fokua jarriz, ikerketa honek autonomoa den nabigazio sistema bat planteatzen du. Helburu horretarako ikaskuntza sakoneko tekniken optimizazioa egiten da eta ikusmen artifizialeko teknikak erabiltzen dira. Nabigazio algoritmoa hiru moduluz osatuta dago, hauek ibilbide plangintza eta kontsigna lorketa, aldibereko lokalizazio eta mapatzea, eta segurtasunerako oztopoen errekonozimendua izanik. Beraz, algoritmoa ingurune batean ibilbide bat eraikitzeko gai izan behar da, aldiune bakoitzean eskuratu beharreko posizioa ezagutuz. Ingurune hori ezezaguna izango denez bertako mapa bat eraiki beharko da, lokalizatzeko ahalmena mantenduz. Bertan oztopoak egotea ohikoa izango denez, hauek detektatu beharko dira segurtasuna mantenduko duten ekintzak erabakitzeko.

