

# UNIVERSITETI I PRISHTINËS “HASAN PRISHTINA”

## Fakulteti i Inxhinierisë Mekanike

Prishtinë

UNIVERSITETI I PRISHTINËS “HASAN PRISHTINA”  
FAKULTETI I INXHINIERISË MEKANIKE  
PRISHTINË

Granuar me: 28.10.2024			
Nj. org.	Numër	Shtojca	
08	1648	—	—

## Këshillit të Fakultetit të Inxhinierisë Mekanike

P r i s h t i n ë

Në bazë të vendimit nr. 480/1 të datës 25.03.2024, të Këshillit të Fakultetit të Inxhinierisë Mekanike në Prishtinë është formuar komisioni në përbërje:

1. Prof. Dr. Arbnor Pajaziti, *kryetar*
2. Prof. Dr. Ahmet Shala, *anëtar*
3. Prof. asoc. Dr. Xhevahir Bajrami, *mentor*

Për vlerësimin e punimit Master me titull “**Integrimi i Robotit A1 me Kamerën Unitree duke Përdorur ROS: Ngritja e Perceptimit dhe Kontrollit**” të kandidatit bachelor **Fitore Basholli**.

Pas kontrollimit të punimit të lartpërmendur Komisioni jep këtë:

### R A P O R T

Punimi Master me titull “**Integrimi i Robotit A1 me Kamerën Unitree duke Përdorur ROS: Ngritja e Perceptimit dhe Kontrollit**” është hartuar në 7 (shtatë) kapituj kryesor dhe është ilustruar me figura, grafe dhe tabela të nevojshme.

**Kapitulli i parë** paraqet hyrjen dhe qëllimin e hulumtimit, duke adresuar rëndësinë e integritit të teknologjive të avancuara të perceptimit dhe kontrollit për robotët katërkëmbësh. Në këtë kapitull, theksohet rëndësia e hulumtimit në fushën e robotikës, me fokus të veçantë në nevojën për të integruar robotët me sensorë të përparuar dhe sistemin ROS (Robot Operating System). Kapitulli synon të paraqesë objektivat kryesore të studimit, të cilat përqendrohen në përmirësimin e perceptimit dhe kontrollit të robotëve në mjedise komplekse. Në këtë mënyrë, lexuesi do të ketë një kuptim të qartë mbi sfidat dhe mundësitë që do të trajtohen në vazhdim të punimit.

**Kapitulli i dytë**, ky kapitull shqyrton literaturën më të fundit në fushën e robotikës dhe sistemet e integruara të perceptimit, veçanërisht për robotët katërkëmbësh. Diskutohet mbi platformat dhe zgjidhjet më të avancuara që janë zhvilluar për të përmirësuar kontrollin dhe perceptimin

e robotëve, duke përfshirë teknologji të tilla si algoritmet SLAM dhe sistemet vizuale. Kapitulli përshkruan zhvillimet e fundit teknologjike dhe përdorimin e tyre në robotikën moderne, duke krahasuar Unitree A1 me robotë të tjerë të ngjashëm.

**Kapitulli i tretë**, në këtë kapitull paraqet një përshkrim të detajuar të robotit katërkëmbësh Unitree A1, duke u fokusuar në arkitekturën e tij harduerike dhe komponentët kryesorë, si sensorët e këmbëve, motorët dhe kamerat. Kapitulli shqyrton edhe mënyrat e ndryshme të funksionimit të robotit, duke përfshirë modalitetet e ecjes dhe mbrojtjes, dhe mënyrat se si këto kontribuojnë në performancën e përgjithshme të robotit. Përmes analizave të detajuara, kapitullohet potenciali i Unitree A1 për të përmbushur detyra komplekse në mjedise të paparashikueshme.

**Kapitulli i katërt**, kinematika është një element kyç në lëvizjen e robotëve, dhe ky kapitull fokusohet në kinematikën direkte dhe inverse të robotit Unitree A1. Diskutohet modeli matematikor që përshkruan lëvizjen e këmbëve të robotit dhe mënyra se si ky model zbatohet për të kontrolluar me saktësi pozicionin dhe orientimin e tij. Për më tepër, theksohet rëndësia e kinematikës në optimizimin e stabilitetit dhe efikasitetit të lëvizjes së robotit, duke siguruar që roboti të ruajë balancën dhe të përdorë energjinë në mënyrë optimale gjatë lëvizjes në terrene të ndryshme. Në këtë kapitull janë përfshirë gjithashtu ilustrime të këmbëve të robotit, të gjeneruara në MATLAB, si dhe grafikë që paraqesin ndryshimin e këndeve të këmbëve dhe lëvizjen e qendrës së masës së robotit gjatë lëvizjes së tij.

**Kapitulli i pestë**, në këtë kapitull trajtohet sistemi operativ i robotëve (ROS) dhe shpjegohet mënyrën se si ky sistem përdoret për të koordinuar dhe kontrolluar funksionet e robotit Unitree A1. Përmes një analize të detajuar të arkitekturës së ROS-it, përshkruhen mjetet kryesore të tij, si RViz dhe Gazebo, të cilat përdoren për vizualizim dhe simulim. Po ashtu, përshkruhet procesi i instalimit dhe konfigurimit të ROS-it për të mundësuar simulimin dhe navigimin e robotit A1. Ky kapitull ofron një kuptim të qartë për rolin e ROS-it në përmirësimin e aftësive të perceptimit dhe ndërveprimit të robotit me mjedisin, duke theksuar rëndësinë e tij në zhvillimin e sistemeve robotike inteligjente.

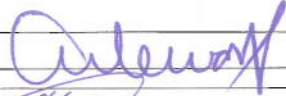

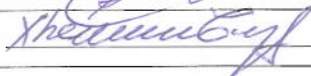
**Kapitulli i gjashtë**, trajton metodat e navigimit për robotët katërkëmbësh, me fokus të veçantë në përdorimin e teknologjive si SLAM (Simultaneous Localization and Mapping) dhe navigimi lokal, të cilat i mundësojnë robotit të lëvizë në mënyrë autonome në mjedise të panjohura. Në të, shqyrtohen në detaje algoritmet e lokalizimit dhe të shmangies së pengesave, duke u fokusuar në rolin e kamerës së thellësisë dhe sensorëve të tjerë, që bashkëpunojnë për të siguruar lëvizje të saktë dhe të sigurt të robotit. Kapitulli ofron një analizë të plotë të

teknologjive dhe metodave që ndihmojnë në përmirësimin e aftësive të navigimit autonom të robotëve katërkëmbësh në mjedise komplekse.

**Kapitulli i shtatë**, që është edhe kapitulli përfundimtar fokusohet në testimet reale të robotit A1, duke përshkruar procesin e lidhjes së robotit me kompjuterin dhe krijimin e hartave të navigimit. Pika fillestare dhe ajo përfundimtare caktohen në hartë, duke u mbështetur në të dhënat e siguruar nga RPLIDAR S1 dhe algoritmet SLAM. Në këtë kontekst, roboti navigon nga pika fillestare, duke shmangur pengesat dhe duke arritur me sukses në pikën përfundimtare. Gjithashtu, në këtë kapitull paraqiten hartat reale të krijuara përmes teknologjisë SLAM dhe RPLIDAR, të cilat demonstrojnë aftësitë e robotit në ambientet e testimit.

## P Ë R F U N D I M

Në bazë të të dhënave të përshkruara më lartë, Komisioni për Vlerësimin e punimit Master konsideron se punimi është hartuar në nivel të duhur, i pasqyruar me figura, diagrame dhe tabela të nevojshme. Prandaj komisioni i propozon Këshillit të Fakultetit të Inxhinierisë Mekanike në Prishtinë, që punimin Master, me titull **“Integrimi i Robotit A1 me Kamerën Unitree duke Përdorur ROS: Ngritja e Perceptimit dhe Kontrollit”** të kandidatit *master Fitore Basholli*, ta aprovoj si punim për Master, dhe ta jep në diskutim publik.

Prishtinë, 21 Tetor 2024	
<b>Komisioni:</b>	
1. Prof. Dr. Arbnor Pajaziti, kryetar	
2. Prof. Dr. Ahmet Shala, anëtar	
3. Prof. Asoc. Dr. Xhevahir Bajrami, mentor	

**UNIVERSITETI I PRISHTINËS “HASAN PRISHTINA”**

**FAKULTETI I INZHINIERISË MEKANIKE**

**PROGRAMI STUDIMOR: MEKATRONIKE**

**DEPARTAMENTI: MEKATRONIKE**

**NIVELI: MASTER**



## **PUNIM DIPLOME**

**Integrimi i Robotit A1 me Kamerën Unitree duke  
Përdorur ROS: Ngritja e Perceptimit dhe Kontrollit**

Mentori:

Prof. Asoc. Dr. Xhevahir Bajrami

Kandidati/ja:

BSc. Fitore Basholli

Prishtinë, 2024

# Përmbajtja

Lista e figurave .....	5
Lista e shkurtesave .....	7
Abstrakti.....	9
Abstract .....	10
1. Hyrje .....	11
2. State of the art (Të rejat e fundit).....	13
3. Roboti A1 Unitree.....	24
3.1. Pjesët e robotit.....	24
3.2. Arkitektura e sistemit robotik.....	25
3.3. Sensori i forcës së këmbës .....	25
3.4. Modaliteti i Funkcionimit të Robotit.....	26
3.4.1. Gjendja Statike e Qëndrimit: .....	26
3.4.2. Modaliteti "A" (Kontroll i Qëndrimit dhe Pozicionit në 3 Aks gjatë Qëndrimit): 26	
3.4.3. Modaliteti "W" (Ecje): .....	26
3.4.4. Modaliteti "Sport":.....	27
3.5. Modaliteti i mbrojtjes .....	27
3.5.1. Mbrojtja nga rënia.....	27
3.5.2. Mbrojtja nga shkëputja .....	27
3.5.3. Frenimi Emergjent .....	27
3.5.4. Motori .....	27
3.5.5. Kujdeset për Motorin .....	27
4. Kinematika e robotit A1.....	29
4.1. Kinematika inverse.....	30
4.2. Kinematika direkte .....	33
4.3. Kinematika e Robotit A1 Unitree përmes MATLAB-it.....	36
5. Sistemi operativ i robotëve (ang. Robot operating system (ROS)) .....	39
5.1. Arkitektura ROS.....	39
5.2. Mjetet për Vizualizim dhe Simulim .....	41

5.3.	Instalimi i ROS.....	42
5.3.1.	Simulimi.....	43
5.3.2.	Lansimi i Gazebo dhe Rviz.....	44
5.3.3.	Simulimi në Gazebo.....	45
5.4.	Harta dhe Navigimi i Robotit në Gazebo .....	47
5.4.1.	Procesi i Hartëzimit me SLAM.....	47
5.4.2.	Procesi i Navigimit të Robotit.....	49
6.	Navigimi i Robotëve katërkëmbësh - Unitree A1.....	50
6.1.	Metodat e Navigimit Lokal.....	50
6.2.	Lokalizimi dhe hartëzimi i njëkohshëm (SLAM) .....	50
6.2.1.	Çfarë është SLAM?.....	51
6.2.2.	Rëndësia e SLAM .....	51
6.2.3.	Funksionimi i SLAM .....	52
6.3.	Llojet e metodave SLAM.....	53
6.3.1.	SLAM Vizuale .....	53
6.3.2.	Lidar SLAM.....	54
6.3.3.	SLAM me shumë sensorë .....	56
6.4.	Zgjedhja e algoritmit të duhur SLAM.....	56
6.5.	Sfidat e zakonshme me SLAM.....	57
6.5.1.	Grumbullimi i Gabimeve të Lokalizimit dhe Optimizimi në Algoritmet SLAM 57	
6.5.2.	Dështimi i Lokalizimit dhe Humbja e Pozicionit në Hartë.....	57
6.5.3.	Kostoja e Lartë Llogaritëse për Përpunimin e Imazhit, Pikës së Resë dhe Optimizimin .....	58
7.	Testimi Real i Robotit.....	59
7.1.	Krijimi i hartes dhe navigimi i robotit.....	59
7.2.	RPLidar S1 .....	61
7.2.1.	Karakteristikat teknike të RPLidarit S1 .....	62
7.2.2.	Funksionimi dhe parimet bazë .....	62
7.2.3.	Përdorimi në aplikimet robotike .....	62
7.2.4.	Avantazhet dhe të metat e RPLidarit S1 .....	62

7.2.5. Zbatimi praktik në robotikë dhe automjetet autonome .....	63
7.3. Navigimi i robotit .....	63
8. Përfundimi.....	66
Referencat .....	67

## Lista e figurave

Figura 1: Arkitektura e sistemit robotik [35] .....	25
Figura 2: Këmba e robotit dhe sensori i forcës .....	25
Figura 3: Mdaliteti i Funksionit të Robotit [36].....	26
Figura 4: Modeli kinematik .....	30
Figura 5: Përkufizimi i modelit të kinematikës dhe parametrave të këmbës robotike.....	31
Figura 6: Vijshmëria në OX.....	31
Figura 7: Vijshmëria në YZ.....	32
Figura 8: Pozicioni i këmbëve të robotit në 3D si dhe pamjet nga vështrimet te ndryshme nga lart dhe anash. ....	36
Figura 9: Paraqitja e lakoreve të cilat tregojnë ndryshimin e këndeve të njëveve për kohën prej 20 sekondash. ....	37
Figura 10: Ndryshimi i pozites së robotit për gjatë X dhe Y është paraqitur me anë të këtyre lakoreve si dhe ndryshimi i tre këndeve në një vend. ....	37
Figura 11: Paraqitja e lakoreve se si ndryshon qendra e masës së rëndesës në funksion të kohës .....	38
Figura 12: Struktura e Catkin_Workspace.....	41
Figura 13 : Ekzekutimi i roscore.....	43
Figura 14: Lanzsimi i Gazebo .....	44
Figura 15: Lansimi i Rviz .....	45
Figura 16: Shtrirja e robotit në tokën e simulimit pas ekzekutimit të kontrolluesit .....	45
Figura 17: Terminali per ekzekutimin e kontrolluesit si dhe paraqitja e ndryshimeve te modeve punuese .....	46
Figura 18: Simulimi i robotit në Gazebo, ku është ngarkuar një ambient virtual që përfaqëson një zyrë.....	47
Figura 19 : Gjatë procesit të hartëzimit në RViz shfaq një vizualizim të hartës së krijuar. ...	49
Figura 20 : Pastrimi i dhomës në mënyrë të rastësishme pa SLAM.....	52
Figura 21: Pastrim me kuptimin e paraqitjes së dhomës me SLAM .....	52
Figura 22: Rrjedhja e përpunimit të SLAM-it [33].....	53
Figura 23: Monokular vSLAM .....	54
Figura 24: Stereo vSLAM.....	54
Figura 25: RGB-D vSLAM [33].....	54



Figura 26: SLAM me 2D LiDAR .....	55
Figura 27: SLAM me 3D LiDAR .....	55
Figura 28: Identifikimi i portit të rrjetit dhe konfirmimi i komunikimit ndërmjet PC-së dhe robotit përmes rrjetit. ....	59
Figura 29: Harta e krijuar ne Paint 3D.....	60
Figura 30: Paraqitja e trajektores së robotit në hartën e krijuar. ....	60
Figura 31: Paraqitja e trajektores së robotit në hartën e krijuar. ....	61
Figura 32: RPLidari S1 që është zbatuar në rastin tonë.....	61
Figura 33: Përgaditja e robotit në pikën fillestare për navigim . ....	64
Figura 34: Roboti në gjysmën e rrugës, duke shmangur pengesat. ....	64
Figura 35: Roboti arrin në pikën përfundimtare me sukses pas përfundimit të navigimit. ....	65

## Lista e shkurtesave

<b>AHRS</b>	Attitude and Heading Reference System
<b>CKF</b>	Cubature Kalman Filter
<b>COM</b>	Center of Mass
<b>CPG</b>	Central Pattern Generator
<b>CPU</b>	Central Processing Unit
<b>DoF</b>	Degrees of Freedom
<b>DSO</b>	Direct Sparse Odometry
<b>DTAM</b>	Dense Tracking and Mapping
<b>FGR</b>	Fast Global Registration
<b>FPFH</b>	Fast Point Feature Histograms
<b>GPS</b>	Global Positioning System
<b>GPU</b>	Graphics Processing Unit
<b>GRF</b>	Ground Reaction Force
<b>GRU</b>	Gated Recurrent Unit
<b>ICP</b>	Iterative Closest Point
<b>IMU</b>	Inertial Measurement Unit
<b>INS</b>	Inertial Navigation System
<b>LiDAR</b>	Light Detection and Ranging
<b>LMPC</b>	Linear Model Predictive Control
<b>LQR</b>	Linear Quadratic Regulator
<b>LSD-SLAM</b>	Large- Scale Direct Monocular SLAM
<b>MACNCK</b>	Multiplicative Additive Cubature Nonlinear Kalman Filter
<b>MPC</b>	Model Predictive Control
<b>NDT</b>	Normal Distributions Transform
<b>NNMPC</b>	Neural Network Model Predictive Control
<b>ORB- SLAM</b>	Oriented FAST and Rotated BRIEF SLAM
<b>PTAM</b>	Parallel Tracking and Mapping
<b>QP</b>	Quadratic Programming
<b>RBFNN</b>	Radial Basis Function Neural Network
<b>RGB</b>	Red, Green, Blue (color model)
<b>RGB-D</b>	Red Green Blue and Depth (camera)
<b>ROS</b>	Robot Operating System
<b>SLAM</b>	Simultaneous Localization and Mapping
<b>SLC</b>	Sequential Linear Control
<b>STT</b>	Speech-to-Text
<b>SVO</b>	Semi- Direct Visual Odometry

<b>SWL</b>	Sliding Window Localization
<b>TBC</b>	Trajectory- Based Control
<b>ToF</b>	Time of Flight
<b>TTS</b>	Text-to-Speech
<b>UWB</b>	Ultra-Wideband
<b>VMC</b>	Virtual Model Control
<b>VMC (repeated)</b>	Virtual Model Control
<b>vSLAM</b>	Visual Simultaneous Localization and Mapping

## Abstrakti

Në këtë punim, paraqitet një qasje gjithëpërfshirëse për zhvillimin dhe përmirësimin e aftësive të perceptimit dhe kontrollit të robotëve katërkëmbësh, duke përdorur sistemin operativ të robotëve (ROS) dhe teknologjinë e avancuar të kamerës së thellësisë Unitree. Qëllimi i këtij hulumtimi është të zgjerojë mundësitë e navigimit dhe ndërveprimit të robotit A1 në mjedise komplekse përmes një platforme të fuqishme perceptimi vizual dhe përpunimi të të dhënave. Kamera Unitree, e pajisur me aftësi të avancuara për regjistrimin e thellësisë dhe sensorë të fuqishëm, ofron informacione të detajuara mbi rrethinën, të cilat integrohen me teknologjinë SLAM për krijimin e hartave dhe lokalizimin e robotit.

Punimi shqyrton edhe algoritmet e kontrollit për përmirësimin e qëndrueshmërisë dhe saktësisë së lëvizjes së robotit, duke analizuar sfidat që lidhen me integrimin e sensorëve të ndryshëm dhe optimizimin e trajektores së lëvizjes. Përmes ROS, arrihet një koordinim i avancuar mes sensorëve dhe komandave të kontrollit, duke u mundësuar robotit që të reagojë shpejt dhe me saktësi ndaj pengesave dhe ndryshimeve të papritura në terren. Kjo qasje jo vetëm që përmirëson sigurinë dhe stabilitetin e lëvizjes, por gjithashtu ofron një ndërveprim më të mirë me mjedisin, duke rritur aftësinë e robotit për të naviguar në mënyrë autonome në mjedise të brendshme dhe të jashtme.

Në aspektin praktik, ky studim përfshin testime dhe simulime në Gazebo, një platformë e avancuar simulimi për robotikë, ku tregohet se si roboti A1 mund të lëvizë në hapësira të simuluar, duke përdorur kamera dhe sensorë për të shmangur pengesat dhe për të ndjekur rrugët e përcaktuara. Hulumtimi përqendrohet gjithashtu në optimizimin e algoritmeve SLAM për të rritur performancën e perceptimit dhe shmangien e gabimeve të lokalizimit.

Rezultatet e këtij studimi tregojnë se integrimi i suksesshëm i robotit A1 me kamerën e thellësisë dhe ROS jo vetëm që përmirëson ndjeshëm perceptimin e tij, por edhe aftësitë për të marrë vendime inteligjente në kohë reale, duke krijuar kështu një platformë të fuqishme për aplikime në kërkime shpëtimi, inspektim industrial, dhe detyra të tjera të sofistikuar në mjedise të paparashikueshme. Ky integrim përfaqëson një hap të rëndësishëm drejt zhvillimit të robotëve autonomë më të avancuar dhe më të sigurt, me aplikacione të gjera në industrinë moderne dhe robotikën.

## **Abstract**

This paper presents a comprehensive approach to developing and enhancing the perception and control capabilities of quadruped robots by utilizing the Robot Operating System (ROS) and advanced Unitree depth camera technology. The objective of this research is to expand the A1 robot's navigation and interaction capabilities in complex environments through a robust visual perception and data processing platform. The Unitree camera, equipped with advanced depth sensing and powerful sensors, provides detailed environmental information, which is integrated with SLAM (Simultaneous Localization and Mapping) technology for map generation and robot localization.

The paper also examines control algorithms aimed at improving the robot's movement stability and accuracy, analyzing the challenges related to integrating various sensors and optimizing the robot's motion trajectory. Through ROS, an advanced coordination is achieved between sensors and control commands, enabling the robot to respond swiftly and accurately to obstacles and sudden changes in terrain. This approach not only enhances the safety and stability of the robot's movements but also improves its interaction with the environment, increasing its capability to autonomously navigate both indoor and outdoor settings.

From a practical standpoint, this study includes tests and simulations in Gazebo, an advanced robotics simulation platform, demonstrating how the A1 robot can move through simulated environments using cameras and sensors to avoid obstacles and follow predefined paths. The research also focuses on optimizing SLAM algorithms to improve perception performance and minimize localization errors.

The results of this study show that the successful integration of the A1 robot with the depth camera and ROS not only significantly enhances its perception but also its ability to make intelligent real-time decisions, creating a powerful platform for applications in search and rescue, industrial inspection, and other sophisticated tasks in unpredictable environments. This integration represents a significant step towards the development of more advanced and safer autonomous robots, with wide-ranging applications in modern industry and robotics.

# 1. Hyrje

Në një epokë të zhvillimit të shpejtë të teknologjisë së robotikës dhe inteligjencës artificiale, Unitree A1 dallohet si një shembull i shkëlqyer i inovacionit, duke ndryshuar qasjen ndaj perceptimit dhe ndërveprimit. Dizajni inxhinierik i tij është mbresëlënës: një sistem i sofistikuar i kamerave të thellësisë me shumë lente, i dizajnuar për të plotësuar mungesat në kuptimin dhe performancën. Duke përdorur teknologjinë e fundit të thellësisë stereo infra të kuqe, Unitree A1 tejkalon standardet e perceptimit vizual, duke ofruar një nivel të jashtëzakonshëm të saktësisë dhe thellësisë. Me një gamë prej 0,3 deri në 10 metra, ai mund të navigojë ndër objekte të afërta dhe largëta, duke përfshirë një gamë të gjerë të situatave. Përveç qartësisë së jashtëzakonshme të imazhit me rezolucion 1080P, Unitree A1 ofron një nivel të lartë të saktësisë, duke vendosur standarde të larta për teknologjinë e sensorit të thellësisë. Në një kohë ku perceptimi është çelësi, Unitree A1 shërben si pionier duke fuqizuar botën e robotikës dhe A1 me një perceptim të thellë dhe të saktë të mjedisit për të kuptuar dhe lundruar botën në mënyrë efikase dhe të saktë.

Unitree A1 është një shembull i shkëlqyer i inxhinierisë së lehtë, por të fuqishme, duke vendosur standarde të reja në botën e robotikës. Me një peshë të lehtë prej vetëm 5 kg dhe një kapacitet efektiv ngarkese, ai trajton pa mundim detyra të ndryshme me saktësi, duke e bërë atë një aset të domosdoshëm në të gjithë industrinë. Unitree A1 tregon një qëndrueshmëri mbresëlënëse prej 1-2,5 orë, duke siguruar funksionim të zgjatur pa kompromis. Sistemi i tij i funksionimit në kohë reale, i mbështetur nga teknologjia më e fundit, siguron performancë të qetë në çdo situatë. Siguria është një prioritet i rëndësishëm për Unitree A1, e cila ofron tre mënyra të veçanta të mbrojtjes për siguri të pashembullt në operacione. Ndërkohë, sistemi i tij motorik, i pajisur me një çift rrotullues të përbashkët prej 33,5 NM (MAX) dhe një shpejtësi maksimale të bashkimit prej 21 rad/s, jep fuqi dhe shkathtësi të pashoqe. Me kushineta me rrotulla të shkallës industriale, mirëmbajtja është e lehtë, duke siguruar jetëgjatësi dhe rezistencë aksiale ndaj goditjeve. Aftësitë e sensorit të Unitree A1 janë të pakrahasueshme. Ai lexon të gjitha të dhënat e sensorit me saktësi, duke lejuar që komandat e sakta të pozicionit, shpejtësisë dhe çift rrotullues të dërgohen në secilin motor individualisht. Për më tepër, me katër sensorë të presionit në fundin e këmbës, qasja në informacionin e mbështetjes së këmbës bëhet jo vetëm e saktë, por edhe më e besueshme, efikase dhe e përshtatshme. Shkathtësia plotësohet nga ndërfaqet e pasura të jashtme të Unitree A1. Me porte të dyfishta HDMI, porta të dyfishta Ethernet dhe porta USB të katërfishta, integrimi pa probleme në sistemet ekzistuese është i lehtë, duke ofruar fleksibilitet të pashembullt për një mori aplikacionesh.

Prezantimi i Unitree A1 është një mrekulli e inxhinierisë robotike që ripërcakton shkathtësinë, stabilitetin dhe shpejtësinë. Me një shpejtësi maksimale të vazhdueshme të vrapimit prej 3,3 metra në sekondë (11,88 kilometra në orë), Unitree A1 tejkalon konkurrentët e tij dhe mban titullin prestigjioz të të qenit më i shpejti dhe më i qëndrueshëm në mesin e kolegëve të tij, duke vendosur një rekord botëror prej 3,7 metra në sekondë nën kujdesin e MIT në një rutine. Kjo arritje e jashtëzakonshme e pozicionon Unitree A1 jo vetëm si një makinë të thjeshtë, por si një shoqërues dinamik të aftë për të mbajtur ritmin me lëvizjen njerëzore. Shpejtësia e saj mbresëlënëse, e ngjashme me atë të vrapimit të një të rrituri, e vendos atë si një partner ideal vrapimi, duke u integruar pa probleme në rutinat e përditshme të fitnesit ose aplikacionet

profesionale me efikasitet dhe besueshmëri të pakrahasueshme. Duke shprehur inovacionin dhe performancën, Unitree A1 paralajmëron një epokë të re në robotikë, ku shkathtësia takohet me praktikitetin për të ripërcaktuar mundësitë.

Unitree A1 shfaqet si një kulm i inxhinierisë robotike, me qëndrueshmëri dhe ndjeshmëri të pashembullt, i krijuar me përpikëri për të lundruar në një mori mjedisesh me saktësi. Në thelbin e tij qëndron një algoritëm i avancuar balancues dinamik, duke siguruar stabilitet të patëmetë edhe përballë ndikimeve të papritura. Një nga veçoritë e spikatura të Unitree A1 është integrimi i tij i një kamere standarde të thellësisë RealSense, duke mundësuar aftësi të fuqishme ndërtimi dhe shmangie të pengesave. Për më tepër, sistemi i tij i sensorit të skajit të këmbës, i pajisur me sensorë të integruar të forcës, ofron reagime në kohë reale për gjendjen e kontaktit të secilës këmbë, duke siguruar kontroll të saktë të lëvizjes dhe duke rritur ndjeshmërinë dhe besueshmërinë e përgjithshme. Dizajni i papërshkueshëm nga uji dhe pluhuri i sensorit të këmbës nënvizon më tej qëndrueshmërinë dhe elasticitetin e A1 në kushte të ndryshme funksionimi.

Unitree A1 qëndron në ballë të inovacionit në fushën e robotikës, duke shprehur teknologjitë më të avancuara që ripërcaktojnë standardet e lëvizshmërisë autonome. Me një fokus thelbësor në autonominë e bazuar në vizion, A1 është projektuar për të shkëlqyer në mjedise dinamike, duke ofruar aftësi të pashembullta në ndjekjen dhe shmangien e pengesave. Në zemër të funksionalitetit të tij qëndron sistemi i ndjekjes autonome i bazuar në vizion, një veçori novator që i mundëson A1 të gjurmojë pa probleme objektivat brenda gamës së tij vizuale në kohë reale. Qoftë individë apo objekte specifike lëvizëse, A1 shfaq aftësi të jashtëzakonshme në identifikimin dhe ndjekjen e objektivave me një qëndrueshmëri të jashtëzakonshme. Përmes mësimin në internet, A1 përmirëson vazhdimisht aftësinë e tij për të gjurmuar karakteristikat e synuara, duke siguruar saktësi dhe përshtatshmëri në skenarë të ndryshëm. Me një shpejtësi mbresëlënëse të rifreskimit të trafikut video prej 25-30 korniza për sekondë, shoqëruar me një ekran të besimit në kohë reale të përfaqësuar nga kutia e gjurmimit që ndryshon ngjyra, duke filluar nga jeshile në të kuqe për të treguar besueshmërinë e gjurmimit nga 100% në 0%, A1 vendos një standard të ri për përgjegjshmërinë dhe saktësinë. Duke shmangur me shpejtësi pengesat brenda 0,8 metrave nga këndi i tij i shikimit, A1 monitoron në mënyrë dinamike shpërndarjen e pengesave në kohë reale, duke rregulluar pozicionin e trupit të tij në përputhje me rrethanat. Mekanizmat e reagimit në kohë reale u ofrojnë përdoruesve njohuri thelbësore për këndin e rregullimit të trupit të robotit, distancën maksimale të pengesës dhe zonën brenda këndit të shikimit dhe trajektoren e ardhshme, duke siguruar manovrim dhe siguri pa probleme. Për përdoruesit që kërkojnë funksione të avancuara, A1 ofron mbështetje opsionale Lidar me precizion të lartë, duke mundësuar ndërtimin e hartës, pozicionimin autonom, planifikimin e navigimit dhe shmangien dinamike të pengesave. Kjo veçori shtesë rrit më tej shkathtësinë dhe përshtatshmërinë e A1, duke e bërë atë një aset të domosdoshëm për një mori aplikacionesh. Në thelb, Unitree A1 ripërcakton kufijtë e robotikës, duke kombinuar autonominë vizionare me besueshmërinë e pashembullt për të vendosur një standard të ri në performancën dhe aftësinë robotike. Pavarësisht nëse bëhet fjalë për lundrimin në mjedise komplekse ose ekzekutimin e manovrave të sakta, A1 qëndron si një testament i potencialit të pakufishëm të robotikës në formësimin e së ardhmes së teknologjisë dhe inovacionit.

## **2. State of the art (Të rejat e fundit)**

### ***MIT Cheetah 3: Projektimi dhe kontrolli i një roboti katërkëmbësh të fuqishëm dhe dinamik [2]***

Ky punim ka prezantuar robotin katërkëmbësh MIT Cheetah 3 si një platformë të re robotësh me këmbë. Një dizajn i përmirësuar në krahasim me MIT Cheetah 2, ky robot ka një gamë më të madhe lëvizjesh me aftësi të plota kontrolli 3D. Fortësia e demonstruar pa përdorimin e sensorit të jashtëm do të thotë se është në gjendje të lëvizë me sukses në situata sfiduese pa u mbështetur në njohuritë e mjedisit përpara se të ekzekutojë strategjinë e tij të kontrollit. Arkitektura e përgjithshme e kontrollit është e përshtatshme për përdorimin e çdo numri kontrolluesish të ndryshëm me modifikime minimale në softuer ose pajisje. Rezultatet fillestare premtuese të një kuadri të ri të kontrollit parashikues të modelit të rregulluar jolinear të politikave (PR-MPC) zgjerohen më tej në kontrollorët aktualë të paraqitur duke optimizuar si forcat e reagimit tokësor ashtu edhe vendndodhjet e hapave duke përdorur heuristika të thjeshta të bazuara në fizikë për të rregulluar parashikimin [3]. Roboti do të vendosë në mënyrë inteligjente se ku t'i vendosë më mirë këmbët për çdo ecje të caktuar dhe do të përfitojë nga dinamika e tij për lëvizjen e qëndrueshme dhe refuzimin e shqetësimeve. Ndërsa roboti aktualisht funksionon duke marrë hyrjet e përdoruesit për shpejtësinë e komanduar dhe shpejtësinë e rrotullimit, do të zhvillohet një planifikues i nivelit më të lartë të rrugës për të lejuar funksionimin autonom në botë. Ky planifikues i nivelit më të lartë mund të integrohet natyrshëm në kornizë si si planifikues i rrugës, ashtu edhe si ofrimin e informacionit për përmirësimin e kontrolluesve dhe vlerësimin e gjendjes.

### ***Dizajni dhe kontrolli dinamik i lëvizjes së robotit katërkëmbësh me përshtatje terreni më pak perceptim [4]***

Lëvizja dinamike dhe përshtatshmëria ndaj terrenit kompleks janë tregues të rëndësishëm të performancës së robotit me katër këmbë. Me anë të analizës krahasuese të strukturave të ndryshme, ky studim projektoi një robot paralel simetrik katërkëmbësh me avantazhet e prodhimit të forcës së fortë dhe performancës së shkëlqyer të lëvizjes. Më pas, u krijua një kornizë kontrolli e bazuar në VMC për të rritur aftësinë e robotit për t'u përshtatur në terrene të panjohura komplekse. Ky kuadër përfshin kryesisht shpërndarjen optimale të GRF-ve bazuar në QP dhe një metodë të plotë të përshtatjes së terrenit. Metoda e propozuar e përshtatjes së terrenit përdor vlerësimin e planit mbështetës hapësinor bazuar në metodën e përgjithësuar të katrorëve më të vegjël dhe bashkon informacionin e orientimit të trupit dhe informacionin e koduesit të përbashkët pa informacion shtesë perceptues ose vizual për të realizuar kontrollin e ekuilibrit të robotit duke përshtatur qëndrimin e trungut dhe trajektoren e lëvizjes së këmbës, respektivisht. Kjo metodë është e qëndrueshme ndaj lëvizjes së matjes së qëndrimit të robotit dhe llogaritjet e përfshira janë të lehta. Simulimi dhe rezultatet eksperimentale të marra për lëvizjen e brendshme dhe të jashtme zbulojnë se roboti ka aftësi të forta përshtatëse në terrene të pabarabarta dhe mund t'i rezistojë disa ndikimeve të jashtme, gjë që tregon efektivitetin dhe qëndrueshmërinë e metodës së propozuar. Në një punë të ardhshme, ne planifikojmë të përmirësojmë lëvizjen shumë dinamike, si vrapimi dhe kërcimi i robotit katërkëmbësh.

### ***Mësimi i një kontrolluesi të fortë të bazuar në perceptim për robotin katërkëmbësh [5]***



Kjo punë tregon performancën e një politike studentore optimale të përbërë nga rrjete të ndryshme, duke i mundur asaj të lundrojë në terrene të rastësishme edhe në prani të zhurmave si në vëzhgimet eksterceptive ashtu edhe ato proprioceptive. Politika e studentëve përbëhet nga rrjete të përsëritura, kodues, dekoder dhe MLP. Rrjeti i përsëritur trajton vëzhgimet e zhurmshme dhe vlerëson gjendjet për të zbutur praninë e gjendjeve të pavëzhgueshme. Rrjetet e koduesit dhe dekoderit punojnë së bashku për të reduktuar zhurmat dhe për të rindërtuar vlerat e së vërtetës tokësore të vëzhgimeve. Së fundi, rrjeti MLP përdor informacionin latent nga rrjeti i koduesit për të parashikuar veprimet optimale. Ky dizajn është vërtetuar se rrit qëndrueshmërinë e politikës së kontrollit në prani të zhurmave në vëzhgime. Për më tepër, efektiviteti i trajnimit paralel është demonstruar, duke rezultuar në kohë më të shkurtra trajnimi në krahasim me qasjet e mëparshme në literaturë. Përveç kësaj, është vlerësuar performanca e rrjeteve të zakonshme të përsëritura, me rrjetin GRU që shfaqet si opsioni më efektiv. Në këtë punim, terrenet në hartë janë projektuar afër kufijve kinematik të robotit, me supozimin se politika mund të përshkojë shumicën e zonave pa nevojën për të shmangur pengesat. Punimet e ardhshme mund të sjellin pengesa rreth të cilave roboti duhet të lundrojë, duke bërë të nevojshme optimizimin e strategjive të shmangies së pengesave [6], [7], [8]. Për më tepër, ky studim vendos hiperparametrat vetëm një herë, pasi përdor një hartë të vetme në simulim. Hulumtimet e ardhshme mund të përdorin harta të ndryshme për të përmirësuar përgjithësimin e metodës. Për të arritur këtë, përdorimi i metodave adaptive të akordimit të hiperparametrave mund të përmirësojë efikasitetin e të mësuarit [9], [10].

### ***Udhëzues robotik katërkëmbësh me ndërveprim vokal njeri-robot [11]***

Është testuar ndërfaqja e të folurit në simulim duke përdorur një hartë të simuluar navigimi që roboti do ta hartonte duke përdorur kamerën e tij LiDAR në bord, siç është paraqitur në Figurën 1. Në këtë simulim, përdoruesi i tha robotit, "Hej A1, më ço në laborator." Ndërfaqja e të folurit dëgjoi me sukses komandën e përdoruesit dhe e përktheu atë në një varg teksti. Më pas, publikoi koordinatat e paracaktuara të laboratorit nga fjalori te nyja e synimit të lundrimit. Navigimi i robotit ishte në gjendje të abonohej në atë nyje dhe krijoi një shteg për në atë vendndodhje të qëllimit, siç është paraqitur në Figurën 2. Më në fund, roboti iu përgjigj përdoruesit duke thënë: "Mirë, lundrimi në laborator." Përdoruesi më pas tha: "Më çoni në zyrë". Ndërfaqja e të folurit e injoroi me sukses fjalimin edhe pse mund të ishte një komandë, pasi përdoruesi nuk përdori fjalën zgjuese, e cila ishte caktuar në "Hej A1". Navigimi i robotit nuk u ndikua dhe nuk u kthye asnjë përgjigje. Vetëm kur përdoruesi tha të njëjtën fjali, por me fjalën zgjuese, algoritmi e njohu atë si një komandë të vlefshme. Kjo do të thoshte se ndërfaqja e të folurit dërgoi koordinatat e zyrës në tubacionin e navigimit të robotit, gjë që rezultoi në krijimin e një rruge të re të treguar në Figurën 3.

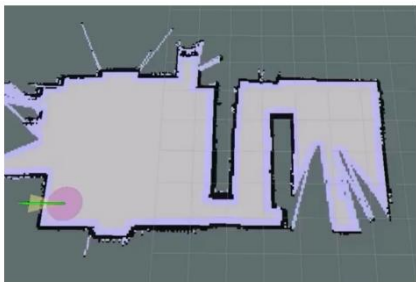


Figure 1. *Harta e simuluar e navigimit që tregon pozicionin fillestar në rrethin e purpurt.*

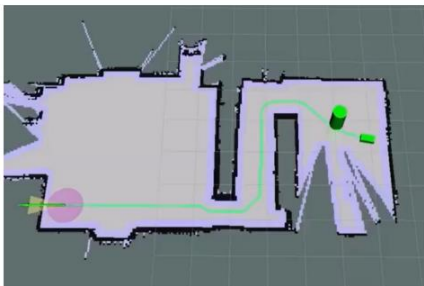


Figure 2. *Harta e lundrimit pas marrjes së koordinatave të synimeve të lundrimit nga ndërfaqja e të folurit. Shifrat e gjelbra portretizojnë pozicionin përfundimtar të përdoruesit dhe robotit, ndërsa vija e gjelbër është rruga e krijuar për në atë qëllim të pozicionit përfundimtar.*

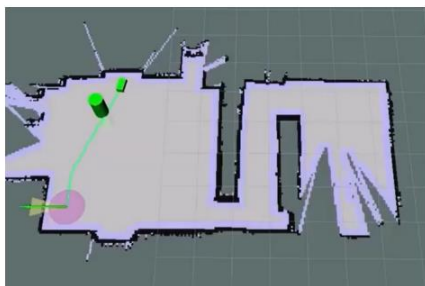


Figure 3. *Harta e navigimit pas marrjes së koordinatave të reja të synimeve të lundrimit nga ndërfaqja e të folurit.*

Pasja e një ndërfaqeje të të folurit e bën më të thjeshtë për përdoruesin që t'i dërgojë komanda robotit. Ne zhvilluam dhe testuam një algoritëm të suksesshëm të ndërfaqes së të folurit që është në gjendje të komunikojë me motorët TTS dhe STT, si dhe të lidhet me tubacionin e navigimit të robotit. Përparësitë kryesore të kësaj pune janë se jemi në gjendje të personalizojmë fjalën e zgjimit për shkak se kemi ndërfaqen tonë të zotëruar të të folurit dhe jemi në gjendje të krijojmë komanda të personalizuara mjaft lehtë duke i shtuar ato në fjalorin e fjalëve. Kemi mundësinë të kemi një fjalë zgjimi të personalizuar dhe ta integrojmë këtë ndërfaqe të të folurit me një zinxhir ndërsa përdorim një robot të manovrueshëm.

### ***Sistemi i vizionit të të mësuarit të thellë për rregullimin e modelit të ecjes së robotëve me katër këmbë [12]***

Në këtë punim, është paraqitur dhe vërtetuar një metodë për të kapërcyer terrenin e pastrukturuar duke përdorur një robot katërkëmbësh. Kjo metodë përdor modele të ecjes të modeluara nga robotët dhe identifikimin-karakterizimin automatik të mjedisit për të rregulluar automatikisht këto modele të ecjes. Studimi i sistemeve të lëvizjes të frymëzuara nga biologjia të kafshët katërkëmbëshe ka lejuar imitimin e tyre nga robotë të vërtetë. Këto lëvizje dhe

modele të ecjes janë kombinuar me sisteme inteligjente për të rregulluar lëvizjen në mjedise të pastrukturuara. Kjo njohuri përdoret për trajnimin fillestar të rrjeteve nervore, gjë që ka lejuar kryerjen e suksesshme të zhvendosjes në terren të pastrukturuar me robotin. Faza e simulimit ka lejuar vërtetimin e imitimit të modeleve të ecjes së qenve dhe analizimin e efektivitetit të tyre në lloje të ndryshme të terrenit me pengesa. Në këtë mënyrë, lloji alternativ 2-2 i modelit të ecjes zbulohet si më i përshtatshmi për të kapërcyer mjediset me mbeturina. Ky model konsiderohet se ushqehet në gjeneratorin qendror të modelit të ecjes, i cili shërben si bazë për lëvizjet përpara. Identifikimi vizual autonom i terrenit dhe karakterizimi i pengesave duke përdorur rrjetet nervore konvolucionale ka treguar efikasitet të lartë, me një përqindje të lartë saktësie (>90%) në vendndodhjen e pengesave në mjedise reale dinamike. Kjo metodë krahasohet me standarde të ngjashme moderne dhe përkatëse, duke marrë rezultate optimale të funksionalitetit. Metoda e propozuar ofron një kuptim të zgjeruar të mjedisit dhe stabilitetit të tij për identifikimin e terrenit duke përdorur imazhe RGB. Në të kundërt, qasjet konvencionale që mbështeten në retë e pikave shpesh kërkojnë harta të kalueshmërisë. Megjithatë, këto harta vuajnë nga mungesa e detajeve mjedisore, duke përfshirë stabilitetin e zonës dhe informacionin për pengesat. Metoda e propozuar e bazuar në vizion ka treguar efikasitetin e funksionimit jashtë. Ajo mund të ekstrapolohet në sisteme të tjera robotike dhe mjete lundrimi autonome. Linjat e ardhshme të kërkimit dhe zhvillimit e mëvonshme të bazuara në shkrirjen ndijore me sistemet lidar për të marrë matje më të sakta të mjedisit të karakterizuar mund të zhvillohen prej saj.

### ***GenLoco: Kontrollues të përgjithësuar të lëvizjes për robotët katërkëmbësh [13]***

Në këtë punim, ne prezantuar një kornizë të bazuar në RL për trajnimin e kontrollorëve të përgjithësuar të lëvizjes që mund të vendosen në një grup të ndryshëm robotësh katërkëmbësh. Ne tregojmë se një model i thjeshtë i bazuar në histori, i trajnuar në robotë të gjeneruar proceduralisht në simulim, mund të transferohet me sukses në një shumëllojshmëri të madhe robotësh të rinj, të cilët nuk u vëzhguan gjatë trajnimit. Modelet e trajnuara gjithashtu mund të vendosen drejtpërdrejt në robotë të vërtetë, pa kërkuar ndonjë trajnim shtesë në sistemet fizike. Ndërsa eksperimentet tona janë fokusuar në robotët katërkëmbësh, metoda jonë është e përgjithshme dhe mund të zbatohet gjithashtu për robotët në fusha të tjera, si krahët robotikë dhe katërkëmbëshët. Megjithatë, efektiviteti i modeleve tona mbetet i kufizuar tek robotët që kanë të njëjtin numër DoF dhe ndjekin një shabllon morfologjik të paracaktuar. Pavarësisht këtyre kufizimeve, ne shpresojmë se puna jonë do të ofrojë një hap drejt më shumë kontrolluesve me qëllime të përgjithshme që mund të vendosen gjerësisht dhe me lehtësi në një katalog të larmishëm robotësh.

### ***Drejt navigimit robotik të bazuar në mësimin e përforcimit me grafikët e skenës 3D [14]***

Kjo tezë përshkruan hapat fillestarë drejt vendosjes së një politike navigimi të bazuar në mësimin përforcues (RL) në një robot katërkëmbësh Unitree A1 duke përdorur skenën 3D. Ndërsa rezultatet janë shumë paraprake për të qenë përfundimtare, testet tona fillestare tregojnë se transferimi i një politike direkt nga simulimi mund të çojë në rezultate të pavlera kur funksionon në një robot të vërtetë. Ritrenimi i një politike me të dhënat e sensorëve të mbledhura nga ekzekutimi i këtij tubacioni në A1 ka të ngjarë të çojë në performancë më të mirë në detyrat e navigimit pikë-qëllim. Puna e ardhshme mund të eksplorojë përgjithësimin e

metodave të llogaritjes së shpërblimeve të përdorura në politikat e trajnimit për detyrat e pikës qëllimi, eksplorimin dhe kërkimin e objekteve. Performanca e këtyre politikave mund të krahasohet gjithashtu me klasa të ndryshme agjentësh - kjo punë e fokusuar në katërkëmbësh dhe vlerësimet krahasuese mund të bëhen me robotë me rrota. Një tjetër rrugë interesante për punën e ardhshme përfshin reduktimin e llogaritjes në bord, si dhe madhësinë e platformës robotike. Në një shkallë më të madhe, vepra të tilla si [74], të cilat modelojnë emetimet nga llogaritja në bord të një flote globale automjetesh autonome (AV), motivojnë nevojën për rritjen e efikasitetit të harduerit. Autorët tregojnë se energjia e nevojshme për të drejtuar kompjuterët e fuqishëm në AV mund të gjenerojë po aq emetime të gazrave serrë sa të gjitha qendrat e të dhënave në botë sot. Studiues të tjerë po hetojnë robotë që mbledhin energji dhe që nuk kanë nevojë për sasi masive energjie për të funksionuar. Kjo linjë kërkimi tregon se aktivizimi i navigimit të bazuar në RL në robotë të vegjël dhe në shkallë do të kërkojë zhvillimin e arkitekturave të reja llogaritëse për të trajtuar në mënyrë më efikase këto algoritme shumë të kërkuara.

### ***Një metodë tele-operimi me realitet të përzier për kontrollin e nivelit të lartë të një roboti manipulues me këmbë [15]***

Optimizimi i zonës së punës përmes modelimit kinematik ka lejuar përcaktimin e zonave anësore të robotit katërkëmbësh si zona optimale të punës. Kjo metodë ruan stabilitetin e montimit robotik dhe shmang stresin e tepërt në pozicione të kufizuara, duke siguruar mbylljen e sistemeve ndijore të robotit. Simulimi në Gazebo para zbatimit fizik ka vërtetuar kushtet e stabilitetit dinamik për detyrat e manipulimit dhe gjenerimin e trajektoreve pa përplasje për manipuluesin. Përdorimi i ndërfaqeve të përziera për teleoperacion ka treguar përparësi në krahasim me metodat konvencionale. Këto përmirësime kanë rritur besimin e operatorit për vendimmarrje dhe kanë përmirësuar efikasitetin për të kryer detyra të trajtimit të objekteve me gjeometri dhe peshë të parregullt. Gjenerata e dytë e Hololens ka demonstruar performancë më të mirë sesa gjenerata e parë e Hololens dhe ndërfaqet konvencionale, duke përmirësuar cilësinë grafike, kohën e përgjigjes, reduktimin e vonësës dhe lehtësinë e përdorimit. Kjo teknologji për teleoperacion ka një avantazh të rëndësishëm për shkak të përparësive të saj në krahasim me teknologjitë konvencionale që përdorin një ekran dhe tastierë. Për punimet e ardhshme, propozohet studimi i operacioneve në distancë, ku operatori dhe sistemi robotik janë në skenarë të ndryshëm (nga vend i sigurt deri në vend katastrofik), të lidhur përmes rrjetit 5G. Propozohet gjithashtu përdorimi i sistemeve ndihmëse të perceptimit që ofrojnë operatorit informacion shtesë për mjedisin vendimmarrës.

### ***Planifikimi dhe operacionet e robotëve bashkëpunues të aktivizuar me celular për skenarët e kërkim-shpëtimit [16]***

Në operacionet kritike të misioneve, roli i flotës së robotëve bashkëpunues të aktivizuar me celular për aftësitë e kërkimit dhe shpëtimit është vendimtar. Këtu propozojmë një kornizë të re SAR për planifikimin e misionit të robotëve bashkëpunues të aktivizuar me celular. Kjo kornizë përdor informacion hyrës si zona e eksplorimit, madhësia e flotës, profili i energjisë, shkalla e kërkimit të synuar dhe koha e përgjigjes së synuar për të lejuar reaguesve të parë të marrin vendime të informuara për numrin e robotëve të nevojshëm për të përfunduar me sukses një mision. Rezultatet tona demonstrojnë rëndësinë e përshtatjes së flotës së robotëve në varësi

të llojeve të tyre (me rrota kundër katërkëmbësh) në lidhje me numrin e robotëve, zonën e eksploruar dhe kohën e përgjigjes. Për punën e ardhshme, synojmë të zgjerojmë kornizën tonë SAR për të shqyrtuar më tej skenarë në shkallë më të madhe, duke përfshirë zonat më të mëdha, numrin dhe heterogjenitetin e robotëve, diversitetin e terrenit dhe pengesat potenciale. Këto skenarë mund të përfshijnë edhe parametra hyrës të detajuar si informacioni i sipërfaqes, granulariteti më i lartë i energjisë së robotëve dhe kontrolli i tyre. Në raste të tilla, kompleksiteti i problemit mund të rritet, por aplikimi i zgjidhjeve analitike dhe/ose të mësimimit të makinave mund të ndihmojë në marrjen e vendimeve të informuara dhe të realizueshme për misionet e robotëve të kërkimit dhe shpëtimit.

### ***Zhurmat shumëfishuese dhe zhurmat e ndërlidhura shtuese Filtri kubature Kalman dhe aplikimi i tij në robotin katërkëmbësh [17]***

Në këtë artikull propozohet MACNCKF për zgjidhjen e problemeve të vlerësimit të gjendjes që përfshijnë zhurmën shumëfishuese dhe shtesë. MACNCKF kombinon avantazhet e algoritmit CKF dhe kapërcen kufizimet e tij të mundshme. Ky filtr mund të transplantohet lehtësisht në filtra të ndryshëm CKF dhe përmirëson fushën e aplikimit të filtrit CKF duke përfshirë zhurma të pavarura shumëfishuese dhe shtesë. Kur zhurma shumëfishuese dhe zhurma shtesë janë të pavarura nga njëra-tjetra, MACNCKF shfaq performancë të përgjithshme të avancuar, duke rritur kështu qëndrueshmërinë dhe besueshmërinë e tij. Algoritmi është nënshtruar rigorozisht derivimit matematik dhe performanca e tij është verifikuar përmes eksperimenteve simuluese në kohë reale. Aplikimi i MACNCKF në sistemin robotik katërkëmbësh ka konfirmuar saktësinë, fizibilitetin dhe besueshmërinë e tij në situata të ndryshme. Rezultatet e mësipërme tregojnë se MACNCKF ofron performancë të lartë të vlerësimit të gjendjes dhe efikasitet në zgjidhjen e çështjeve të vlerësimit të gjendjes që përfshijnë zhurmën shumëfishuese dhe shtesë.

### ***Mësimi i aftësive të lëvizjes përshtatëse të robotit katërkëmbësh në shpatet bazuar në algoritmin e shtuar të rastit të kërkimit [18]***

Në këtë punim u propozua një metodë për mësimin e aftësive të lëvizjes adaptive të robotit katërkëmbësh në shpatet, me qëllim realizimin e ecjes së qëndrueshme të një roboti katërkëmbësh në shpatet  $5^\circ$  dhe  $11^\circ$ . Së pari, u zhvillua një algoritëm për vlerësimin e pjerrësisë së terrenit, ku pjerrësia vlerësohet bazuar në pozicionin e robotit, për të përcaktuar këndin e dëshiruar të qëndrimit të trupit të robotit. Më pas, algoritmi i kërkimit të rastësishëm të shtuar (ARS) u përdor për të trajnuar një strategji lëvizjeje të robotit katërkëmbësh, duke i mundur robotit që të rregullojë në mënyrë adaptive qëndrimin e tij për të ndjekur këndin e dëshiruar të qëndrimit të trupit. Përveç kësaj, kurba Bezier u përdor për të parametrizuar trajektoren e robotit, gjë që përshpejtoi shumë shpejtësinë e të mësuarit. Fizibiliteti dhe efektiviteti i kornizës së algoritmit u verifikuan në mjedisin e simulimit. Roboti mund të rregullojë në mënyrë adaptive qëndrimin e tij për të realizuar lëvizjen e tij në shpat dhe për të ruajtur stabilitetin e mirë. Tranzicioni nga simulimi në realitet është një punë shumë interesante dhe sfiduese. Për të vërtetuar më tej algoritmin tonë, planifikojmë të kryejmë eksperimente fizike në punën e ardhshme.

### ***Një model kontrolli parashikues i drejtuar nga të dhënat për drejtimin e robotëve katërkëmbësh në sipërfaqe të rrëshqitshme [19]***

Në këtë punim, kontrolli i lëvizjes në një robot katërkëmbësh të simuluar Mini-Cheetah që lëviz në një terren të rrëshqitshëm u analizua thellë. Strategjia e përgjithshme e lëvizjes u konsiderua një detyrë e kontrollit hierarkik. Sinkronizimi i nivelit të ulët të zhvendosjes së fazës midis këmbëve të robotit, për manovrat e përgjithshme duke përfshirë drejtimin, iu besua një CPG. Për kontrollin e trajektores së nivelit të lartë, MPC-të lineare dhe jolineare (të bazuara në rrjetin nervor) u krahasuan. MPC drejton drejtimin e robotit bazuar në një referencë që përbëhet nga shpejtësia këndore (shpejtësia e lëvizjes) dhe që vepron në përfitime specifike duke moduluar sinjalet nervore të aplikuara si referenca të kontrollit të pozicionit në nyjet e robotit. Për të krahasuar siç duhet rezultatet, u zhvillua dhe u optimizua një rrjet nervor dhe një model i funksionit të transferimit linear, duke përdorur një qasje të drejtuar nga të dhënat për të modeluar sjelljen e robotit katërkëmbësh. Të dhënat e miratuara u krijuan duke përdorur një model të robotit Mini Cheetah të simuluar në një mjedis dinamik, si në kushte tipike pune ashtu edhe në prani të terrenit të rrëshqitshëm. Rezultatet e marra me NN MPC u krahasuan me qasjen lineare të bazuar në MPC. Zgjedhja e modelit optimal u arrit në të dyja rastet nëpërmjet indeksit AIC, duke marrë parasysh si saktësinë ashtu edhe kompleksitetin e modelit. Është kryer një analizë krahasuese duke marrë parasysh indekset Fit dhe MSE. Dallimi midis dy sistemeve të kontrollit është i dukshëm në rastin e kushteve përkatëse të rrëshqitjes: LMPC nuk ishte më në gjendje të plotësonte trajektorët e kërkuara të drejtimit, duke shkaktuar rënien e robotit. Arsyeja kryesore për këtë rezultat është se forcat impulsive të krijuara në trupin e robotit gjatë fazës së prekjës dhe qëndrimit të këmbës mund të filtrohen vetëm në rastin e një gjendjeje normale të fërkimit. Në këtë rast, një model linear mund të përafrojë mjaftueshëm përgjigjen dinamike të robotit ndaj komandave të drejtimit. Në rastin e sipërfaqeve të rrëshqitshme, avantazhi i një NN MPC është i dukshëm. Cilësia e arkitekturës së kontrollit u demonstrua gjithashtu duke përdorur indekset tipike të miratuara për lëvizjen me këmbë, të tilla si stabiliteti dhe harmonia. Këto përmirësime sasiore të marra me kontrollin NN MPC janë të dukshme në rastet me fërkim të ulët. Analiza dinamike e simulimit të robotit katërkëmbësh përfaqëson një hap të nevojshëm për implementimet e ardhshme në prototipin e harduerit. Vëmendje e veçantë i është kushtuar reduktimit të kompleksitetit llogaritës në funksion të një zbatimi në bord, si për CPG-në e zhvilluar ashtu edhe për MPC-në, përmes përzgjedhjes së modeleve optimale të robotëve lineare dhe jolineare. Qasja e miratuar, e cila në thelb është e drejtuar nga të dhënat, mund të shtrihet në skenarë të ndryshëm, duke përfshirë terrenin e pabarabartë dhe arkitekturat e ndryshme të robotëve, siç diskutohet në dorëshkrim. Aktivitete të mëtejshme kërkimore do t'i kushtohen integritet të segmenteve shtesë të këmbëve të frymëzuara nga një tarsus i insektit, i cili përmirëson aftësitë ngjitëse përmes zgjidhjeve të ndryshme mekanike, duke përfshirë praninë e kthetrave.

### ***Kontrollimi i robotit katërkëmbësh Solo12 me mësim të thellë përforcues [20]***

Është prezantuar një qasje nga fundi në fund për kontrolluesit e të mësuarit për robotin katërkëmbësh Solo12. Metoda jonë e trajnimit përfshinte zgjedhjen e hapësirës së gjendjes, hapësirës së veprimit dhe funksionit të shpërblimit, së bashku me strategjinë e kurrikulës dhe metodën e rastësisë së fushës/dinamike për të mësuar politikat e transferueshme për ndjekjen e

komandave të shpejtësisë 3D. Janë prezantuar rezultatet për gjurmimin e shpejtësisë dhe humbjen e energjisë. Teste të shumta eksperimentale në robotin e vërtetë treguan se politikat e fuqishme të lëvizjes me profile të ndryshme energjetike mund të mësohen duke randomizuar peshat e variablave të humbjes së fuqisë. Bazuar në këtë punë dhe botime të mëparshme, ne planifikojmë të kryejmë një studim në shkallë të gjerë në të ardhmen e afërt për të krahasuar potencialin e kontrollorëve aktualë të bazuar në model dhe të bazuar në RL në Solo12.

### ***Mësimi i lëvizjes së fuqishme perceptuese për robotët katërkëmbësh në natyrë [21]***

Lëvizje e shpejtë dhe e fuqishme në natyrë është vendosur kontrolluesi jonë në një shumëllojshmëri të gjerë terreni. Kjo përfshin mjediset alpine, pyjore, nëntokësore dhe urbane. Kontrolluesi ishte vazhdimisht i fortë dhe kishte zero rënie gjatë të gjitha vendosjeve. Për shkak të perceptimit eksterceptiv, roboti mund të parashikonte terrenin dhe të përshtatte lëvizjen e tij për të arritur ecje të shpejtë dhe të qetë. Kjo ishte veçanërisht e dukshme për strukturat që kërkojnë hapësirë të lartë të këmbës, si shkallët dhe pengesat e mëdha. Roboti ishte në gjendje të shfrytëzonte inputin eksterceptiv për të pushtuar terrenin që ishte përtej aftësive të punës së mëparshme që nuk përdorte eksterceptivë. ANYmal ka përshkuar me sukses mjedise natyrore sfiduese me pjerrësi të madhe, sipërfaqe të rrëshqitshme, bar dhe borë. Roboti ishte i fortë në këto kushte, edhe kur okluzioni dhe vetitë e sipërfaqes, të tilla si reflektimi i lartë, penguan eksterceptimin. Kontrolluesi ynë u vendos gjithashtu në mënyrë të fuqishme në mjediset nëntokësore me zhavorr të lirshëm, rërë, pluhur, ujë dhe ndriçim të kufizuar. Mjediset urbane paraqesin gjithashtu sfida të rëndësishme. Për kalimin e shkallëve, roboti i fundit katërkëmbësh Spot nga Boston Dynamics kërkon që të aktivizohet një modalitet i dedikuar dhe roboti duhet të orientohet siç duhet në lidhje me shkallët. Në të kundërt, kontrolluesi ynë nuk kërkon ndonjë modalitet të veçantë për shkallët dhe mund të përshkojë shkallët në mënyrë origjinale në çdo drejtim dhe çdo orientim, si p.sh. anash, diagonalisht dhe rrotullimi në shkallë. Shikoni filmin S1 për demonstrime të kalimit të qetë dhe të fortë të shkallëve në drejtime arbitrare me kontrolluesin tonë. Kontrolluesi ishte gjithashtu i qëndrueshëm ndaj kombinimeve të sfidave të ndryshme, madje edhe në shkallët me borën. Bora i bën shkallët të rrëshqitshme dhe jep të dhëna eksterceptive jo të plota dhe të gabuara. Sensorët e thellësisë ose dështojnë për shkak të reflektueshmërisë së lartë të borës ose vlerësojnë se profili i sipërfaqes është në majë të borës, ndërsa këmbët e robotit zhyten nën këtë nivel. Rrëshqitja e këmbës në dëborë mund të shkaktojë gjithashtu zhvendosje të madhe në vlerësimin e pozës kinematike, duke e bërë hartën edhe më të paqëndrueshme. Megjithatë, kontrolluesi mbeti vazhdimisht i fortë, me zero dështime edhe në këtë regjim.

### ***Një algoritëm mësimi i nevojave i aplikuar për gjenerimin e ecjes së qëndrueshme të robotit me katër këmbë [22]***

Bazuar në teorinë e hierarkisë së nevojave të Maslow-it, ky punim lidhi nivelin e lëvizjes së një individi me sjelljet shprehëse dhe përballuese të Maslow-it, dhe propozoi një algoritëm të të mësuarit të nevojave që mund të kombinojë mjedisin dhe nevojat e vetë agjentit për të marrë vendime sipas gjendjeve të ndryshme. Formulimi i tij i përgjithshëm dhe procesi janë gjithashtu ofruar. Përveç kësaj, projektua një eksperiment për të krahasuar të mësuarit e nevojave me të mësuarit të forcuar. Rezultatet eksperimentale treguan se të mësuarit e nevojave performoi më mirë se të mësuarit të forcuar në problemet që kërkonin një shkëmbim ndërmjet nevojave të

vetë agjentit dhe nevojave të objektivit. Më në fund, e vendosëm këtë qasje në një robot katërkëmbësh për të gjeneruar ecje të qëndrueshme. Rezultatet treguan se mësimi i nevojave funksionon mirë në këtë detyrë dhe është gjithashtu më i mirë se të mësuarit përforcues. Për ta përmbledhur, kuadri i të mësuarit të nevojave i kushton më shumë vëmendje ndikimit të motivimit të brendshëm në sjellje sesa të mësuarit përforcues. Përmes një modeli valë motivimi të frymëzuar nga teoria e hierarkisë së nevojave, ne përcaktuam sasinë e nevojave dhe vendosëm një mekanizëm të të mësuarit në të cilin veprimi i agjentit korrespondonte me lëvizjen. Ne gjithashtu përcaktuam se mund të jetë më mirë të simulojmë nivele të shumta nevojash në mekanizmat njerëzorë për të përmbushur disa detyra. Kjo gjithashtu ofron një bazë të rëndësishme për kërkime të mëtejshme.

### ***Modelimi kinematik dhe dinamik i gjithë trupit për robot katërkëmbësh nën ecje dhe topologji të ndryshme mekanizmash [23]***

Ky punim prezantoi metodat e modelimit kinematik dhe dinamik të të gjithë trupit për robotët katërkëmbësh bazuar në teorinë e vidhave. Krahasuar me modelet tradicionale LIP ose centroidale, ky model merr parasysh masën dhe inerciën 10-dimensionale të të gjitha pjesëve, duke ofruar saktësi më të lartë. Modeli u nda në tre faza: qëndrim, ecje dhe fazë të lidhur. Strategjitë e kontrollorëve dhe planet e bazuara në këto modele u propozuan për secilin shtet. Lakoret e çift rrotullimit motorik për ecje të ndryshme u llogaritën dhe u krahasuan me softuerin simulues, duke treguar se rezultatet e llogaritjes dhe simulimit ishin të përputhshme. Janë kryer eksperimente prototipe të fazës në këmbë, dhe rezultatet treguan se kthesat e matura ishin shumë afër me ato teorike. Në të ardhmen, do të fokusohemi në identifikimin e parametrave dinamikë. Gjithashtu, modelet e kontaktit duhet të përmirësohen duke marrë parasysh fërkimin, për të arritur rezultate më të mira në kushte të rrëshqitjes.

### ***Një qasje dinamike e balancimit për një robot katërkëmbësh të mbështetur nga këmbët diagonale [24]***

Kontrolli i bilancit është një konsideratë shumë e rëndësishme për robotët me këmbë; shumica e tyre ruajnë ekuilibrin me një poligon mbështetës të formuar nga këmbë të mëdha ose më shumë se tre këmbë, ose janë robotë që kërcëjnë që duhet të kërcëjnë gjatë gjithë kohës për të mbajtur ekuilibrin. Në këtë punim, u propozua një metodë e re e kontrollit të bilancit për të stabilizuar lëvizjen e një roboti katërkëmbësh. Modeli përdor vetëm dy këmbë diagonale për të mbështetur trupin, dhe secila këmbë bie në kontakt me tokën vetëm në një pikë të vetme, kështu që nuk përfshihet as një poligon mbështetës dhe as lëvizje kërcimi. Disa modele ishin ndërtaur për të zgjidhur problemin e mësipërm; këto evoluuan hap pas hapi, përpara se të ndërtohej përfundimisht modeli mbështetës i këmbës diagonale. Në metodën e propozuar, kontrolli LQR u përdor për të stabilizuar sistemet. Për të kontrolluar modelin me gjashtë DOF, kontrolli i të gjithë trupit të robotit u nda në tre pjesë të pavarura, me secilën pjesë të stabilizuar nga kontrolluesi i vet. VMC është përdorur në modelin e fundit. Secili model u testua nën disa lloje shqetësimesh në Robot- Builder, dhe rezultatet e simulimit treguan se të gjithë kishin stabilitet të mirë; ata jo vetëm që mundën të ruanin ekuilibrin, por edhe u rezistuan këtyre shqetësimeve. Në veçanti, këto modele treguan qëndrueshmëri të madhe ndaj shqetësimeve të shpejtësisë anësore; trangu do të rregullonte papritur pozicionin e tij automatikisht për të zvogëluar shpejtësinë anësore dhe më pas do të kthehej në një gjendje të qëndrueshme. Kjo metodë mund



të zbatohet gjithashtu për robotët me këmbë me numër të tjerë këmbësh: robotët me dy këmbë mund të përdorin drejtpërdrejt modelin e parë dhe robotët me gjashtë këmbë ose oktopodë gjithashtu mund të mbajnë ekuilibrin duke përdorur vetëm dy këmbë mbështetëse.

### ***Një metodë e re e kontrollit të përcjelljes së trajektores për robotët katërkëmbësh plotësisht me lëvizje elektrike [25]***

Në këtë punim, është projektuar një metodë e re e kontrollit të gjurmimit të trajektores për robotët me katër këmbë. Metoda vendos TBC bazuar në teorinë MPC dhe SWL bazuar në RBFNN dhe modelet virtuale, respektivisht. TBC plotëson dy aspekte të modelit parashikues dhe zgjidhjen optimale për gjurmimin e trajektores së robotit katërkëmbësh sipas teorisë MPC. SLC kombinon avantazhet e modelit virtual dhe RBFNN, dhe kontrolluesi VMC dhe kontrolluesi RBFNN janë projektuar, përkatësisht, dhe daljet e dy kontrollorëve janë të koordinuara me një funksion peshe. Metoda e kontrollit të këtij punimi është verifikuar eksperimentalisht: eksperimentet e gjurmimit të trajektores në fund të këmbës tregojnë se metoda e kontrollit zgjidh problemet e gabimit të gjendjes së qëndrueshme dhe efektit të pakënaqshëm të gjurmimit të metodave konvencionale VMC; eksperimentet e gjurmimit të trajektores lineare dhe rrethore me COM tregojnë se metoda e kontrollit mund të arrijë gjurmim të shpejtë dhe të saktë për trajektorët e dëshiruara, e cila mund të plotësojë kërkesat e kontrollit të robotit katërkëmbësh dhe është e lehtë për t'u aplikuar në inxhinieri.

### ***Një robot me katër këmbë për shmangien e pengesave dhe personel që ndjek strategjinë e bazuar në radar lazer me brez ultra të gjerë dhe tredimensionale [26]***

Ky artikull paraqet një strategji të shmangies së pengesave dhe ndjekjes njerëzore për robotin katërkëmbësh bazuar në radarin lazer 3D dhe sistemin e pozicionimit UWB. Sistemi ynë UWB miraton metodën e tre stacioneve bazë, dy etiketave dhe një monitorimi. Sipas informacionit origjinal të distancës midis etiketës dhe stacionit bazë, futet algoritmi i pozicionimit për të llogaritur koordinatat, në mënyrë që të përcaktojë pozicionin aktual të robotit katërkëmbësh dhe pozicionin e personit të gjurmuar. Reja origjinale e pikës merret nga pengesat e skanimit të radarit lazer. Pas përpunimit të një pike të caktuar të resë, merret informacioni i thjeshtuar i koordinatave të pengesave. Rrjeti 2D i informacionit të koordinatave të pengesave është më i favorshëm për t'u integruar në algoritmin e planifikimit të rrugës. Informacioni i pozicionit të marrë nga UWB është gjithashtu i hartuar në hartën e rrjetit 2D përmes transformimit të koordinatave. Së fundi, algoritmi inkremental A\* i bazuar në një dritare rrëshqitëse përdoret për të realizuar shmangien e pengesave dhe gjurmimin njerëzor të robotit katërkëmbësh. Megjithëse UWB është përmirësuar shumë në saktësinë e pozicionimit, ajo do të luhet për shkak të ndërhyrjeve të jashtme në komunikim. Për shembull, ndikimi i okluzionit do të zbusë fuqinë e sinjalit të shkaktuar nga gabimi i pozicionimit dhe madhësive të ndryshme. Mbyllja materiale në ndikimin e sinjaleve të komunikimit është e ndryshme. Në testin aktual, roboti katërkëmbësh do të shfaqet herë pas here me luhate të vogla në procesin e lëvizjes. Prandaj, në punën e ardhshme, ne konsiderojmë shkrirjen e mëtejshme të sensorëve të tjerë mbi këtë bazë për të formuar një sistem robotik katërkëmbësh me shkrirje me shumë sensorë për shmangien dhe gjurmimin e pengesave. Në aspektin e planifikimit të rrugës, puna e ardhshme do të marrë në konsideratë bashkimin e algoritmit të planifikimit global të rrugës dhe algoritmit

lokal të planifikimit të rrugës për të përmirësuar më tej stabilitetin e algoritmit të planifikimit të rrugës.

### ***Projektimi strukturor, simulimi dhe eksperimenti i robotit katërkëmbësh [27]***

Ky punim fokusohet në dizajnin strukturor, modelimin kinematik, planifikimin e ecjes, simulimin e dinamikës, fabrikimin fizik dhe verifikimin eksperimental të një roboti katërkëmbësh. Modeli Ristow përdoret për të karakterizuar fenomenin dinamik jolinear të kontaktit këmbë-tokë. Një ekuacion i përbërë i trajektores së këmbës cikloide është zhvilluar për të zvogëluar ndikimin në tokë në momentin e uljes në një masë të caktuar, gjë që është e dobishme për stabilitetin e robotit. Është projektuar dhe zbatuar një model fizik i robotit katërkëmbësh. Mikrokontrolluesi STM32 zgjidhet si kontrollues dhe një sistem FreeRTOS përdoret si kornizë kontrolli. Në eksperimentin diagonal të ecjes me trot dhe testin e vetë-balancimit, roboti katërkëmbësh është në gjendje të kontrollojë këndet e rrotullimit dhe të hapit të bustit të tij. Një sasi e caktuar rrëshqitjesh vërehet midis këmbës së robotit dhe tokës, e cila kufizon shpejtësinë e lokomotivës së robotit katërkëmbësh. Shkaqet e mundshme përfshijnë modelimin dhe kontrollin e pamjaftueshëm të kontaktit dhe ndërveprimit këmbë-tokë, si dhe përdorimin e servove hobi pa reagime, të cilat do të jenë fokusi i kërkimit tonë të ardhshëm. Dizajni i kësaj pune ofron një platformë të përbalueshme por gjithëpërfshirëse për studiuesit fillestarë në këtë fushë për të studiuar problemet komplekse të dinamikës, modelimit të kontakteve, planifikimit të ecjes dhe kontrollit të qëndrimit për robotët me katër këmbë.

### 3. Roboti A1 Unitree

A1 përbëhet kryesisht nga sistemet e kontrollit, sistemet e komunikimit, sistemet e energjisë dhe paketat e baterive. Në vazhdim do të tregohen funksionet e komponentëve të ndryshëm të robotit.

#### 3.1. Pjesët e robotit



1. Trupi
2. Kapaku i përparmë
3. Pllaka mbështetëse abdominale
4. Montimi i këmbës
5. Nyja e ijës
6. Pjesa e kofshës
7. Nyja e gjurit
8. Pulpa



9. Motori dhe reduktori i kofshës
10. Motori dhe reduktori i pulpës
11. Ndërprerësi i energjisë
12. Motori dhe reduktori i trupit
13. Kapësja e baterisë
14. Kamera inteligjente me thellësi
15. Rregullatori i këndit të kamerës



16. Ndërfaqet e rezervuara për energji (në zhvillim)
17. Ndërfaqet e rezervuara për komunikim (në zhvillim)
18. Lidari

### 3.2. Arkitektura e sistemit robotik

Arkitektura bazë e diagramit skematik të sistemit është treguar më poshtë.

Sistemi operativ i PC-së së integruar është Linux në kohë reale (Ubuntu), i cili ka një brez komunikimi maksimal në kohë reale prej 1000Hz. Nëse platforma X86 është Upboard dhe nëse keni nevojë të përdorni ndërfaqen grafike HDMI, duhet të lidhni HDMI-në përpara ndezjes së pajisjes.

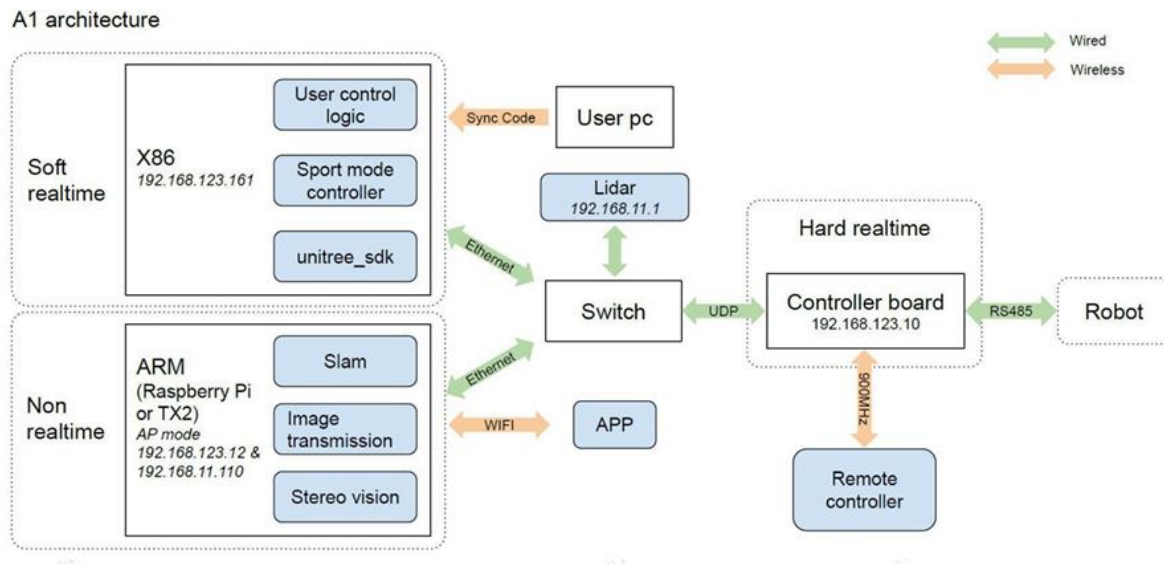


Figura 1: Arkitektura e sistemit robotik [35]

### 3.3. Sensori i forcës së këmbës

Çdo këmbë e robotit A1 është e pajisur me sensorë të forcës, të vendosur në të njëjtën pozicion. Pozicioni i tyre është si më poshtë:



Figura 2: Këmba e robotit dhe sensor i forcës

Pjesa e kaltër në figurë përfaqëson sensorin, ndërsa shigjeta e kuqe tregon drejtimin e sensorit të forcës. Drejtimi i forcës në sensor përcaktohet nga forma specifike e kontaktit midis skajit të këmbës dhe tokës, që është pingul me sipërfaqen e kontaktit. Në rast se kontakti i këmbës

është në disa pika, madhësia e forcës mund të barazohet me forcën e kombinuar të këtyre pikave kontakti. Ky sensor është i ndjeshëm ndaj drejtimit të vijës vertikale në çdo pikë të sipërfaqes sferike. Për më tepër, sensori ka tendencë të devijojë, prandaj është e nevojshme kalibrimi periodik i pikës zero (si për shembull kur këmba është e shkëputur nga toka).

### 3.4. Modaliteti i Funkzionimit të Robotit

A1 përdor një arkitekturë të re harduerike dhe një sistem kontrolli. Sistemi i kontrollit është si më poshtë:

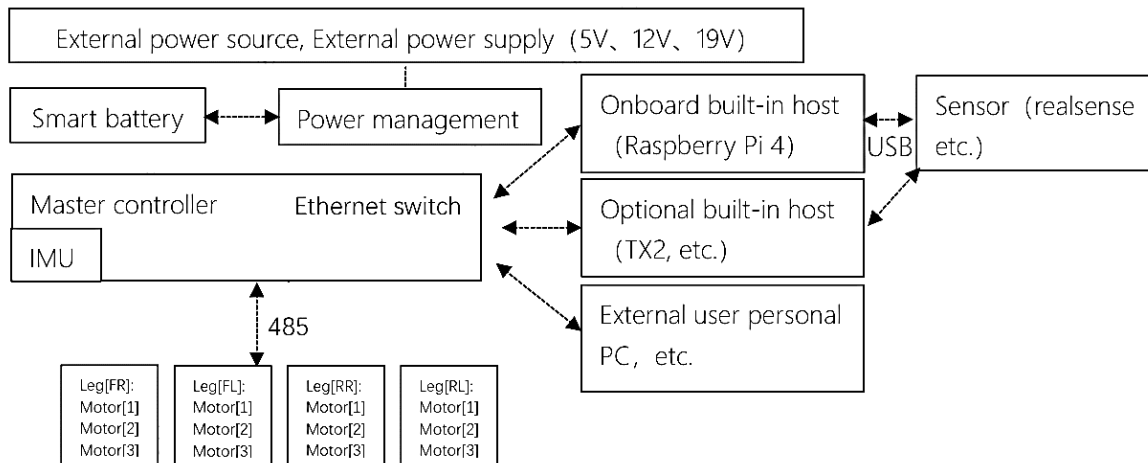


Figura 3: Modaliteti i Funkzionimit të Robotit [36].

A1 mbështet gjendjet dhe modalitetet e funksionimit të mëposhtme:

#### 3.4.1. Gjendja Statike e Qëndrimit:

Gjendja statike e qëndrimit i referohet lartësisë fillestare të pozicionit të trupit të robotit pas ndezjes, me trupin në nivel dhe pa asnjë veprim të kryer nga leva e komandimit. Në këtë gjendje, konsumi i energjisë së të gjithë pajisjes është minimal, duke mundësuar jetëgjatësinë më të madhe të baterisë.

#### 3.4.2. Modaliteti "A" (Kontroll i Qëndrimit dhe Pozicionit në 3 Aks gjatë Qëndrimit):

Kur roboti është në qëndrim, leva e komandimit mund të përdoret për të kontrolluar qëndrimin dhe pozicionin e robotit në 3 akse. Kjo përfshin lëvizje të trupit në pitch, roll, yaw, ulje dhe ngritje të vazhdueshme, ulje të pjesshme, qëndrim të vazhdueshëm, ngritje të pjesshme, dhe kombinime të ndryshme.

#### 3.4.3. Modaliteti "W" (Ecje):

Kur roboti është në modalitetin "W", ai mund të realizojë lëvizjen hap pas hapi pa manipuluar levën e komandimit; leva e komandimit mund të përdoret për të realizuar lëvizjen përpara dhe mbrapa, zhvendosjen në anë, rrotullimin në vend dhe ecjen sipas disa rregullave në terren të sheshtë (vijë e drejtë, rreth, hark, drejtkëndësh), zvarritje përpara, ecje në pjerrësi ose shkallë, dhe rikuperimin e ekuilibrit në shkallë të gjerë. Gjithashtu ka një përshtatshmëri të lartë në terrene të parregullta, me një shpejtësi maksimale ecjeje prej 3.3 m/s.

#### **3.4.4. Modaliteti "Sport":**

Kur roboti është në këmbë (në modalitetin "A"), duke mbajtur të shtypur butonin L2 dhe duke klikuar një herë butonin B, roboti ulet dhe kalon në modalitetin e amortizimit. Më pas, duke mbajtur shtypur butonin L1 dhe duke klikuar një herë butonin START, roboti hyn në modalitetin "Sport". Do të ketë një zhurmë të dukshme të rrymës dhe roboti do të ngrihet sërish. Duke liruar butonin L1 dhe duke klikuar një herë butonin START, roboti hyn në modalitetin "Sport". Në këtë modalitet, roboti ka një përshtatshmëri të lartë ndaj terreneve të parregullta, me një shpejtësi maksimale lëvizjeje prej 3.3 m/s.

### **3.5. Modaliteti i mbrojtjes**

#### **3.5.1. Mbrojtja nga rënia**

Kur roboti është nën kontroll të nivelit të lartë dhe përballet me një rënie ose ngecje gjatë qëndrimit ose ecjes, ai kalon në modalitetin e mbrojtjes: Të gjitha nyjet kalojnë në modalitetin e amortizimit të plotë dhe ndizen dritat LED të kuqe për sinjalizim.

#### **3.5.2. Mbrojtja nga shkëputja**

Mbrojtja nga shkëputja, e njohur gjithashtu si mbrojtja nga humbja e vazhdueshme e paketave, është një reagim ndaj komunikimit të dobët. Shkaqet e mundshme përfshijnë: rrjet të paqëndrueshëm, transferim masiv të të dhënave përmes USB-së, ndërprerje nga GPU-ja, etj.

Nëse humbet një paketë, por kjo nuk zgjat më shumë se 30 milisekonda, si në rastin e humbjes së një ose më shumë paketave, roboti do të vazhdojë të funksionojë sipas komandës së fundit të marrë. Nëse nuk merren komanda për më shumë se 30 milisekonda, roboti do të kalojë në modalitetin e mbrojtjes nga shkëputja derisa të vijë një komandë e re.

→ Shkëputja në nivel të lartë: Roboti do të kalojë në modalitetin e qëndrimit dhe të gjitha komandat e nivelit të lartë do të humbasin efektivitetin.

→ Shkëputja në nivel të ulët: Nyjet do të kalojnë në modalitetin e frenimit elektronik.

#### **3.5.3. Frenimi Emergjent**

Në rast të ndonjë aksidenti, duke shtypur butonin OFF (për 1.5 sekonda) përmes telekomandës emergjente, roboti do të ndalojë të gjitha veprimet dhe do të fiket. Për më shumë detaje, referojuni manualit të përdoruesit.

#### **3.5.4. Motori**

Komandat (MotorCmd) dhe gjendja (MotorState) që lidhen me motorin janë pas reduktuesit përkatës, prandaj kontrolli i motorit mund të konsiderohet ekuivalent me kontrollin e nyjeve. Në këtë mënyrë, raporti i reduktimit nuk ka nevojë të merret parasys.

#### **3.5.5. Kujdeset për Motorin**

Përdoruesi duhet t'i kushtojë vëmendje fenomenit të nxehtësisë gjatë përdorimit të motorit. Dy sensorë termikë janë ndërtuar në motor për të monitoruar temperaturën në kohë reale. Bazuar në dy formulat e mëposhtme:

→ Sipas Ligjit të Joule:  $Q = I^2Rt$  (Q: nxehtësia, I: rryma, R: rezistenca, t: koha)

→ Raporti mes momentit të forcës dhe rrymës:  $T = K \cdot I$  (T: momenti, I: rryma, K: koeficienti i momentit)

Arrijmë në përfundimin se momenti i forcës ka një marrëdhënie katrorike me rrymën e daljes së motorit, prandaj kur kërkohet një moment më i madh, prodhohet më shumë nxehtësi. Në periudha të shkurtra kohore, për shkak të kapacitetit specifik të nxehtësisë së motorit, prodhimi i momentit të lartë në mënyrë të menjëhershme nuk do të rrisë ndjeshëm temperaturën e motorit. Megjithatë, për shkak të ngadalësisë së përçimit të nxehtësisë (rezistenca termike), ndryshimet në temperaturë të monitoruara nga sensori do të kenë një vonesë të dukshme. Kur temperatura e çdo sensori termik e monitoruar nga bordi i drejtimit të motorit kalon  $60^\circ\text{C}$ , motori do të fiket automatikisht deri sa temperatura të bjerë në një nivel të pranueshëm për t'u riaktivizuar.

## 4. Kinematika e robotit A1

Kinematika është një degë e fizikës që studion lëvizjen e trupave pa marrë parasysh forcat që shkaktojnë këtë lëvizje. Ajo përqendrohet në përshkrimin e trajektores së trupave, shpejtësinë dhe përshpejtimin e tyre. Elementet kryesore që studion kinematika përfshijnë:

- Pozicioni: Vendndodhja e një trupi në hapësirë në një moment të caktuar.
- Shpejtësia: Shpejtësia me të cilën ndryshon pozicioni i një trupi me kalimin e kohës, e cila mund të jetë konstante ose ndryshuese.
- Përshpejtimi: Shpejtësia me të cilën ndryshon shpejtësia e një trupi me kalimin e kohës.

Kinematika është një komponent thelbësor për të kuptuar mekanikën dhe luan një rol thelbësor në inxhinieri, pasi ajo ndihmon inxhinierët të analizojnë dhe projektojnë sisteme të ndryshme mekanike dhe struktura. Në robotikë, kinematika përdoret për të planifikuar dhe kontrolluar lëvizjet e robotëve. Analiza kinematike lejon përcaktimin e trajektores së robotit dhe koordinimin e lëvizjeve të tij për të kryer detyra specifike. Në përgjithësi, kinematika është një mjet i fuqishëm për inxhinierët, duke u ofruar atyre mjetet e nevojshme për të analizuar, dizajnuar dhe përmirësuar sisteme të ndryshme mekanike dhe struktura. Kjo kontribuon në zhvillimin e teknologjive të reja dhe përmirësimin e sigurisë dhe efikasitetit të pajisjeve dhe strukturave inxhinierike.

Kinematika direkte dhe inverse janë dy aspekte kyçe të analizës së performancës së një roboti. Në rastin e robotit A1 të prodhuar nga kompania Unitree, studimi i këtyre aspekteve është thelbësor për të kuptuar dhe kontrolluar lëvizjet dhe pozicionet e robotit në hapësirë.

Kinematika direkte është procesi i llogaritjes së pozicionit dhe orientimit të fundit të robotit bazuar në vlerat e zgjedhura të këndit të lidhur me motorët. Përdorimi i transformimeve matematikore, siç janë transformimet homogjene, lejon përcaktimin e pozicionit dhe orientimit të fundit të robotit në lidhje me sistemin e referencës së përdorur. Në anën tjetër, kinematika inverse është procesi i gjetjes së vlerave të nevojshme të këndit të motorëve për të arritur një pozicion ose orientim të caktuar të fundit për robotin. Ky proces është shumë i rëndësishëm në kontrollin e precizionit të lëvizjeve të robotit, pasi lejon drejtimin e robotit drejt një pozicioni ose orientimi të caktuar në hapësirë.

Për të kuptuar në mënyrë të plotë kinematikën drejtpërdrejtë dhe inverse të robotit A1 nga Unitree, është e rëndësishme të analizohen me detaje karakteristikat fizike të strukturës së robotit, si dhe lidhjet dhe lëvizjet e tyre në raport me një sistem koordinatash të caktuar. Në këtë mënyrë, mund të zhvillohen modele matematikore dhe algoritme të përshtatshme për të llogaritur pozicionet dhe orientimet e robotit në një mënyrë efikase dhe të saktë.

Në vazhdim do të analizojmë kinematikën e robotit katërkëmbësh që është një sistem robotik që përbëhet nga një trup i ngurtë dhe katër këmbë ku secila ka tre shkallë lirie, këmbët kanë të njëjtën strukturë.



#### 4.1. Kinematika inverse

Robotët katërkëmbësh, siç është roboti Unitree A1 në rastin tonë, mund të kenë konfigurime të ndryshme në varësi të koordinatave të këmbëve. Për këtë arsye, studiohet ekuacioni kinematik midis lëvizjeve rrotulluese rreth qendrës së sistemit koordinativ të trupit dhe sistemit të koordinatave të çdo pike fundore të këmbës. Fillimisht, për të përcaktuar pozicionin dhe orientimin e qendrës së trupit të robotit në hapësirën e punës, matrica e transformimit merret duke përdorur matricat e rrotullimit.

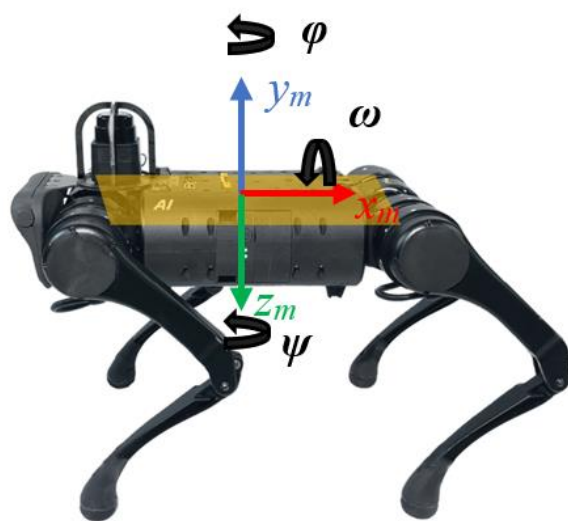


Figura 4: Modeli kinematik

Matricat e rrotullimit për akset  $x$ ,  $y$ ,  $z$  janë si më poshtë:

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\omega) & -\sin(\omega) & 0 \\ 0 & \sin(\omega) & \cos(\omega) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots\dots(1)$$

$$R_y = \begin{bmatrix} \cos(\varphi) & 0 & \sin(\varphi) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\varphi) & 0 & \cos(\varphi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots\dots(2)$$

$$R_z = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 & 0 \\ \sin(\psi) & \cos(\psi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots\dots(3)$$

$$R_{xyz} = R_x R_y R_z \dots\dots\dots(4)$$

Matrica e transformimit është paraqitur në ekuacioni (5).

$$T_M = R_{xyz} \begin{bmatrix} 1 & 0 & 0 & x_m \\ 0 & 1 & 0 & y_m \\ 0 & 0 & 1 & z_m \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots\dots(5)$$

**Llogaritja e këndeve të nyjeve**

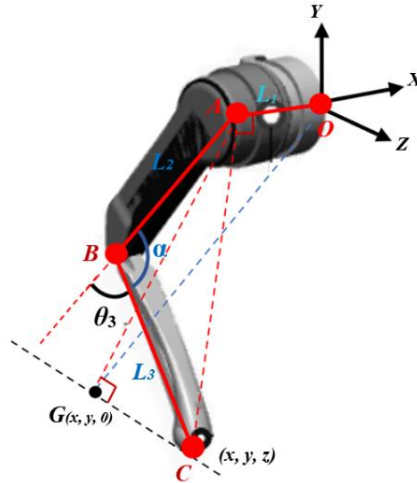


Figura 5: Përkufizimi i modelit të kinematikës dhe parametrave të këmbës robotike

Distanca ndërmjet origjinës dhe pikës së kontaktit të këmbës në rrafshin xy:

$$OG^2 = x^2 + y^2 \dots\dots\dots(6)$$

OG gjithashtu mund të shpjegohet si:

$$OG^2 = OA^2 + AG^2 \dots\dots\dots(7)$$

Duke përdorur (6) dhe (7) :

$$x^2 + y^2 = L_1^2 + AG^2$$

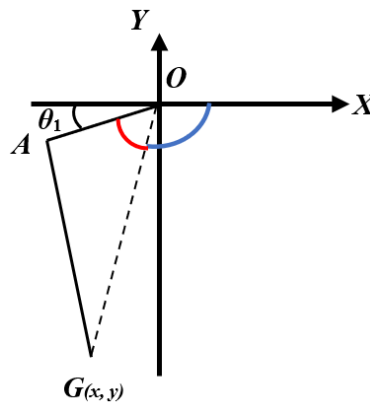


Figura 6: Vijushmëria në OX

AG shprehet si :

$$AG = \sqrt{x^2 + y^2 - L_1^2}$$

AC shprehet si :

$$AC = \sqrt{AG^2 + GC^2}$$

$\theta_1$  shprehet si :

$$\theta_1 = \pi - \left( \tan^{-1} \left( \frac{y}{x} \right) + \tan^{-1} \left( \frac{AG}{OA} \right) \right)$$

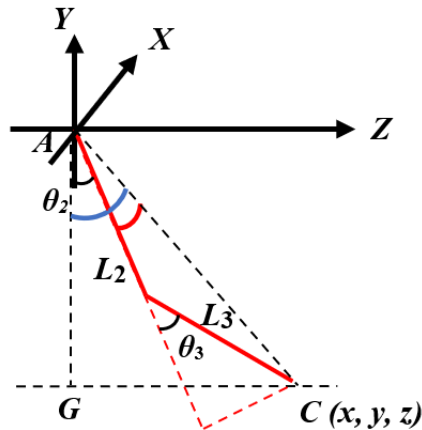


Figura 7: Vijushmëria në YZ

Përdorim ligjit e kosinuset :

$$AC^2 = L_2^2 + L_3^2 - 2L_2L_3 \cos(\alpha)$$

$\alpha$  shprehet si :

$$\alpha = \cos^{-1} \left( -\frac{AC^2 - L_2^2 - L_3^2}{2L_2L_3} \right)$$

$\theta_3$  shprehet si

$$\theta_3 = \pi - \alpha$$

$\theta_2$  shprehet si :

$$\theta_2 = \tan^{-1} \left( \frac{GC}{AG} \right) - \tan^{-1} \left( \frac{L_3 \sin \theta_3}{L_2 + L_3 \cos \theta_3} \right)$$

## 4.2. Kinematika direkte

Kinematika direkte për robotët katërkëmbësh merret me përcaktimin e pozicionit dhe orientimit të përfundimtarëve të robotit që zakonisht janë këmbët duke dhënë parametrat e nyjeve. Konventa Denavit-Hartenberg (DH) është një metodë sistematike për t'u caktuar korniza koordinatave nyjeve të robotit për një analizë më të lehtë kinematike.

Hapat më poshtë tregojnë mënyren për aplikimin e parametrave të DH për një robot me katër këmbë:

- **Identifikimi i nyjeve dhe lidhjeve:** Ku për një robotë me katër këmbë zakonisht ka nyje të shumta dhe lidhje.
- **Përcaktimi i parametrave DH:** Ku për çdo lidhje  $i$ , përcaktohen parametrat DH:
  - $a_i$ : Gjatësia e lidhjes, distanca midis  $z_i$  dhe  $z_{i+1}$  përgjatë  $x_i$ .
  - $\alpha_i$ : Këndi i lidhjes, kënd midis  $z_i$  dhe  $z_{i+1}$  rreth  $x_i$ .
  - $d_i$ : Kompensimi i lidhjes, distanca midis  $x_i$  dhe  $x_{i+1}$  përgjatë  $z_i$ .
  - $\theta_i$ : Këndi i nyjes kënd midis  $x_i$  dhe  $x_{i+1}$  rreth  $z_i$ .
- **Krijohet matrica e transformimit:** Ku për çdo nyje, krijohet matricën e transformimit homogjen  $A_i$  duke përdorur parametrat DH:

$$A_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \cos(\alpha_i) & \sin(\theta_i) \sin(\alpha_i) & a_i \cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \cos(\alpha_i) & -\cos(\theta_i) \sin(\alpha_i) & a_i \sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- **Llogaritet kinematika direkte:** Ku shumëzohen matricat e transformimit nga korniza bazë te korniza e përfundimtarit. Për një këmbë me  $n$  nyje:

$$T = A_1 A_2 \dots A_n$$

Matrica rezultuese  $T$  jep pozicionin dhe orientimin e efektorit fundor në kornizën bazë.

Në vazhdim do të paraqesim parametrat Denavit-Hartenberg (DH) për kinematikën direkte të këmbës së robotit për rastin tonë, roboti unitree A1.

**Tabela 1:** Parametrat e Denavit-Hartenberg

Lidhjet	$a_{i-1}$	$\alpha_{i-1}$	$d_i$	$\theta_i$
0-1	$L_1$	0	0	$\theta_1$
1-2	0	$-\pi/2$	0	$-\pi/2$
2-3	$L_2$	0	0	$\theta_2$
3-4	$L_3$	0	0	$\theta_3$

Çdo matrice transformimi jepet në ekuacionet e më poshtëme dhe matrica kinematike direkte është ekuacioni 12.

$$A_0^1 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & -L_1 \cos(\theta_1) \\ \sin(\theta_1) & \cos(\theta_1) & 0 & -L_1 \sin(\theta_1) \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots \dots \dots (8)$$

$$A_1^2 = \begin{bmatrix} 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots \dots \dots (9)$$

$$A_2^3 = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & L_2 \cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & L_2 \sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots \dots \dots (10)$$

$$A_3^4 = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & L_3 \cos(\theta_3) \\ \sin(\theta_3) & \cos(\theta_3) & 0 & L_3 \sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots \dots \dots (11)$$

$$T = A_0^1 A_1^2 A_2^3 A_3^4 = \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \\ t_{41} & t_{42} & t_{43} & t_{44} \end{bmatrix} \dots \dots \dots (12)$$

Elementet e matricës kinematike direkte të një këmbë të robotit janë dhënë në tabelën 2

**Tabela 2 :** Elementet e matricës kinematike direkte

$t_{11}$	$\cos(\theta_2) \cos(\theta_3) \sin(\theta_1) - \sin(\theta_1) \sin(\theta_2) \sin(\theta_3)$
$t_{12}$	$-\cos(\theta_2) \sin(\theta_1) \sin(\theta_3) - \cos(\theta_3) \sin(\theta_1) \sin(\theta_2)$
$t_{13}$	$-\cos(\theta_1)$
$t_{14}$	$L_2 \cos(\theta_2) \sin(\theta_1) - L_1 \cos(\theta_1) + L_2 \cos(\theta_2) \cos(\theta_3) \sin(\theta_1) - L_3 \sin(\theta_1) \sin(\theta_2) \sin(\theta_3)$
$t_{21}$	$\cos(\theta_1) \sin(\theta_2) \sin(\theta_3) - \cos(\theta_1) \cos(\theta_2) \cos(\theta_3)$
$t_{22}$	$\cos(\theta_1) \cos(\theta_2) \sin(\theta_3) + \cos(\theta_1) \cos(\theta_3) \sin(\theta_2)$
$t_{23}$	$-\sin(\theta_1)$
$t_{24}$	$L_3 \cos(1) \sin(\theta_2) \sin(\theta_3) - L_2 \cos(\theta_1) \cos(\theta_2) - L_3 \cos(\theta_1) \cos(\theta_2) \cos(\theta_3) - L_1 \sin(\theta_1)$
$t_{31}$	$\cos(\theta_2) \sin(\theta_3) + \cos(\theta_3) \sin(\theta_2)$
$t_{32}$	$\cos(2) \cos(\theta_3) - \sin(\theta_2) \sin(\theta_3)$

$t_{33}$	0
$t_{34}$	$L_2 \sin(\theta_2) + L_3 \cos(\theta_2) \sin(\theta_3) + L_3 \cos(\theta_3) \sin(\theta_2)$
$t_{41} = t_{42} = t_{43}$	0
$t_{44}$	1

### 4.3. Kinematika e Robotit A1 Unitree përmes MATLAB-it

Në këtë punim, ne do të analizojmë kinematikën e robotit A1 Unitree, duke përdorur MATLAB për të gjeneruar dhe studiuar lakoret që përshkruajnë sjelljen e robotit gjatë një periudhe të caktuar kohore. Në veçanti, do të fokusohemi në analizimin e ndryshimeve në këndet e nyjeve, pozicionin e robotit, si dhe lëvizjen e qendrës së masës (CoM) gjatë një trajektorie të paracaktuar.

Fillimisht, kemi përdorur algoritme të kinematikës për të përcaktuar pozitat e këmbëve të robotit në bazë të koordinatave kartesiane. MATLAB-i na ka ofruar mjetet e duhura për të vizualizuar këto pozita në kohë reale, duke na mundësuar të kuptojmë më mirë sjelljen e secilës këmbë gjatë lëvizjes së robotit. Ky proces është thelbësor për sigurimin e qëndrueshmërisë së robotit dhe për kryerjen e lëvizjeve të sakta dhe të koordinuara.

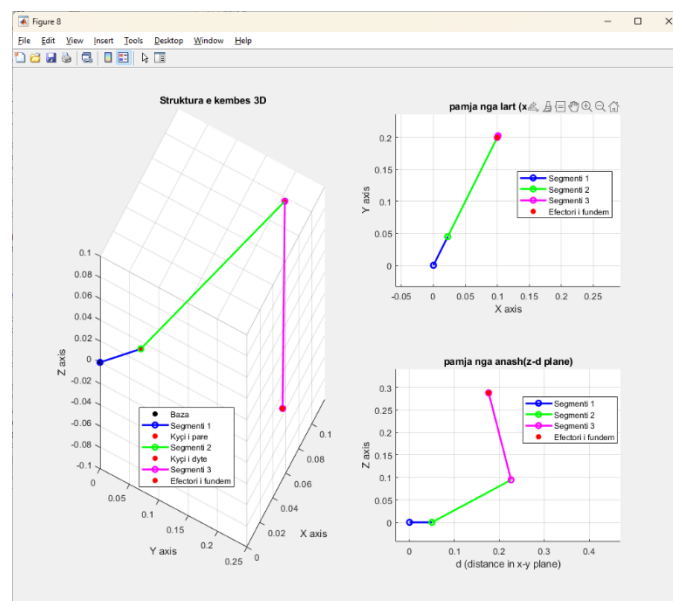


Figura 8: Pozicioni i këmbëve të robotit në 3D si dhe pamjet nga vështrimet te ndryshme nga lart dhe anash.

Këto lakore tregojnë sesi këndet e nyjeve të robotit ndryshojnë me kalimin e kohës gjatë ekzekutimit të një trajektorie specifike. Çdo këmbë e robotit është e pajisur me tre nyje kryesore: nyja e kofshës, nyja e gjurit dhe nyja e kyçit të këmbës. Lakoret që kemi gjeneruar tregojnë sjelljen e këtyre nyjeve në përgjigje të kompozimeve të ndryshme të komandave lëvizëse, duke na dhënë një pasqyrë të qartë mbi mënyrën sesi robotit menaxhon ekuilibrin dhe lëvizjen e tij.

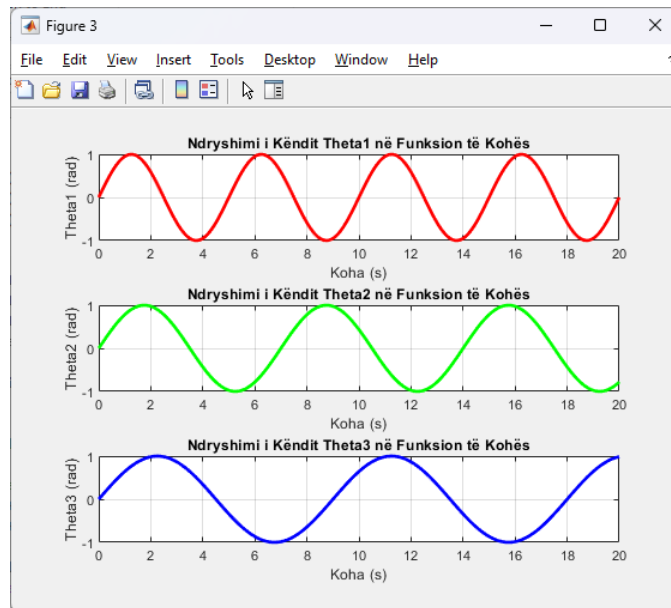


Figura 9: Paraqitja e lakoreve të cilat tregojnë ndryshimin e këndeve të nyjeve për kohën prej 20 sekondash.

Përveç këndeve të nyjeve, është e rëndësishme të analizojmë edhe trajektorinë e robotit në hapësirë. Kemi krijuar lakoret që përshkruajnë ndryshimin e pozitës së përgjithshme të robotit gjatë lëvizjes. Këto lakore janë të rëndësishme për të kuptuar si lëvizet roboti në hapësirë, duke marrë parasysh të gjithë strukturën e tij. Me ndihmën e këtyre lakoreve, mund të analizojmë dhe optimizojmë trajektoren e robotit për të arritur qëllime të ndryshme, si ecja, kthimi, apo kryerja e detyrave të tjera dinamike.

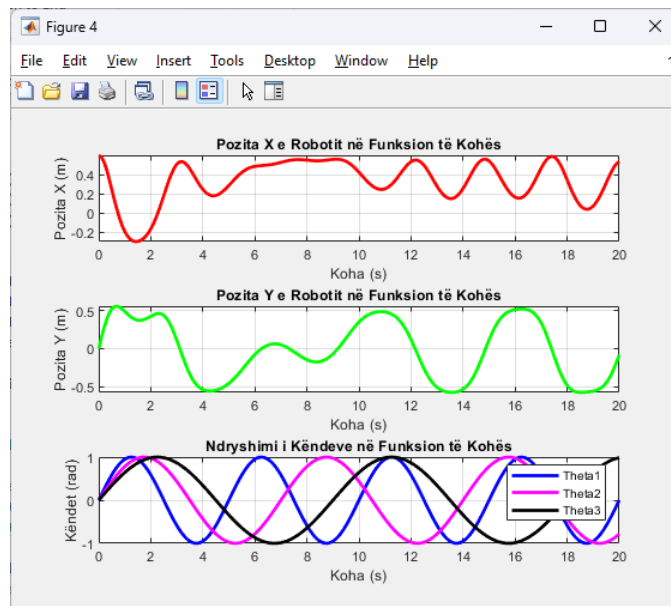


Figura 10: Ndryshimi i pozitës së robotit për gjatë X dhe Y është paraqitur me anë të këtyre lakoreve si dhe ndryshimi i tre këndeve në një vend.

Qendra e masës së rëndësës është një element kyç në ruajtjen e stabilitetit të robotit, veçanërisht gjatë lëvizjeve komplekse apo në terrene të pabarabarta. Lakoret që paraqesin lëvizjen e qendres së masës së rëndësës tregojnë sesi ndryshon pozicioni i qendrës së masës së rëndësës



së robotit gjatë kohës. Kjo analizë është kritike për të vlerësuar aftësinë e robotit për të ruajtur balancën dhe për të parandaluar rrokullisjen gjatë lëvizjeve të shpejta apo të papritura.

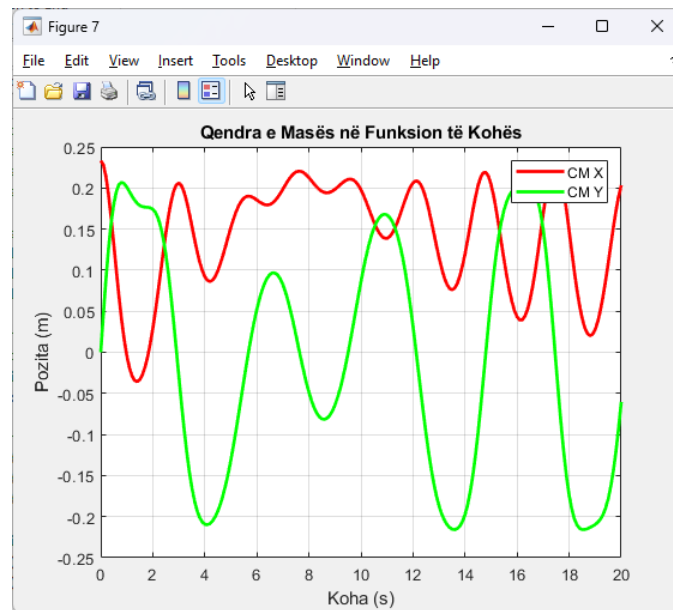


Figura 11: Paraqitja e lakoreve se si ndryshon qendra e masës së rëndësës në funksion të kohës

Analizat e kryera përmes MATLAB na kanë ofruar një pasqyrë të thelluar mbi sjelljen kinematike të robotit A1 Unitree. Lakoret e gjeneruara na kanë ndihmuar të kuptojmë më mirë se si roboti menaxhon lëvizjet e tij në një hapësirë të caktuar dhe si i përgjigjet ndryshimeve në komandat lëvizëse. Këto rezultate janë të rëndësishme për zhvillimin e mëtejshëm të algoritmeve të kontrollit dhe për përmirësimin e performancës së përgjithshme të robotit në aplikime praktike.

## **5. Sistemi operativ i robotëve (ang. Robot operating system (ROS))**

ROS (Robot Operating System) është një sistem meta-operativ me burim të hapur që ofron një gamë të gjerë shërbimesh për robotët tuaj. Kjo platformë siguron abstraksionin e harduerit, kontrollin e pajisjeve në nivel të ulët, kalimin e mesazheve midis proceseve dhe menaxhimin e paketave. Përveç kësaj, ROS furnizon mjete dhe biblioteka të përshtatshme për zhvillimin, shkrimin dhe ekzekutimin e kodit nëpër kompjuterë të shumtë.

Sistemi Operativ i Robotëve (ROS) është një kornizë fleksibël dhe e fuqishme e krijuar për zhvillimin e softuerit robotikë. Pavarësisht nga emri i tij, ROS nuk është një sistem operativ në kuptimin tradicional; përkundrazi, është një program i mesëm që funksionon në krye të një sistemi operativ konvencional si Linux. Ai siguron një sërë bibliotekash dhe mjetesh që ndihmojnë zhvilluesit të krijojnë aplikacione komplekse dhe të fuqishme robotësh.

ROS është i ngjashëm me 'kornizat e robotëve' të tjera, duke përfshirë Player, YARP, Orocos, dhe të tjera, por ndryshon në mënyrën sesi implementon komunikimin mes komponenteve të sistemit. Për shembull, ROS përdor stile të ndryshme komunikimi si komunikimi sinkron mbi shërbimet RPC, transmetimi asinkron mbi tema dhe ruajtjen e të dhënave në një server parametrash. Megjithatë, është e rëndësishme të theksohet se ROS nuk është një kornizë në kohë reale në origjinën e saj, megjithatë ka mundësi për të integruar ROS me kodin që punon në kohë reale.

ROS (Robot Operating System) është një platformë shpërndarë që ka si qëllim kryesor mbështetjen e ripërdorimit të kodit në kërkimin dhe zhvillimin e robotikës. Për dallim nga kornizat e tjera softuerike robotike, ROS është projektuar për të lejuar ekzekutuesit (Nodes) të dizajnohen dhe të lidhen lirshëm në kohën e ekzekutimit. Këto Nodes mund të grupohen në Paketa dhe Stacks, të cilat mund të ndahen dhe shpërndahen lehtësisht. ROS gjithashtu përdor një sistem federuar të depove të kodit që mundëson shpërndarjen dhe bashkëpunimin e kodit në një komunitet të gjerë [31]

### **5.1. Arkitektura ROS**

Arkitektura ROS përbëhet nga disa komponentë kyç që punojnë së bashku për të mundësuar zhvillimin dhe ekzekutimin e aplikacioneve robotike

#### **❖ Nyjet**

Nyjet janë blloqet bazë të ndërtimit të një sistemi ROS. Një nyje është një ekzekutues që kryen llogaritjen. Mund të jetë një drejtues sensori, një algoritëm kontrolli, një proces simulimi ose ndonjë komponent tjetër softuerësh.

Nyjet regjistrohen në ROS Master për t'i bërë të njohur sistemit. ROS Master është një mekanizëm koordinues i centralizuar që ndihmon nyjet të zbulojnë njëri-tjetrin dhe të krijojnë komunikim.

#### **❖ Master: (thelbi i një programi ROS)**

ROS Master është një sistem koordinimi i centralizuar që lehtëson komunikimin ndërmjet nyjeve. Mban gjurmët e nyjeve aktive, menaxhon regjistrimin dhe zbulimin e nyjeve dhe ndihmon në krijimin e lidhjeve ndërmjet tyre.

Pasi të përfundojë procesi i regjistrimit dhe zbulimit, nyjet mund të komunikojnë drejtpërdrejt me njëri-tjetrin duke përdorur informacionin e marrë nga Masteri ROS.

Në ROS1, Master konsiderohet problematike dhe pika e vetme e dështimit.

#### ❖ Temat

Temat quhen kanale për komunikim ndërmjet nyjeve. Nyjet mund të publikojnë mesazhe për tema dhe nyjet e tjera mund të abonohen në ato tema për të marrë mesazhet.

#### ❖ Mesazhet

Mesazhet janë strukturat e të dhënave që përdoren për komunikimin midis nyjeve mbi tema. Mesazhet përcaktojnë formatin dhe përmbajtjen e të dhënave që shkëmbehen.

#### ❖ Biblioteka e klientit

ROS mbështet shumë gjuhë programimi dhe secila gjuhë ka bibliotekën e vet të klientit. Dy gjuhët më të përdorura në zhvillimin e ROS janë C++ dhe Python.

#### ❖ TCPROS dhe UDPROS

shërbejnë si protokolle komunikimi të përdorura për të transportuar të dhëna ndërmjet nyjeve ROS. Këto protokolle vendosin procedurat për serializimin, transmetimin dhe deserializimin e mesazheve nëpër lidhjet e rrjetit.

**Për të filluar zbatimin e një shembulli të tillë në ros, fillimisht duhet të kuptojmë se si është strukturuar hapësira jonë e punës:**

Struktura tipike e hapësirës së punës organizohet duke përdorur sistemin e ndërtimit Catkin i cili gjeneron një direktori që përmban një ose më shumë paketa ROS dhe përdor sistemin e ndërtimit Catkin për të ndërtuar dhe organizuar këto paketa.

Le të krijojmë hapësirën tonë të re të punës duke përdorur komandat e mëposhtme në ubuntu terminal:

```
mkdir -p catkin_ws/src
cd catkin_ws
```

sistemi i ndërtimit të catkin do të gjenerojë drejtori në strukturën e mëposhtme:

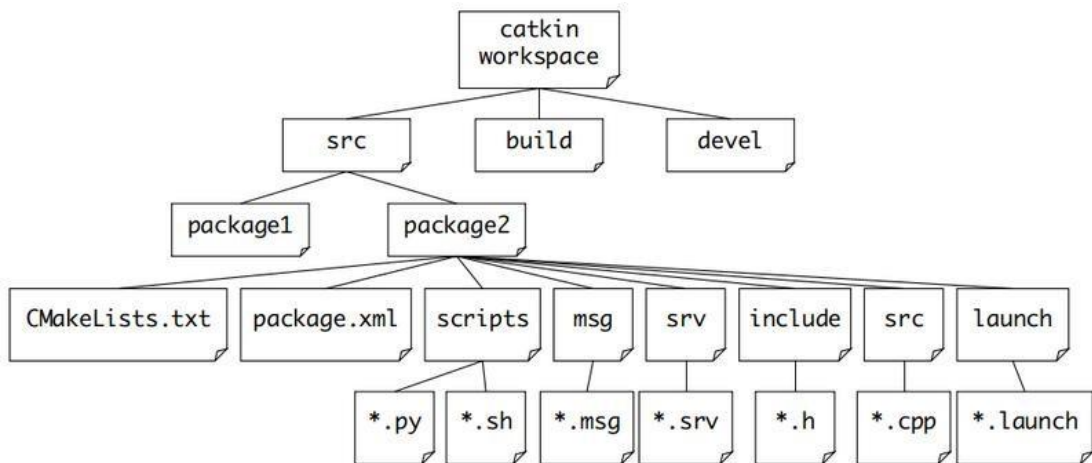


Figura 12: Struktura e Catkin\_Workspace

Drejtoria kryesore brenda një hapësire pune Catkin janë [ src, build and devel ].

- Drejtoria src:  
Kjo direktori është vendi ku vendosni paketat tuaja ROS.  
Çdo paketë është një direktori që përmban nyje ROS, biblioteka, skedarë konfigurimi dhe burime të tjera.
- Ndërtoni direktorinë:  
Ky direktori gjenerohet automatikisht nga Catkin dhe përmban objekte ndërtimi.  
Kur ndërtoni hapësirën tuaj të punës (catkin\_make), këtu ruhen binarët e përpiluar, bibliotekat dhe skedarët e tjerë të lidhur me ndërtimin.
- Drejtoria e zhvillimit:  
Gjeneruar gjithashtu nga Catkin gjatë procesit të ndërtimit.  
Ai përmban lidhje simbolike me skedarët në drejtorinë src dhe përdoret për të ekzekutuar nyjet ROS direkt nga hapësira e punës e zhvillimit [29].

## 5.2. Mjetet për Vizualizim dhe Simulim

Simulatorët në ROS ofrojnë mjete të ndryshme për zhvillimin dhe testimin e aplikimeve robotike në një mjedis virtual. Disa nga simulatorët më të njohur përfshijnë Gazebo, Stage, V-REP, Webots dhe CoppeliaSim. Secili prej tyre ka përparësi dhe kundërshtime të veta, të cilat mund të përcaktohen në bazë të nevojave dhe preferencave tuaja të specifikuara.

- Gazebo: Gazebo është një simulator i përdorur gjerësisht që ofron fizikë të besueshme, grafikë realiste dhe një gamë të gjerë modeleve dhe shtojcave. Megjithatë, mund të jetë intensiv dhe kompleks për t'u konfiguruar nga ana e performancës llogaritëse.
- Stage: Stage është një simulator i thjeshtë dhe i shpejtë, i cili mund të trajtojë një numër të madh robotësh dhe sensorësh. Ai ofron një performancë të mirë në lidhje me shpejtësinë e ekzekutimit, por ka aftësi të kufizuara grafike dhe fizike krahasuar me disa prej alternativave të tjera.
- V-REP dhe CoppeliaSim: V-REP dhe CoppeliaSim ofrojnë një gamë të gjerë të modelimit dhe simulimit të robotëve në një mjedis 3D. Ata përdorin një qasje të

përshtatshme për të modeluar detaje të avancuara fizike dhe shqyrtime vizuale, por mund të kenë një kosto më të lartë resurse dhe një krivinë më të stërpiktur për të mësuar.

- **Webots:** Webots është një opsion i lehtë për të filluar, i cili ofron një platformë tërheqëse për zhvillim të robotëve dhe simulim. Ai siguron një bashkëveprim të mirë me sensorët dhe aktuatorët, duke përdorur një gamë të gjerë modeleve të integruara dhe mjediseve të simulimit.

Për të përdorur efektivisht këto mjete në zhvillimin tuaj, ju mund të përdorni temat dhe shërbimet e ROS për të komunikuar midis nyjeve të ROS. Temat ofrojnë një mënyrë për publikimin dhe abonimin në mesazhe të specifikuar, siç janë të dhënat e sensorit ose komandat. Një shembull i përdorimit të temave është përdorimi i temës /cmd\_vel për të dërguar komandat e shpejtësisë për robotin tuaj në simulator.

Përveç kësaj, ju mund të krijoni modele dhe shtojca të personalizuara për të përshtatur nevojat dhe qëllimet tuaja specifike në simulim. Modelet mund të përshkruajnë entitete fizike si robotët dhe objektet, ndërsa shtojcat shtojnë funksionalitete shtesë për të simuluar sensorët, kontrolluesit dhe interfacat e tjera.

Në fund, përdorimi i mjeteve të vizualizimit si RViz dhe simulimit si Gazebo mund të përmirësojë përvojën tuaj në zhvillimin dhe testimin e robotëve tuaj në ROS. RViz ofron mundësi të shumta për të vizualizuar dhe ndërvepruar me të dhënat e robotit në një mjedis 3D, ndërsa Gazebo siguron një simulim të saktë fizik dhe sensorësh për zhvillim të detajuar dhe testime. Kombinimi i këtyre mjeteve mund të ofrojë një platformë të fuqishme dhe të përshtatshme për zhvillimin e aplikimeve robotike në ROS [30].

### 5.3. Instalimi i ROS

Për të instaluar ROS, ka disa hapa që duhet të ndiqen. Këtu do të paraqesim hapat specifikisht për instalimin e ROS Noetic në Ubuntu 20.04.6 LTS, versionin të cilin ne kemi përdorur në punën tonë.

[Instalimi i ROS Noetic](#) kërkon Ubuntu 20.04.

- Shkarkoni [Ubuntu 20.04](#)
- Ndiqni udhëzuesin [e instalimit të Ubuntu 20.04](#)

Pasi të keni instaluar Ubuntu 20.04 duhet ndjekur hapat e mëposhtëm

1. Hapni terminalin në Ubuntu dhe shkruani komandën e mëposhtme që është konfigurimi i listës së burimeve:

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'
```

2. Pas së cilës do të vendosni çelësat:

```
sudo apt install curl  
curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo apt-key add -
```

### 3. Pastaj, do të instaloni ROS-noetic:

```
sudo apt update
sudo apt install ros-noetic-desktop-full
```

### 4. Së fundi, ju do të shtoni mjedisin ROS në skedarin tuaj bash, çka do të thotë se sa herë që ngarkoni terminalin tuaj, ai do të gjejë automatikisht paketat e integruara ROS

```
echo "source /opt/ros/noetic/setup.bash" >> ~/.bashrc
source ~/.bashrc
```

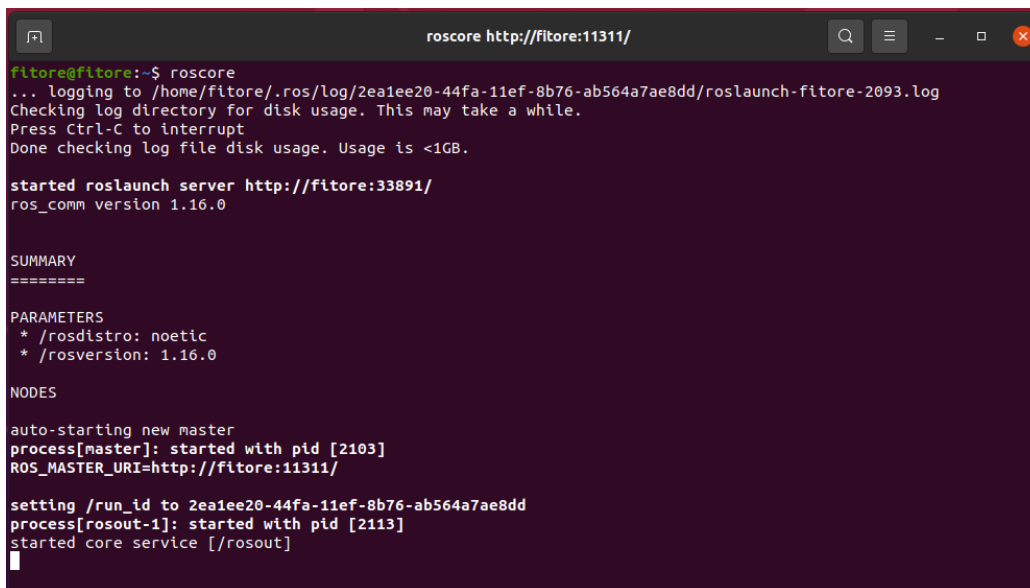
### 5. Instaloni varësitë e python dhe inicializoni repo:

```
sudo apt install python3-rosdep python3-rosinstall python3-rosinstall-generator python3-wstool build-essential
sudo rosdep init
rosdep update
```

Pasi të jetë instaluar ROS, për të testuar se a është bërë instalimi me sukses ekzekutojmë komandën si mëposhtë.

```
roscore
```

Nëse instalimi është bërë me sukses pas ekzekutimit të komandës paraprake në terminal mund të shohim rosdistro që është noetic në rastin tonë si dhe rosversionin. Përveç kësaj, gjithashtu mund të dimë që URI master ROS është në portën 11311, e cila është porti i parazgjedhur ROS. Do të jetë më e kuptueshme nëse ndjekum figurën e mëposhtme.



```
roscore http://fitore:11311/
fitore@fitore:~$ roscore
... logging to /home/fitore/.ros/log/2ea1ee20-44fa-11ef-8b76-ab564a7ae8dd/roslaunch-fitore-2093.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://fitore:33891/
ros_comm version 1.16.0

SUMMARY
=====

PARAMETERS
* /rostdistro: noetic
* /rosversion: 1.16.0

NODES

auto-starting new master
process[roscout-1]: started with pid [2113]
ROS_MASTER_URI=http://fitore:11311/

setting /run_id to 2ea1ee20-44fa-11ef-8b76-ab564a7ae8dd
process[roscout-1]: started with pid [2113]
started core service [/roscout]
```

Figura 13 : Ekzekutimi i roscore

#### 5.3.1. Simulimi

Simulimi në robotikë është procesi i krijimit të një mjedisi virtual në të cilin modelet e robotëve mund të testohen dhe vëzhgohen për të analizuar sjelljen dhe performancën e tyre. Kjo i lejon inxhinierët dhe shkencëtarët të zhvillojnë, testojnë, dhe përmirësojnë algoritmet dhe dizajnet e robotëve pa pasur nevojë të ndërtojnë dhe përdorin një robot fizik.

Disa nga arsyet kryesore për përdorimin e simulimit në robotikë janë:

- **Kostot dhe Koha:** Ndërtimi dhe testimi i një roboti fizik është shpesh i kushtueshëm dhe kërkon shumë kohë. Me simulimin, mund të bëhen ndryshime dhe të testohen në mënyrë të shpejtë dhe me kosto më të ulët.
- **Siguria:** Disa eksperimente mund të jenë të rrezikshme për t'u kryer në botën reale. Për shembull, testimi i një roboti në një mjedis të rrezikshëm mund të shkaktojë dëmtime. Simulimi ofron një mjedis të sigurt për të kryer këto teste.
- **Përsëritshmëria:** Eksperimentet në simulim mund të përsëriten shumë herë në të njëjtat kushte, gjë që e bën më të lehtë analizën e rezultateve dhe identifikimin e anomalive.
- **Mjedisi i Kontrolluar:** Në simulim, mund të krijohet një mjedis plotësisht i kontrolluar ku çdo parametër është nën kontroll, duke lejuar eksperimentime dhe testime të sakta.
- **Integrimi dhe Zhvillimi i Softuerit:** Simulimi ndihmon në zhvillimin dhe testimin e softuerit për robotët, duke lejuar inxhinierët të shohin se si funksionon softueri në kushte të ndryshme para se ta implementojnë atë në një robot real.

Për të realizuar këto simulime, përdoren mjete të tilla si Gazebo, V-REP, RViz, dhe Unity, të cilat ofrojnë mjedise të pasura për simulimin e robotëve dhe mjediseve të tyre të operimit.

### 5.3.2. Lansimi i Gazebo dhe Rviz

Për të lansuar Gazebo apo Rviz fillimisht duhet të ekzekutojmë `roslaunch` e cila do të shoqrohet me emërtimet e paketave në të cilat janë vendodhja e launch dosjeve.

Për të lansuar Gazebo komanda që duhet të ekzekutohet në rastin tonë është:

```
roslaunch unitree_gazebo robot_simulation.launch rname:=a1 wname:=earth
```

Pas një kohe të shkurtë do të hapet Gazebo si mëposhtë:

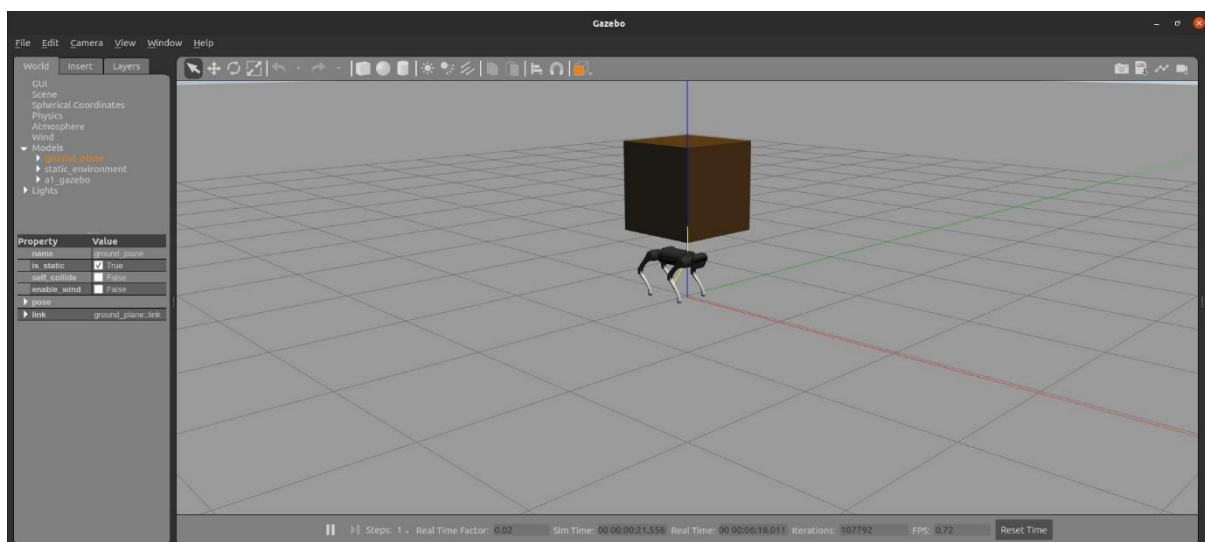


Figura 14: Lansimi i Gazebos

Në mënyrë të ngjashme bëhet edhe lansimi i Rviz

```
roslaunch a1_description a1_rviz.launch
```

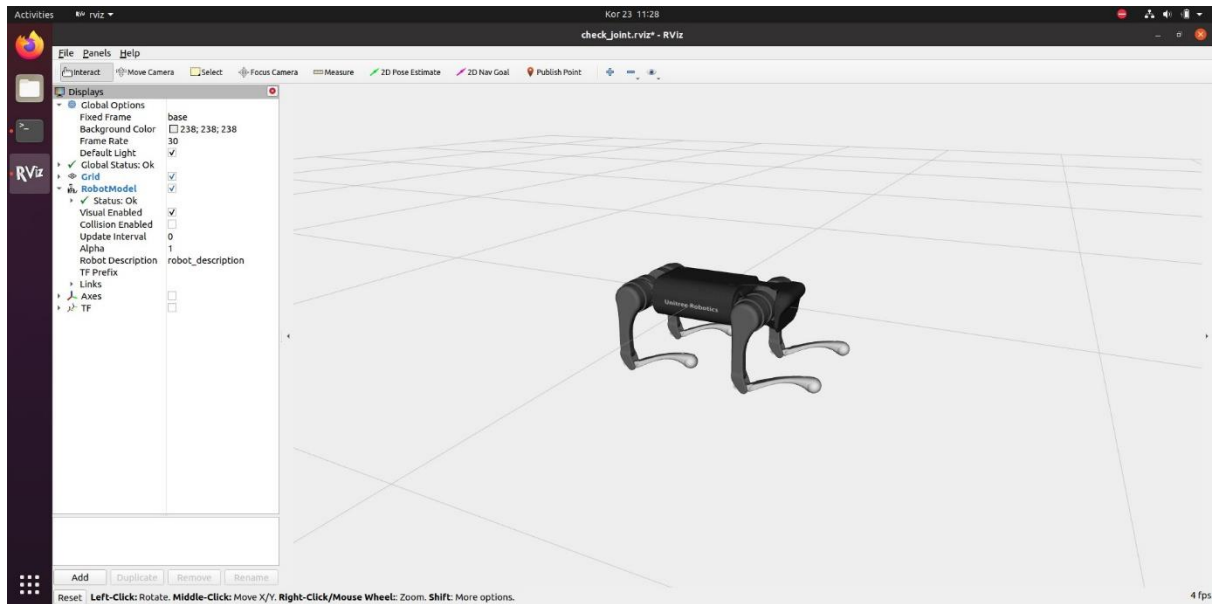


Figura 15: Lansimi i Rviz

### 5.3.3. Simulimi në Gazebo

Për të realizuar simulimin për rastin tonë, fillimisht hapet një terminal dhe niset mjedisi i simulimit Gazebo duke përdorur komandën e specifikuar më sipër për lansimin e Gazebo-s. Pasi të hapet Gazebo, për të aktivizuar kontrolluesin, hapet një terminal i ri dhe ekzekutohet komanda e mëposhtme:

```
roslaunch unitree_guide junior_ctrl
```

Pas ndezjes së kontrolluesit, roboti do të pozicionohet i shtrirë në tokën e simulatorit, siç tregohet në figurën mëposhtë.

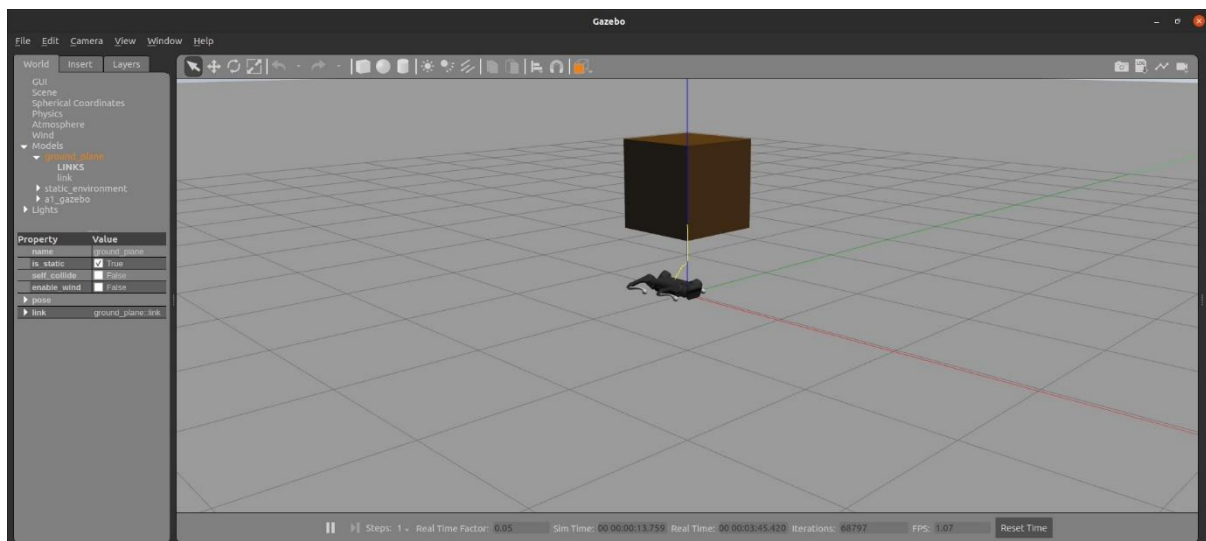
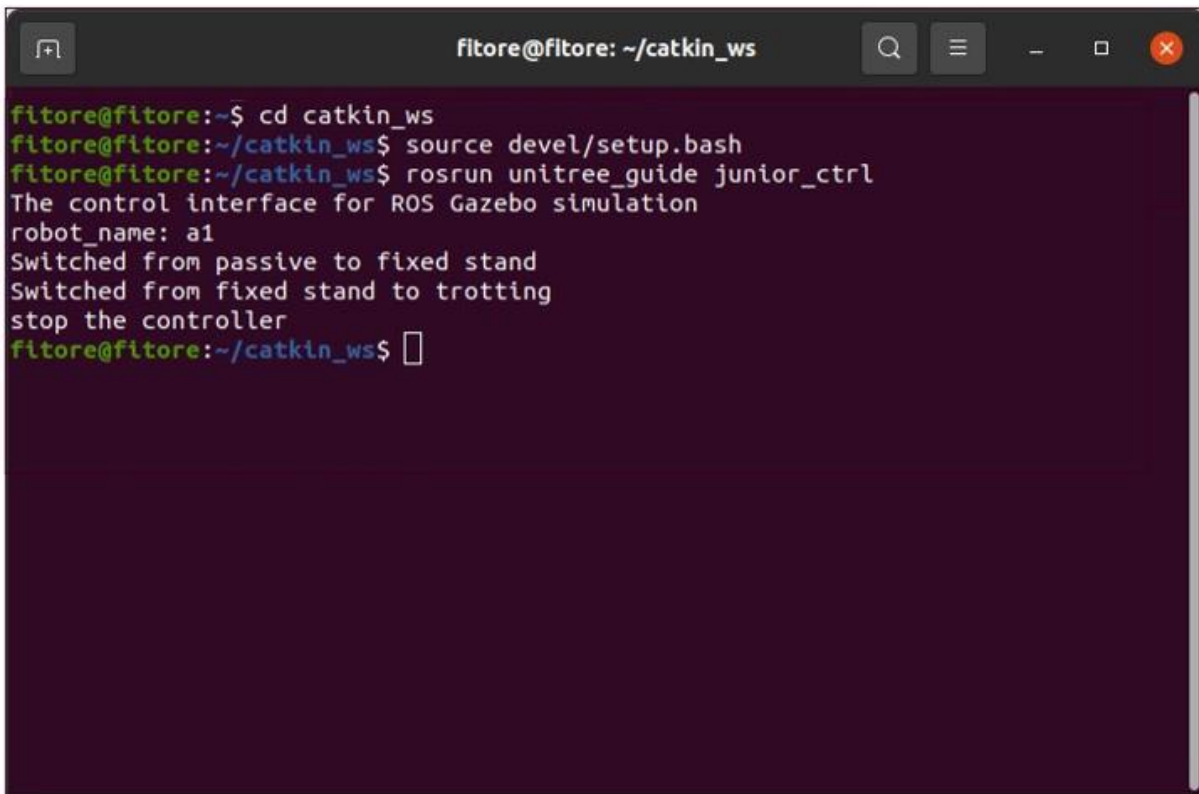


Figura 16: Shtrirja e robotit në tokën e simulimit pas ekzekutimit të kontrolluesit



Duke shtypur tastin '2' në tastierë, roboti do të kalojë nga gjendja pasive në gjendjen e qëndrimit të fiksuar (Fixed Stand). Pasi të ngrihet në këmbë, duke shtypur tastin '4', roboti kalon nga gjendja Fixed Stand në ecje të ritmit të shpejtë (Trotting).

Në figurën e mëposhtme, tregohet se si në terminal shfaqet gjendja e robotit pas shtypjes së tasteve '2' dhe '4', që shkaktojnë kalimin nga gjendja pasive në Fixed Stand dhe më pas në Trotting. Pasi të përfundojë ky proces, shtypja e kombinimit të tasteve `Ctrl+C` ndalon ekzekutimin e kontrolluesit.



```
fitore@fitore: ~/$ cd catkin_ws
fitore@fitore:~/catkin_ws$ source devel/setup.bash
fitore@fitore:~/catkin_ws$ rosrunc unitree_guide junior_ctrl
The control interface for ROS Gazebo simulation
robot_name: a1
Switched from passive to fixed stand
Switched from fixed stand to trotting
stop the controller
fitore@fitore:~/catkin_ws$
```

Figura 17: Terminali për ekzekutimin e kontrolluesit si dhe paraqitja e ndryshimeve të modeve punuese

Për të lëvizur robotin dhe për të kontrolluar kthimin e tij, përdorni tastet 'W', 'A', 'S', dhe 'D' në tastierë. Për të kontrolluar rrotullimin e robotit, përdorni tastet 'J' dhe 'L', ndërsa për ta ndaluar robotin, shtypni tastin 'Spacebar'.

Përveç këtyre, ekziston një listë e kalimeve të gjendjeve të robotit. Lista e plotë e kalimeve midis gjendjeve është si vijon:

- Tasti '1': Nga FixedStand/FreeStand në Pasive
- Tasti '2': Nga Pasive/Trotting në FixedStand
- Tasti '3': Nga FixedStand në FreeStand
- Tasti '4': Nga FixedStand/FreeStand në Trotting
- Tasti '5': Nga FixedStand në MoveBase
- Tasti '8': Nga FixedStand në StepTest
- Tasti '9': Nga FixedStand në SwingTest
- Tasti '0': Nga FixedStand në BalanceTest

## 5.4. Harta dhe Navigimi i Robotit në Gazebo

Për të realizuar simulimin dhe për të kontrolluar robotin, ndiqni hapat e përshkruar më herët, duke bërë ndryshimet e nevojshme për mjedisin aktual, që në këtë rast është një zyre. Simulimi në Gazebo fillohet duke përdorur komandën e mëposhtme:

```
roslaunch unitree_gazebo robot_simulation.launch rname:=a1 wname:=office_small  
rviz:=false
```

Në këtë kontekst, rname përfaqëson emrin e robotit, ndërsa wname përfaqëson emrin e mjedisit të simuluar.

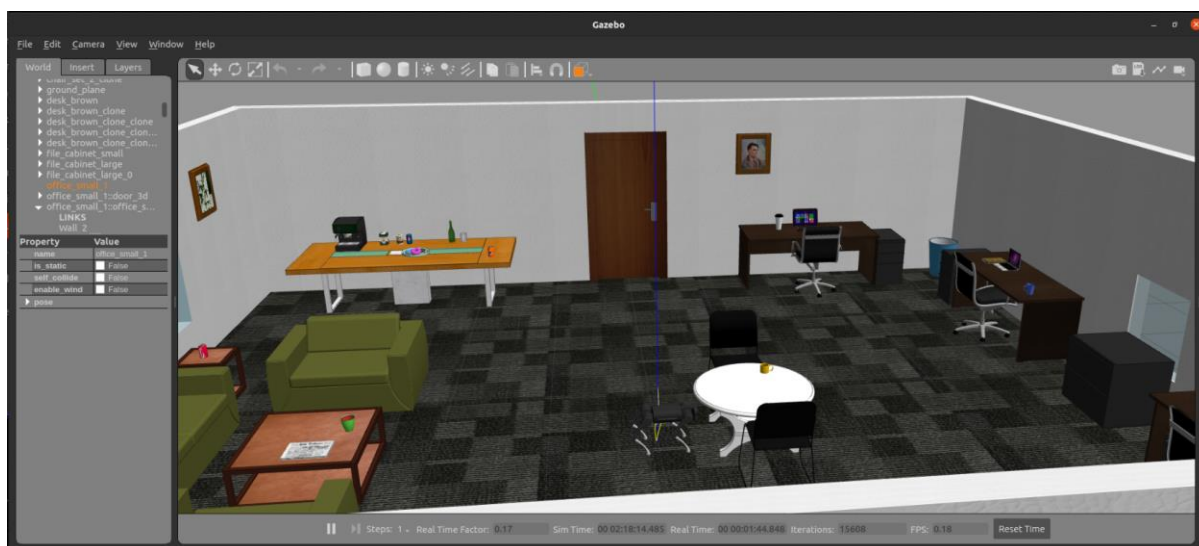


Figura 18: Simulimi i robotit në Gazebo, ku është ngarkuar një ambient virtual që përfaqëson një zyrë.

Për të nisur kontrolluesin e robotit, përdorni komandën:

```
roslaunch unitree_guide junior_ctrl
```

Pasi kontrolluesi të jetë aktivizuar, shtypni butonat '2' dhe '4' në terminalin e kontrolluesit për ta vendosur robotin në modalitetin e lëvizjes.

### 5.4.1. Procesi i Hartëzimit me SLAM

**GMapping** dhe **Hector SLAM** janë algoritme të përdorura për Simultaneous Localization and Mapping (SLAM) në robotikë. Të dyja këto metoda krijojnë një hartë të mjedisit të robotit ndërkohë që robotikë lëviz nëpër atë mjedis dhe gjurmon pozicionin e tij në hartë në kohë reale.

#### ➤ GMapping

**GMapping** është një implementim i SLAM që përdor një algoritëm të bazuar në Filtrimin e Grincave (Particle Filter). Ai është i njohur për aftësinë e tij për të krijuar hartë 2D të mjedisit, ndërkohë që lokalizon pozicionin e robotit. GMapping është veçanërisht efektiv në ambiente të ndryshme dhe të paparashikueshme dhe mund të përballojë një numër të madh të grincave për të përmirësuar saktësinë e hartës. Disa nga karakteristikat kryesore të GMapping janë:

- Përdor sensorë laserik (LIDAR) për të mbledhur të dhëna të distancës.
- Përdor Filtrimin e Grincave për të ruajtur dhe përditësuar hipoteza të shumta për pozicionin e robotit në hartë.

- Është mjaft efektiv në hartimin e mjediseve komplekse dhe të paparashikueshme.

Për të filluar hartëzimin me algoritmin gmapping, përdorni komandën:

```
roslaunch unitree_navigation slam.launch rname:=a1 rviz:=true algorithm:=gmapping
```

### ➤ **Hector SLAM**

**Hector SLAM** është një metodë që nuk përdor filtrimin e grimcave dhe mbështetet kryesisht në përputhjen e skanimeve (scan matching) për lokalizimin dhe hartimin. Hector SLAM është i përshtatshëm për aplikacione që kërkojnë shpejtësi të lartë dhe saktësi të mirë, veçanërisht në mjedise të thjeshta ose në ato ku lëvizjet e robotit janë të shpejta. Disa nga karakteristikat kryesore të Hector SLAM janë:

- Nuk kërkon odometri (të dhëna nga rrotat e robotit) dhe mbështetet tërësisht në të dhënat e LIDAR-it për lokalizim.
- Përdor një metodë të përputhjes së skanimeve për të gjetur pozicionin më të mirë të robotit në hartë.
- Ka aftësi të mira për të lokalizuar robotin edhe në mjedise ku ka mungesë strukturash të qarta për referencë.

Për të filluar hartëzimin me algoritmin hector, përdorni komandën:

```
roslaunch unitree_navigation slam.launch rname:=a1 rviz:=true algorithm:=hector
```

Në përgjithësi, GMapping dhe Hector SLAM janë të dyja metoda të fuqishme për SLAM, por secila ka avantazhet dhe kufizimet e saj, duke u përshtatur në mënyra të ndryshme në varësi të aplikacionit specifik dhe kërkesave të mjedisit ku përdoren.

Gjatë procesit të hartëzimit, Rviz do të shfaqë një vizualizim të hartës së krijuar në kohë reale. Pasi të keni lëvizur robotin rreth mjedisit dhe të keni krijuar hartën, mund ta ruani atë duke përdorur komandën e mëposhtme:

```
roslaunch map_server map_saver -f  
~/catkin_ws/src/ros_unitree/unitree_guide/unitree_navigation/maps/office_small
```

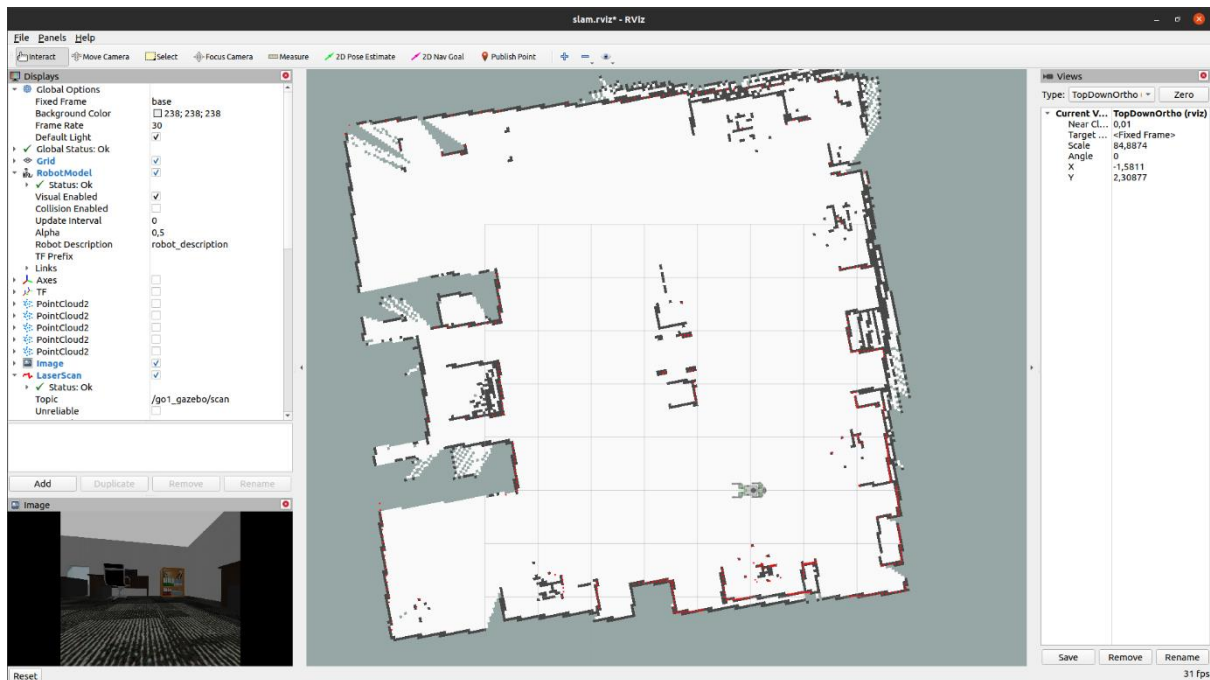


Figura 19 : Gjatë procesit të hartëzimit në RViz shfaq një vizualizim të hartës së krijuar.

#### 5.4.2. *Procesi i Navigimit të Robotit*

Pasi të keni përfunduar hartëzimin, mund të filloni procesin e navigimit.

Përsëritni hapat e simulimit dhe nisjes së kontrolluesit:

```
roslaunch unitree_gazebo robot_simulation.launch rname:=a1 wname:=office_small
rviz:=false
```

```
roslaunch unitree_guide junior_ctrl
```

Shtypni butonat '2' dhe '5' për të aktivizuar modalitetin MoveBase, i cili lejon robotin të ndjekë objektivat e navigimit.

Për të filluar grupin e lundrimit, përdorni komandën:

```
roslaunch unitree_navigation navigation.launch rname:=a1
map_file:=/home/unitree/catkin_ws/src/ros_unitree/unitree_guide/unitree_navigation/maps/
office_small.yaml
```

Në Rviz, vendosni pozicionin fillestar të robotit duke përdorur butonin "2D Pose Estimate". Më pas, caktoni një objektivi navigimi me butonin "Objektivi i Navigimit 2D".

## **6. Navigimi i Robotëve katërkëmbësh - Unitree A1**

Robotët katërkëmbësh kanë fituar një rëndësi të veçantë në fushën e robotikës dhe inteligjencës artificiale për shkak të aftësisë së tyre për të lëvizur me fleksibilitet dhe stabilitet në terrene të ndryshme. Me avancimin e teknologjisë, këta robotë janë bërë të aftë të përballojnë sfida komplekse që përfshijnë kalimin e pengesave, lëvizjen në mjedise të pasigurta dhe kryerjen e detyrave që kërkojnë një shkallë të lartë të autonomisë. Navigimi është një nga aspektet më kritike të funksionimit të robotëve katërkëmbësh, i cili përcakton aftësinë e tyre për të arritur destinacione të caktuara në mënyrë të sigurt dhe efikase, duke minimizuar rreziqet dhe optimizuar rrugën.

Në këtë kontekst, navigimi i robotëve katërkëmbësh nuk kufizohet vetëm në ndjekjen e një rrugë të paracaktuar, por përfshin edhe aftësinë për të perceptuar mjedisin përreth, për të planifikuar rrugën në kohë reale dhe për të marrë vendime autonome bazuar në informacionin e mbledhur nga sensorët e ndryshëm. Një aspekt kyç është përdorimi i teknikave të avancuara të perceptimit, si dhe integrimi i algoritmeve të inteligjencës artificiale dhe mësimit makinerik për të përmirësuar vendimmarrjen gjatë lëvizjes.

Sfida kryesore e navigimit të robotëve katërkëmbësh është trajtimi i mjediseve të ndryshme dhe të paparashikueshme, duke përfshirë terrene të pabarabarta, pengesa dinamike dhe ndryshimet e papritura të mjedisit. Për të përballuar këto sfida, janë zhvilluar teknika të ndryshme për planifikimin e rrugës, shmangien e pengesave, dhe kontrollin e lëvizjes, të cilat synojnë të garantojnë stabilitetin e robotit gjatë lëvizjes dhe të përmirësojnë efikasitetin e tij.

Përmbledhtazi, navigimi i robotëve katërkëmbësh është një fushë komplekse dhe dinamike që kërkon zgjidhje inovative dhe teknologji të avancuara. Zhvillimi dhe përmirësimi i mëtejshëm i këtyre sistemeve do të luajë një rol të rëndësishëm në zgjerimin e aplikacioneve të robotëve katërkëmbësh në mjedise të ndryshme, nga kërkim-shpëtimi deri te eksplorimi i mjediseve të rrezikshme.

### **6.1. Metodatat e Navigimit Lokal**

Navigimi lokal, i njohur gjithashtu si modaliteti në linjë për planifikimin e rrugës, përfshin përdorimin e sensorëve të jashtëm nga roboti për të përcaktuar pozicionin dhe orientimin e tij në mjedis. Këta sensorë i mundësojnë robotit të kontrollojë lëvizjen në mënyrë të saktë dhe të sigurt. Sensorët që përdoren zakonisht për këtë qëllim përfshijnë sensorët infra të kuq, sensorët ultrasonikë, sensorët LASER dhe sensorët vizualë (kamera).

### **6.2. Lokalizimi dhe hartëzimi i njëkohshëm (SLAM)**

Teknikat e navigimit lokal përdorin sensorë të avancuar për të monitoruar orientimin dhe pozicionin e robotit në mjedisin e tij. Një nga sensorët kryesorë të përdorur për automatizimin e këtyre detyrave është LIDAR-i, i cili funksionon në mënyrë të pavarur nga sistemi GPS dhe ofron mundësinë e hartëzimit të saktë të mjedisit.

LIDAR mund të përdoret autonomisht, por kur kombinohet me sensorë të tjerë si GPS, sistemi i navigimit inercial dhe kamerat, mund të përmirësojë ndjeshëm saktësinë dhe performancën e sistemit të navigimit. Për shembull, integrimi me kamerat mund të krijojë një sistem të

fuqishëm për përcaktimin e pozicionit dhe orientimit të robotit. Ky kombinim sensorësh përdoret shpesh për të hartëzuar mjedisin lokal dhe për të lokalizuar piketat referuese, një proces i njohur si Simultaneous Localization and Mapping (SLAM).

Kjo teknikë i mundëson robotit mobil të korrigjojë automatikisht pozicionin dhe orientimin e tij, duke krijuar një përceptim të përditësuar të mjedisit përreth. Për më tepër, përdorimi i sensorëve të koduesve të motorëve së bashku me LIDAR kontribuon në përmirësimin e saktësisë dhe autonomisë së lëvizjes së robotit, duke shtuar një nivel të lartë performance në operacionet në mjedise komplekse.

### **6.2.1. Çfarë është SLAM?**

SLAM (Simultaneous Localization and Mapping) është një metodë shumë e përdorur në robotikë që mundëson krijimin e një harte të mjedisit dhe përcaktimin e pozicionit të robotit në të njëjtën kohë. Gjatë lëvizjes së robotit, matjet e bëra dhe pozicioni i tij ndryshojnë vazhdimisht, duke kërkuar kombinimin e këtyre matjeve nga pozicionet e mëparshme për të krijuar një hartë të saktë dhe të qëndrueshme. Për ta kuptuar më mirë këtë koncept, është e rëndësishme të zërthejmë dhe shqyrtojmë termat lokalizim dhe hartëzim.

#### **➤ Lokalizimi**

Kur një robot autonom është i ndezur, gjëja e parë që bën është të identifikojë se ku ndodhet. Në një skenar tipik lokalizimi, një robot është i pajisur me sensorë për të skanuar rrethinën e tij dhe për të monitoruar lëvizjet e tij. Duke përdorur hyrjet nga sensorët, roboti është në gjendje të identifikojë se ku ndodhet në një hartë të caktuar. Në disa raste, një pajisje gjurmuese si GPS përdoret për të ndihmuar procesin e lokalizimit (kjo është mënyra se si Google mund t'ju japë pikën blu në Google Maps, për shembull).

#### **➤ Hartëzimi**

Pavarësisht se mund të duket si një detyrë e thjeshtë, krijimi i një harte për një robot është një proces kompleks dhe sfidues. Fillimisht, një robot përdor një lloj sensori vizual, si kamera ose lidari, për të regjistruar mjedisin rrethues. Ndërsa roboti lëviz, ai grumbullon më shumë informacion vizual dhe përpiqet të krijojë lidhje mes pikave të ndryshme dhe të nxjerrë veçori të veçanta, siç janë qoshet, për të shënuar pika të identifikueshme. Megjithatë, disa nga këto veçori mund të jenë shumë të ngjashme, duke e bërë të vështirë për robotin të dallojë se cila është cila. Pikërisht këtu hyn në funksion lokalizimi, i cili luan një rol të rëndësishëm në krijimin e hartave më të sakta përmes metodës SLAM (Simultaneous Localization and Mapping).

### **6.2.2. Rëndësia e SLAM**

SLAM (Simultaneous Localization and Mapping) ka qenë objekt i kërkimeve teknike për shumë vite, por zhvillimet e fundit në shpejtësinë e përpunimit kompjuterik dhe disponueshmëria e sensorëve me kosto të ulët, si kamerat dhe sensorët lazer, kanë bërë që algoritmet SLAM të përdoren në një numër gjithnjë në rritje të fushave praktike. Për të kuptuar rëndësinë e SLAM, është e rëndësishme të shqyrtojmë përfitimet e tij dhe shembuj të aplikimeve të suksesshme.

#### **Shembuj të SLAM në praktikë**

Një nga aplikimet më të njohura të SLAM është në robotët fshirës për shtëpi. Pa SLAM, këta robotë do të lëviznin rastësisht brenda një dhome, duke mos pastruar domosdoshmërisht të gjithë sipërfaqen e dyshemesë dhe duke përdorur më shumë energji, gjë që do të shkaktonte shkarkimin më të shpejtë të baterisë. Në kontrast, robotët e pajisur me algoritme SLAM përdorin informacionet nga rrotullimet e rrotave dhe të dhënat nga kamerat dhe sensorët për të përcaktuar lëvizjen e nevojshme. Ky proces njihet si lokalizim. Në të njëjtën kohë, roboti përdor kamerën dhe sensorët për të krijuar një hartë të pengesave përreth tij dhe për të shmangur pastrimin e së njëjtës zonë më shumë se një herë, një proces i njohur si hartogرافي.

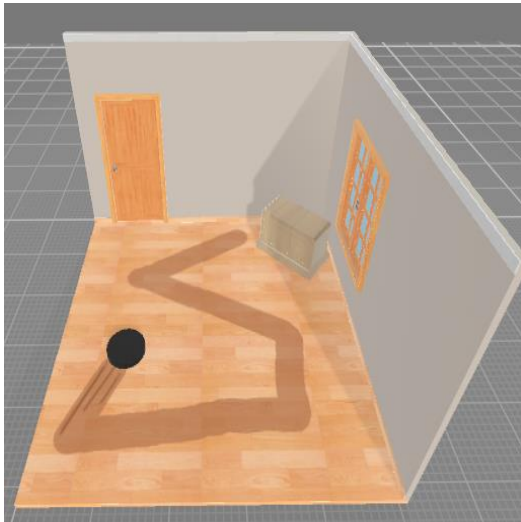


Figura 20 : Pastrimi i dhomës në mënyrë të rastësishme pa SLAM

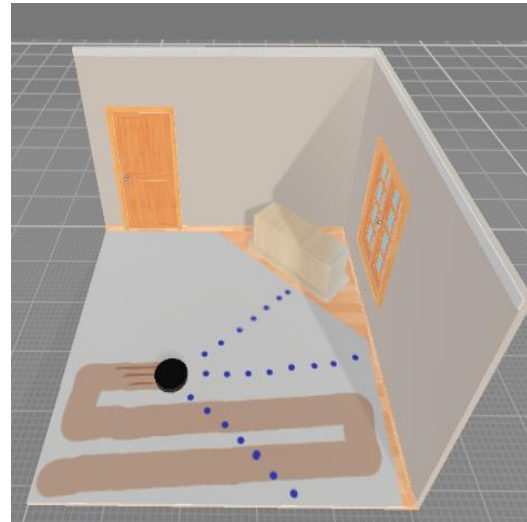


Figura 21: Pastrim me kuptimin e paraqitjes së dhomës me SLAM

### 6.2.3. *Funksionimi i SLAM*

Për të realizuar një proces të saktë dhe të besueshëm të Simultaneous Localization and Mapping (SLAM), një nga qasjet më të zakonshme është përdorimi i një skaneri lazer dhe një sistemi odometrie, i cili mat shpejtësinë dhe distancën bazuar në të dhënat e koduesit. Megjithatë, për shkak të kufizimeve të sensorëve, krijimi i një harte të saktë është një detyrë sfiduese. Këto kufizime, të tilla si rrëshqitja e rrotave, mund të shkaktojnë pasaktësi në klasifikimin e objekteve, duke bërë që i njëjti mur të perceptohet si dy objekte të ndryshme. Për të minimizuar ndikimin e këtyre pasaktësive, algoritmet e SLAM përdorin teknika të ndryshme, si metoda Monte Carlo. Është e rëndësishme të theksohet se algoritmi gmapping gjithashtu përdor metoda të ngjashme për të përmirësuar cilësinë e hartës së krijuar.

Në përgjithësi, ekzistojnë dy komponentë kryesorë të teknologjisë që përdoren për të arritur SLAM: i pari përfshin përpunimin e sinjalit të sensorëve, ku përpunimi në pjesën e përparme varet kryesisht nga lloji i sensorëve të përdorur. I dyti përfshin optimizimin e grafikut të pozicionit, i cili përfshin përpunimin në pjesën e fundit dhe është i pavarur nga sensorët.

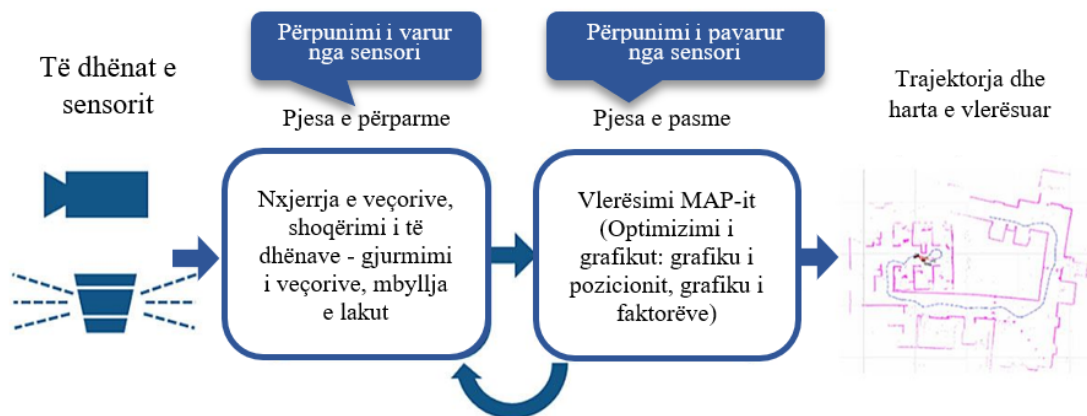


Figura 22: Rrjedhja e përpunimit të SLAM-it [33]

### 6.3. Llojet e metodave SLAM

Për të kuptuar më mirë procesin e përpunimit të pjesës së përparme në sistemet SLAM (Simultaneous Localization and Mapping), është e dobishme të eksploroni llojet e ndryshme të metodave SLAM. Çdo metodë ka karakteristikat dhe përparësitë e saj në përmbushjen e detyrave të lokalizimit dhe hartimit. Ja disa nga metodologjitë kryesore: si SLAM vizuale, lidar SLAM dhe SLAM me shumë sensorë.

#### 6.3.1. SLAM Vizuale

Siç sugjeron emri, SLAM vizuale (ose vSLAM) është një teknikë që përdor imazhe të marra nga kamera dhe sensorë të tjerë optikë për të krijuar një hartë të ambientit dhe për të ndjekur pozicionin e një objekti brenda këtij ambienti. vSLAM mund të përdorë një gamë të gjerë kamerash, përfshirë kamera me kënd të gjerë, kamera me sy peshku, kamera sferike, kamera stereo, shumë kamera, dhe kamera RGB-D (kamera që ofrojnë informacion në përmasa të thellësisë dhe përdorin teknologjinë Time-of-Flight - ToF).

Përdorimi i kamerave për vSLAM mund të realizohet me kosto të ulët, duke përdorur kamera relativisht të lira. Kamerat ofrojnë një volum të madh informacioni, duke mundësuar identifikimin e pikave të referencës (pozitat e matura më parë). Identifikimi i pikave të referencës mund të kombinohet gjithashtu me optimizim të bazuar në grafikë, duke ofruar fleksibilitet në implementimin e SLAM.

SLAM monokular është një lloj i veçantë algoritmi vSLAM që përdor një kamerë të vetme si sensor të vetëm. Kjo ndihmon në identifikimin e thellësisë, e cila është një sfidë, e cila mund të zgjidhet duke përdorur shënues AR (Augmented Reality), tabela, ose objekte të tjera të njohura në imazh për lokalizim, ose duke integruar informacionin e kamerës me sensorë të tjerë, si njësitë matëse inerciale (IMU), të cilat matin sasi fizike si shpejtësia dhe orientimi. Teknologjitë të lidhura me vSLAM përfshijnë strukturën nga lëvizja, odometrinë vizuale dhe rregullimin e paketës.

Algoritmet vizuale SLAM mund të klasifikohen në dy kategori kryesore: metodat e rralla dhe metodat e dendura. Metodatat e rralla përpunohen me pikat e veçorive të imazheve dhe përdorin



algoritme si PTAM (Parallel Tracking and Mapping) dhe ORB-SLAM (Oriented FAST and Rotated BRIEF SLAM). Metodatat e dendura përdorin ndriçimin e përgjithshëm të imazheve dhe përfshijnë algoritme të tilla si DTAM (Dense Tracking and Mapping), LSD-SLAM (Large-Scale Direct Monocular SLAM), DSO (Direct Sparse Odometry) dhe SVO (Semi-Direct Visual Odometry). Algoritmet e vizionit kompjuterik mund të përdoren për të zbuluar veçoritë në imazhe. Për shembull, nëse një robot përdor një kamerë për të hartuar një magazinë, fotografitë e regjistruara mund të përfshijnë paleta, raftet dhe dritat. Duke analizuar ndryshimet në ngjyrën e pikselëve fqinjë, roboti mund të identifikojë objekte të ndryshme në skenë.

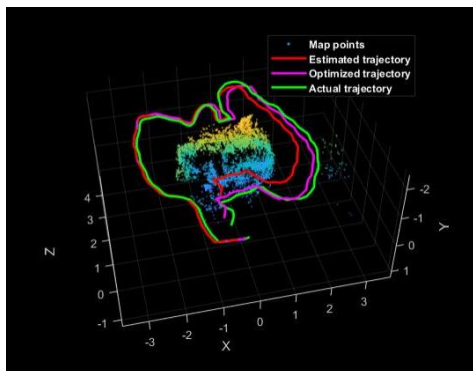


Figura 23: Monokular vSLAM

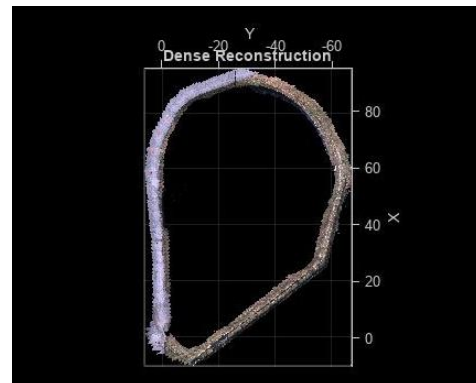


Figura 24: Stereo vSLAM

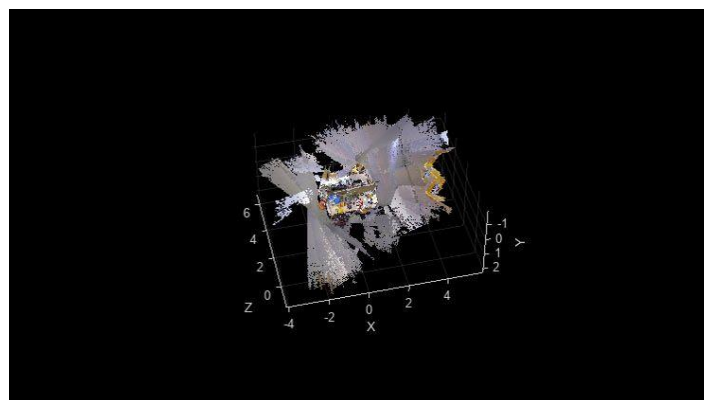


Figura 25: RGB-D vSLAM [33]

### 6.3.2. Lidar SLAM

Lidar SLAM (Simultaneous Localization and Mapping) është një metodë që përdor sensorë lazer për të mbledhur të dhëna rreth ambientit dhe për të krijuar hartë 3D të këtij ambienti. Sensorët Lidar (Light Detection and Ranging) ofrojnë matje të sakta të distancës duke dërguar pulse të dritës dhe duke matur kohën e kthimit të këtyre pulseve.

Në krahasim me SLAM e bazuar në kamera, sensorë ToF (Time-of-Flight), dhe sensorë të tjerë, një sensor Lidar mblodh informacion në mënyrë origjinale për thellësinë dhe gjeometrinë e objekteve në skenë. Nëse një robot magazinash do të përdorte një sensor Lidar në vend të një kamere, ai do të përdorte matjet e distancës dhe format e regjistruara nga sensorët për të

identifikuar objektet. SLAM i bazuar në Lidar përdor skajet dhe rrafshet e regjistruara nga pajisja si veçori për të lidhur informacionin vizual dhe për të krijuar një hartë. Kjo qasje është më efikase për krijimin e një kopjeje dixhitale të sakte dhe realiste të hartës në 3D, duke ofruar një përfaqësim më të saktë të ambientit krahasuar me kamerat.

Daljet nga sensorët Lidar përbëhen nga reja të pikave që mund të jenë 2D (x, y) ose 3D (x, y, z). Reja e pikave e krijuar nga sensorët lazer ofron matje të distancës me saktësi të lartë dhe është shumë efektive për ndërtimin e hartave duke përdorur algoritme SLAM. Procesi i ndjekjes së lëvizjes realizohet në mënyrë sekuenciale duke regjistruar ndryshimet midis reve të pikave. Për lokalizimin e automjetit, përdoren vlerat e lëvizjes të llogaritura (distanca e përshkuar). Për të vlerësuar transformimin relativ midis reve të pikave, përdoren algoritme regjistrimi si Pika e Përsëritur më e Afërt (Iterative Closest Point (ICP)) dhe Transformimi i Shpërndarjeve Normale (Normal Distributions Transform (NDT)). Alternativisht, mund të përdoren qasje të bazuara në veçori, si Odometria dhe Harta e Lidarit (Lidar Odometry and Mapping (LOAM)) ose Regjistrim i Shpejtë Global (Fast Global Registration (FGR)), të cilat përdorin veçoritë e Histogramet të Tipareve të Pikave të Shpejta (Fast Point Feature Histograms (FPFH)). Hartat e reve me pika 2D ose 3D mund të përfaqësohen si harta rrjeti ose harta voxel.

Për shkak të sfidave që lidhen me Lidar SLAM, lokalizimi i robotëve autonom shpesh përfshin shkrirjen e të dhënave nga sensorë të ndryshëm, si odometria e rrotave, Sistemi i Navigimit Global nga Satelitët (Global Navigation Satellite System (GNSS)), dhe të dhënat e njësive matëse inerciale (Inertial Measurement Unit (IMU)). Në aplikacione si robotët e magazinës, zakonisht përdoret Lidar SLAM 2D, ndërsa SLAM me reja të pikave 3D është më i zakonshëm për UAV-të dhe aplikacione të tjera të automatikës së drejtimit.

Në krahasim me SLAM e bazuar në kamera, një sensor Lidar mbledh informacion në mënyrë origjinale për thellësinë dhe gjeometrinë e objekteve në skenë. Nëse një robot magazinash do të përdorte një sensor Lidar në vend të një kamere, ai do të përdorte matjet e distancës dhe format e regjistruara nga sensorët për të identifikuar objektet. SLAM i bazuar në Lidar përdor skajet dhe rrafshet e regjistruara nga pajisja si veçori për të lidhur informacionin vizual dhe për të krijuar një hartë. Kjo qasje është më efikase për krijimin e një kopjeje dixhitale të sakte dhe realiste të hartës në 3D, duke ofruar një përfaqësim më të saktë të ambientit krahasuar me kamerat.



Figura 26: SLAM me 2D LiDAR

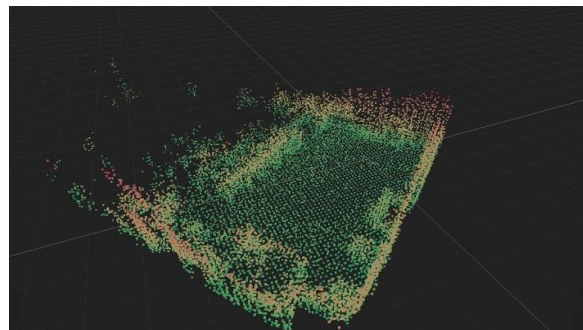


Figura 27: SLAM me 3D LiDAR

### 6.3.3. SLAM me shumë sensorë

SLAM me shumë sensorë është një algoritëm i avancuar që përdor një gamë të gjerë sensorësh, duke përfshirë kamera, njësitë e matjes inerciale (IMU), sistemin e pozicionimit global (GPS), lidar, radar dhe të tjerë, për të përmirësuar saktësinë dhe qëndrueshmërinë e algoritmeve SLAM. Duke kombinuar fuqitë plotësuese të këtyre sensorëve dhe duke zbutur kufizimet e tyre individuale, SLAM me shumë sensorë mund të ofrojë një performancë të përmirësuar në mjedise të ndryshme.

Për shembull, kamerat sigurojnë të dhëna vizuale shumë të detajuara, por mund të hasin vështirësi në mjedise me ndriçim të ulët ose me shpejtësi të lartë. Në anën tjetër, lidar funksionon në mënyrë të qëndrueshme në kushte të ndryshme ndriçimi, por mund të ketë sfida në përpunimin e materialeve të caktuara sipërfaqësore. SLAM me shumë sensorë integron të dhëna nga burime të ndryshme, duke ofruar kështu një zgjidhje më të besueshme sesa qasjet që përdorin vetëm një sensor.

Grafiku i faktorëve është një kornizë modulare dhe fleksibile që mundëson integrimin e llojeve të ndryshme të sensorëve, si kamera, IMU dhe GPS. Për më tepër, kjo kornizë mbështet hyrjet e personalizuar nga sensorë të tjerë, si lidar dhe odometri, duke i konvertuar këto të dhëna në faktorë pozitivi. Kjo qasje mundëson konfigurime të ndryshme të SLAM me shumë sensorë, si SLAM vizual-inercial monocular dhe SLAM lidar-IMU, duke ofruar një fleksibilitet të lartë në aplikime të ndryshme.

## 6.4. Zgjedhja e algoritmit të duhur SLAM

Ekzistojnë disa metoda të ndryshme për Simultaneous Localization and Mapping (SLAM), secila prej tyre me qasje dhe rezultate të ndryshme. Çdo metodë ka avantazhet dhe kufizimet e veta, të cilat ndryshojnë në varësi të aplikimit specifik. Për të përcaktuar algoritmin më të përshtatshëm SLAM, është e rëndësishme të kuptohet natyra e kërkesave dhe të merret parasysh një sërë faktorësh kyç.

Një ndër këta faktorë është prodhimi i kërkuar i algoritmit SLAM. Për shembull, nëse qëllimi është të lokalizohen objekte në një park, duhet përcaktuar nëse është i nevojshëm pozicionimi relativ i objekteve si pemët dhe stolat (qasje topologjike) apo matja e saktë e distancave mes tyre (qasje metrike). Në raste të tjera, mund të jetë e nevojshme të rikrijohet një kopje dixhitale e parkut (qasje volumetrike) ose të ofrohet informacion i mjaftueshëm për të dalluar objektet (qasje e bazuar në veçori). Algoritmet e ndryshme kërkojnë burime të ndryshme të përpunimit dhe lloje të ndryshme sensorësh për t'u realizuar me sukses.

Një tjetër faktor i rëndësishëm është natyra e mjedisit. Mjedisi ku ekzekutohet SLAM mund të jetë statik, duke ruajtur një strukturë të pandryshueshme me kalimin e kohës, ose dinamik, duke kërkuar përditësime të vazhdueshme. Për shembull, një robot që lëviz në një magazinë të zbrazët dhe më pas përballet me tufa paletash gjatë kthimit duhet të jetë në gjendje të lidhë vendndodhjen aktuale me hartën e krijuar më parë, duke marrë parasysh ndryshimet në rrethinë.

Një sfidë tjetër ndërlikohet kur kemi përdorimin e shumë robotëve në të njëjtin mjedis. Në një skenar ku disa robotë lëvizin autonomisht në një hapësirë të përbashkët, si një magazinë, është thelbësore që secili robot të dijë pozicionin e të tjerëve dhe të shmangë përfshirjen e tyre në hartografi. Nëse robotët mund të komunikojnë me njëri-tjetrin, mund të krijohen harta të vogla

lokale të cilat më pas ndahen me një sistem qendror për të formuar një hartë të përbashkët dhe më të saktë të të gjithë mjedisit.

Përfundimisht, nuk ekziston një algoritëm universal SLAM që mund të përballojë të gjitha rastet e përdorimit. Megjithatë, përparimet teknologjike po e zhvillojnë vazhdimisht këtë fushë. Për të përzgjedhur algoritmin më të përshtatshëm, është thelbësore të kuptohet thellësisht rasti i përdorimit dhe të ekzplorojnë zgjidhje të ndryshme për të përcaktuar atë më të efektshmen.

## **6.5. Sfidat e zakonshme me SLAM**

Pavarësisht nga përdorimi i gjerë i algoritmeve SLAM në aplikime praktike, disa sfida teknike vazhdojnë të pengojnë adoptimin e tyre në shkallë më të gjerë. Megjithatë, për secilën prej këtyre sfidave ekzistojnë kundërmasa specifike që mund të ndihmojnë në kapërcimin e këtyre pengesave dhe përmirësimin e performancës së algoritmeve SLAM në mjedise komplekse.

### **6.5.1. Grumbullimi i Gabimeve të Lokalizimit dhe Optimizimi në Algoritmet SLAM**

Algoritmet e SLAM (Simultaneous Localization and Mapping) vlerësojnë lëvizjen sekuenciale të robotit, duke përfshirë një kufi të caktuar gabimi në çdo hap të vlerësimit. Ky gabim ka tendencën të grumbullohet me kalimin e kohës, duke çuar në devijime të ndjeshme nga vlerat aktuale të pozicionit. Si pasojë, mund të ndodhin deformime ose gabime në të dhënat e hartës, të cilat e vështirësojnë procesin e mëvonshëm të lokalizimit dhe hartëzimit. Një shembull i zakonshëm është rasti i lëvizjes së robotit në një trajektore katrore, ku akumulimi i gabimit bën që pikat e fillimit dhe përfundimit të mos përputhen më. Ky fenomen njihet si "problemi i mbylljes së lakut". Gabime të tilla të vlerësimit të pozave janë të pashmangshme, prandaj është jetike të identifikohen momentet e mbylljes së lakut dhe të zhvillohen teknika për korrigjimin ose eliminimin e gabimeve të akumuluar.

Në kontekstin e SLAM me shumë sensorë, kalibrimi i saktë i sensorëve luan një rol vendimtar. Gabimet në kalibrim ose mospërputhjet mes sensorëve mund të çojnë në pasaktësi në shkrirjen e të dhënave, duke dëmtuar funksionalitetin e përgjithshëm të sistemit SLAM. Një metodë efektive për përmirësimin e kalibrimit është optimizimi i grafikut të faktorëve, i cili përfshin integrimin e sistemeve kamera-IMU.

Një masë kundër këtyre problemeve është ruajtja e disa karakteristikave nga vende të vizituara më parë si pikë referimi, në mënyrë që të minimizohet gabimi i lokalizimit në lëvizje të mëvonshme. Për të ndihmuar në këtë proces, grafikët e pozave përdoren për të përmirësuar saktësinë e hartëzimit. Duke trajtuar minimizimin e gabimeve si një problem optimizimi, mund të krijohen të dhëna më të sakta të hartës. Ky proces i optimizimit, i njohur si "rregullim i paketës", është veçanërisht i rëndësishëm në SLAM-in vizual.

### **6.5.2. Dështimi i Lokalizimit dhe Humbja e Pozicionit në Hartë**

Lokalizimi i bazuar në harta të pikë-re dhe imazhe nuk merr parasysh karakteristikat e veçanta të lëvizjes së robotit. Kjo qasje mund të çojë në vlerësime të gabuara dhe të papërshtatshme të pozicionit, duke rezultuar në situata të pasakta. Një shembull tipik është kur një robot që lëviz me një shpejtësi prej 1 m/s papritur shfaqet 10 metra më përpara, duke shkaktuar një ndërprerje në sistemin e lokalizimit. Këto dështime mund të shmangen ose të korrigjohen përmes

përdorimit të algoritmeve rikuperuese ose integritit të modeleve të lëvizjes me të dhëna nga sensorë të shumtë, për të përmirësuar saktësinë e vlerësimeve të pozicionit.

Një metodë efektive për përmirësimin e lokalizimit përmes shkrirjes së sensorëve është përdorimi i filtrave të avancuar. Një teknikë e zakonshme është Filtrimi Kalman, i cili përdoret për të përmirësuar vlerësimet e pozicionit. Duke qenë se shumica e robotëve me lëvizje diferenciale dhe automjete me katër rrota kanë modele të lëvizjes jolineare, përdoren shpesh Filtrat e Zgjeruar Kalman dhe Filtrat e Grimcave (të njohur si Lokalizimi Monte Carlo). Në disa raste, mund të përdoren edhe filtra më fleksibël si Filtrat Pa Aroma Kalman (UKF) ose metodologji të tjera bazuara në Filtrimin Bayes. Pajisjet e zakonshme për matjen e të dhënave përfshijnë sensorët inercialë si IMU (Njësia Matëse Inercialë), Sistemi i Referencës së Qëndrimit dhe Drejtimit (AHRS), Sistemi i Navigimit Inercial (INS), përshpejtuesit, girokopët dhe sensorët magnetikë. Koduesit e rrotave, të montuar në automjet, janë gjithashtu të rëndësishëm për përmirësimin e odometrisë.

Kur lokalizimi dështon, një kundërmasë e rëndësishme për rikuperim është përdorimi i një pike referimi nga një vend i vizituar më parë, e cila shërben si një kornizë kyçe. Procesi i identifikimit të kësaj pike referimi përfshin nxjerrjen e veçorive për të mundësuar një skanim të shpejtë dhe efikas. Disa metoda të njohura për lokalizim përfshijnë teknikat e bazuara në veçoritë e imazhit, si Çanta e Veçorive (BoF) dhe Çanta e Fjalëve Vizuale (BoVW). Kohët e fundit, algoritmet e mësimin të thellë janë përdorur për krahasimin e distancave midis veçorive për të përmirësuar procesin e lokalizimit.

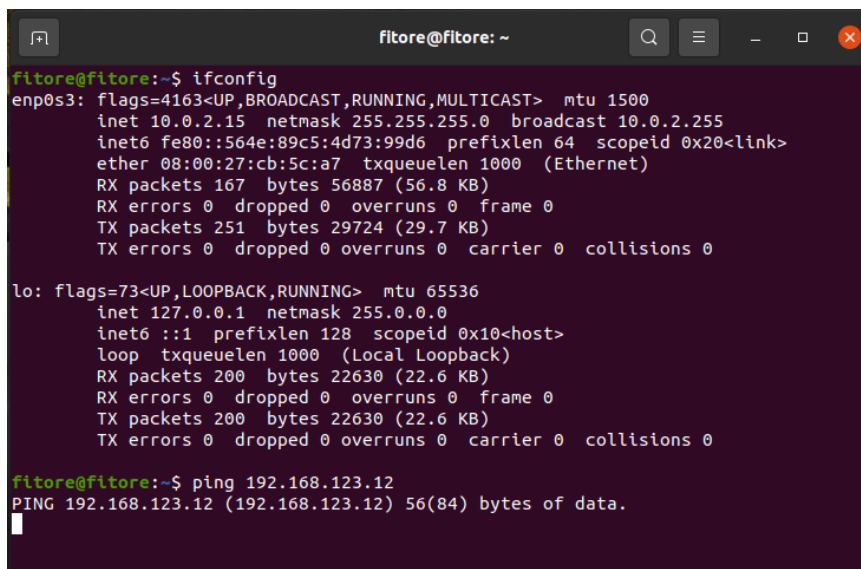
### ***6.5.3. Kostoja e Lartë Llogaritëse për Përpunimin e Imazhit, Pikës së Resë dhe Optimizimin***

Një sfidë e rëndësishme në zbatimin e algoritmeve SLAM në automjete ose robotë është kostoja e lartë llogaritëse, sidomos kur përdoren mikroprocesorë të integruar kompakt dhe me konsum të ulët energjie, të cilët kanë kapacitete të kufizuara përpunuese. Për të arritur një lokalizim të saktë, kërkohet përpunimi i imazhit dhe përputhja e pikës së resë me frekuencë të lartë, ndërsa optimizimet komplekse, siç është mbyllja e lakut, rrisin ndjeshëm kërkesat për llogaritje. Sfidat qëndron në ekzekutimin e këtyre proceseve llogaritëse të ndërlikuara në mikrokompjuterë të integruar me burime të kufizuara.

Një zgjidhje për këtë problem është drejtimi i proceseve në mënyrë paralele. Proceset si nxjerrja e veçorive, që është një fazë parapërpunimi për përputhjen, janë veçanërisht të përshtatshme për paralelizim. Duke përdorur CPU-të me shumë bërthama, përpunimi i të dhënave në blloqe të shumta me një udhëzim të vetëm (SIMD), dhe shfrytëzimi i GPU-ve të integruara mund të ndihmojnë në rritjen e efikasitetit llogaritës. Për më tepër, duke kryer optimizimin e grafikut të pozave në intervale të rregullta dhe me prioritet të ulët, mund të zvogëlohet ngarkesa e përgjithshme dhe të përmirësohet performanca e sistemit pa ndikuar në lokalizimin e saktë në kohë reale.

## 7. Testimi Real i Robotit

Për të kryer testimin real të robotit, hapi i parë është konfigurimi i lidhjes së internetit midis kompjuterit dhe robotit, duke përdorur një kablo **Ethernet**. Pas lidhjes, duhet të ekzekutohet komanda **ifconfig** në një terminal për të identifikuar emrin e portit të rrjetit, siç ilustron në figurën e mëposhtme.



```
ftlore@ftlore: ~  
ftlore@ftlore:~$ ifconfig  
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255  
inet6 fe80::564e:89c5:4d73:99d6 prefixlen 64 scopeid 0x20<link>  
ether 08:00:27:cb:5c:a7 txqueuelen 1000 (Ethernet)  
RX packets 167 bytes 56887 (56.8 KB)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 251 bytes 29724 (29.7 KB)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
inet 127.0.0.1 netmask 255.0.0.0  
inet6 ::1 prefixlen 128 scopeid 0x10<host>  
loop txqueuelen 1000 (Local Loopback)  
RX packets 200 bytes 22630 (22.6 KB)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 200 bytes 22630 (22.6 KB)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
ftlore@ftlore:~$ ping 192.168.123.12  
PING 192.168.123.12 (192.168.123.12) 56(84) bytes of data.  
|
```

Figura 28: Identifikimi i portit të rrjetit dhe konfirmimi i komunikimit ndërmjet PC-së dhe robotit përmes rrjetit.

Në figurën e mësipërme ilustron përdorimi i komandës ping dhe adresës IP të robotit (192.168.123.12) për të verifikuar lidhjen mes PC-së dhe robotit përmes rrjetit. Nëse rrjeti është konfiguruar siç duhet, sistemi do të marrë një përgjigje nga roboti, duke konfirmuar kështu një lidhje të suksesshme.

### 7.1. Krijimi i hartës dhe navigimi i robotit

Në këtë pjesë, do të krijojmë një hartë të detajuar duke përdorur aplikacionin Paint 3D. Në hartë do të paraqesim dy pika kryesore: pikën fillestare (start) dhe pikën përfundimtare (goal), të cilat janë thelbësore për planifikimin e levizjes së robotit. Pika fillestare simbolizon vendin ku robotit i fillon misioni, ndërsa pika përfundimtare tregon destinacionin që ai duhet të arrijë.



Figura 29: Harta e krijuar ne Paint 3D

Për të ilustruar më mirë hartën, në këtë pjesë janë paraqitur disa fotografi të ndryshme të hartës nga kënde të ndryshme. Këto pamje do të ndihmojnë në përshkrimin e rrugës që do të ndjekë roboti gjatë levizjes së tij, duke theksuar veçoritë kryesore dhe pengesat e mundshme që mund të hasë gjatë trajektorës.

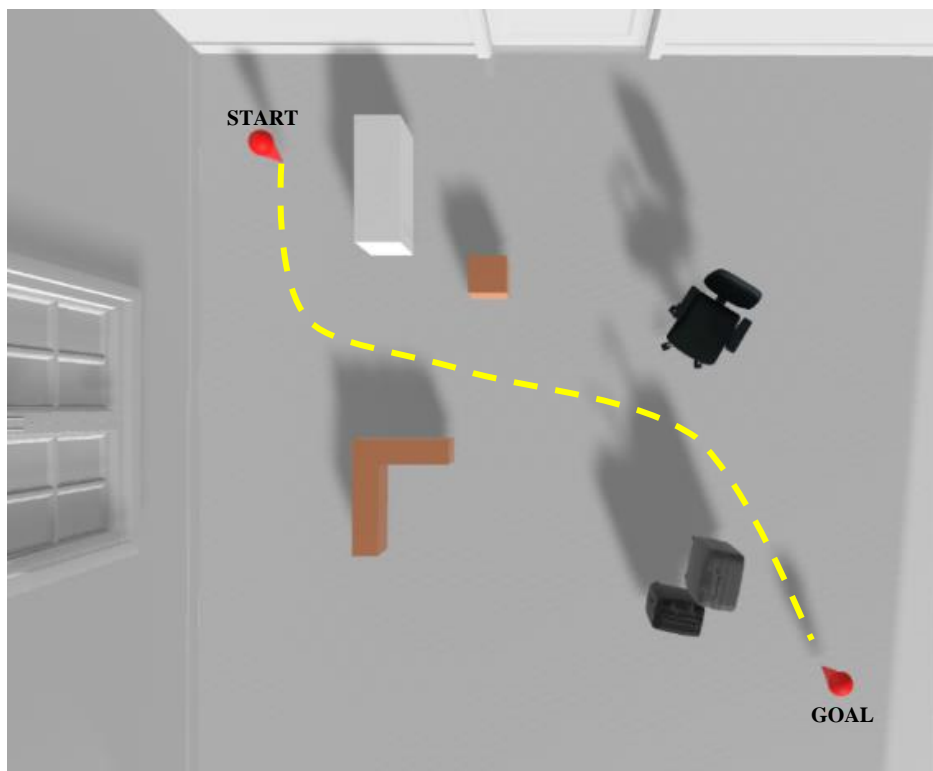
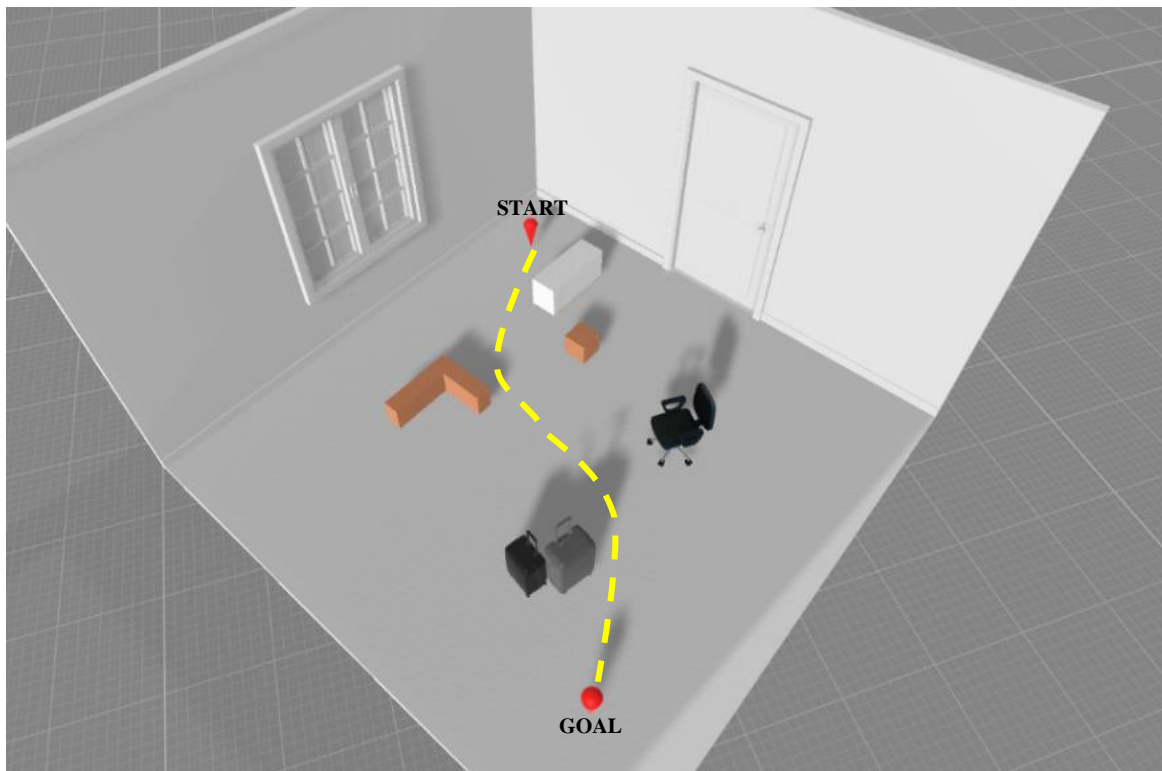


Figura 30: Paraqitja e trajektorës së robotit në hartën e krijuar.



*Figura 31: Paraqitja e trajektores së robotit në hartën e krijuar.*

## **7.2. RPLidar S1**

Lidari (Light Detection and Ranging) është një teknologji e cila përdor impulse lazer për të matur distancën dhe për të krijuar një hartë të saktë të mjedisit përreth. Një nga sensorët më të përdorur në këtë fushë është RPLidar S1, i cili gjen aplikim të gjerë në robotikë, sistemet autonome, dhe studimet e lokalizimit dhe hartografisë. Këtu janë paraqet karakteristikat kryesore të RPLidar S1, mënyrën se si funksionon, avantazhet dhe sfidat që paraqet, si dhe përdorimet e tij praktike në aplikime të ndryshme robotike.



*Figura 32: RPLidari S1 që është zbatuar në rastin tonë.*



### **7.2.1. Karakteristikat teknike të RPLidarit S1**

RPLidar S1 është një sensor lazer që mundëson skanimin e plotë 360° të një mjedisi përreth, duke ofruar matje të sakta të distancës deri në 40 metra. Ky sensor përdor teknologjinë Time of Flight (ToF) për të matur kohën që i duhet një impuls lazeri të reflektohet nga një objekt dhe të kthehet në sensor. Kjo e bën atë ideal për detyrat që kërkojnë saktësi të lartë dhe reagime të shpejta.

Shpejtësia e skanimit është një nga avantazhet kryesore të këtij sensori, pasi arrin deri në 18,000 skanime për sekondë, duke siguruar frekuenca të azhurnimit prej 10 deri në 20 Hz, në varësi të cilësimeve. Kjo frekuencë i mundëson RPLidar S1 të përpunojë të dhëna në kohë reale dhe të ofrojë rezolucion të lartë për mjedise komplekse dhe të ndryshueshme.

Ndër të tjera, fusha e plotë e skanimit prej 360° dhe saktësia e lartë prej  $\pm 3$  cm e bëjnë këtë sensor një zgjedhje të shkëlqyer për aplikime robotike të avancuara. Ai funksionon në një fushë të operimit prej  $-10^{\circ}\text{C}$  deri në  $45^{\circ}\text{C}$ , duke garantuar stabilitet në kushte të ndryshme mjedisore.

### **7.2.2. Funksionimi dhe parimet bazë**

RPLidar S1 përdor teknologjinë ToF (Time of Flight) për të llogaritur distancat nga objektet përreth. Ai lëshon impulse lazer në drejtim të mjedisit dhe pret kthimin e tyre pas reflektimit nga objektet. Bazuar në kohën e kthimit, llogaritet distanca midis sensorit dhe objektit të reflektuar. Ky proces përsëritet për të gjithë fushën e skanimit 360°, duke gjeneruar një hartë të pikave (point cloud) të mjedisit përreth.

Një komponent kyç i RPLidar S1 është mekanizmi rrotullues, i cili lejon sensorin të skanojë ambientin në mënyrë të vazhdueshme. Ndërveprimi i saktë midis mekanikës dhe softuerit është kritik për marrjen e matjeve të sakta dhe të qëndrueshme.

### **7.2.3. Përdorimi në aplikimet robotike**

RPLidar S1 ka gjetur një vend të rëndësishëm në aplikimet robotike, kryesisht për detyra të tilla si navigimi, shmangia e pengesave, dhe SLAM (Simultaneous Localization and Mapping). Kjo teknologji u mundëson robotëve të kuptojnë mjedisin përreth tyre dhe të ndërveprojnë me të në mënyrë autonome. Përmes hartës së pikave të krijuar nga lidari, robotët mund të krijojnë harta të sakta të mjedisit dhe të përcaktojnë rrugët më të sigurta për të shmangur pengesat.

Një përdorim i rëndësishëm i RPLidar S1 është në algoritmet SLAM, ku roboti duhet të lokalizojë veten brenda një mjedisi të panjohur dhe të krijojë një hartë të tij njëkohësisht. Falë frekuencës së lartë të azhurnimit dhe rezolucionit të lartë të skanimit, RPLidar S1 është në gjendje të përpunojë dhe të japë të dhëna të besueshme në kohë reale, duke i lejuar robotët të ndërmarrin vendime të shpejta dhe të sakta.

### **7.2.4. Avantazhet dhe të metat e RPLidarit S1**

Avantazhet e këtij sensori përfshijnë:

- *Distanca e gjatë e matjes:* Deri në 40 metra, duke mbuluar një zonë të gjerë të mjedisit përreth.
- *Shpejtësia dhe frekuenca e lartë e skanimit:* Kjo siguron që të dhënat të përpunohen shpejt dhe të jenë të përditësuara, të domosdoshme për aplikimet në kohë reale.
- *Konsum i ulët energjie:* Sensori ka një konsum prej vetëm 5W, duke e bërë atë ekonomik për robotët që funksionojnë me bateri.
- *Kompatibilitet me ROS:* Integrimi i tij me Robot Operating System (ROS) lehtëson zhvillimin e algoritmeve të avancuara dhe përpunimin e të dhënave.

Megjithatë, RPLidari S1 ka edhe disa të meta:

- *Ndjeshmëri ndaj dritës së fortë:* Në mjedise me dritë të lartë ose ekspozim të drejtpërdrejtë ndaj rrezeve të diellit, performanca mund të bjerë ndjeshëm.
- *Vështirësi me materialet reflektuese ose të tejdukshme:* Objektet si xhami apo metalet reflektuese mund të ndikojnë në saktësinë e skanimit.
- *Mbrojtja nga pluhuri dhe ndotja:* Ndërsa është një pajisje precize, mbajtja e saj e pastër është thelbësore për të ruajtur performancën.

### **7.2.5. Zbatimi praktik në robotikë dhe automjetet autonome**

Përdorimi i RPLidarit S1 është i gjerë në robotët autonomë dhe automjetet autonome, për shkak të aftësive të tij për të gjeneruar të dhëna të sakta të mjedisit përreth. Në sistemet autonome, lidari përdoret për hartimin e rrugëve, lokalizimin dhe planifikimin e lëvizjeve, duke lejuar këto sisteme të lëvizin në mënyrë të sigurt dhe të shmangin pengesat.

Në aplikimet industriale dhe dronët gjithashtu përdoret për navigim, ndërsa në automjetet autonome, lidari luan një rol kyç në detektimin e pengesave dhe menaxhimin e distancave të sigurisë.

RPLidari S1 është një sensor i fuqishëm për detyrat e matjes dhe skanimit në aplikimet robotike, duke ofruar saktësi dhe performancë të lartë në kohë reale. Karakteristikat e tij, përfshirë shpejtësinë e skanimit, distancën e zbulimit dhe fushën e plotë të pamjes, e bëjnë atë të përshtatshëm për një gamë të gjerë aplikimesh në robotikë dhe sisteme autonome. Pavarësisht disa kufizimeve të tij në mjedise të ndriçuara fort ose me objekte reflektuese, ai mbetet një komponent kritik për zhvillimin e teknologjive moderne të lokalizimit dhe hartografisë

### **7.3. Navigimi i robotit**

Në këtë pjesë, do të shqyrtojmë procesin e navigimit të robotit duke përdorur teknologjitë RPLIDAR S1 dhe SLAM (Simultaneous Localization and Mapping). Ky sistem lejon robotin të lëvizë nga një pikë fillestare në një pikë përfundimtare, duke shmangur pengesat në ambientin e tij.

Roboti nis nga pika fillestare, ku RPLIDAR S1 aktivizohet për të skanuar hapësirën përreth. Ky sensor lidars ofron informacion në kohë reale mbi pengesat e mundshme dhe konfigurimin e ambientit, duke i mundur robotit të krijojë një hartë të saktë të mjedisit. Gjatë këtij momenti, roboti analizon të dhënat e marra dhe planifikon rrugën më efikase për të arritur destinacionin e tij.

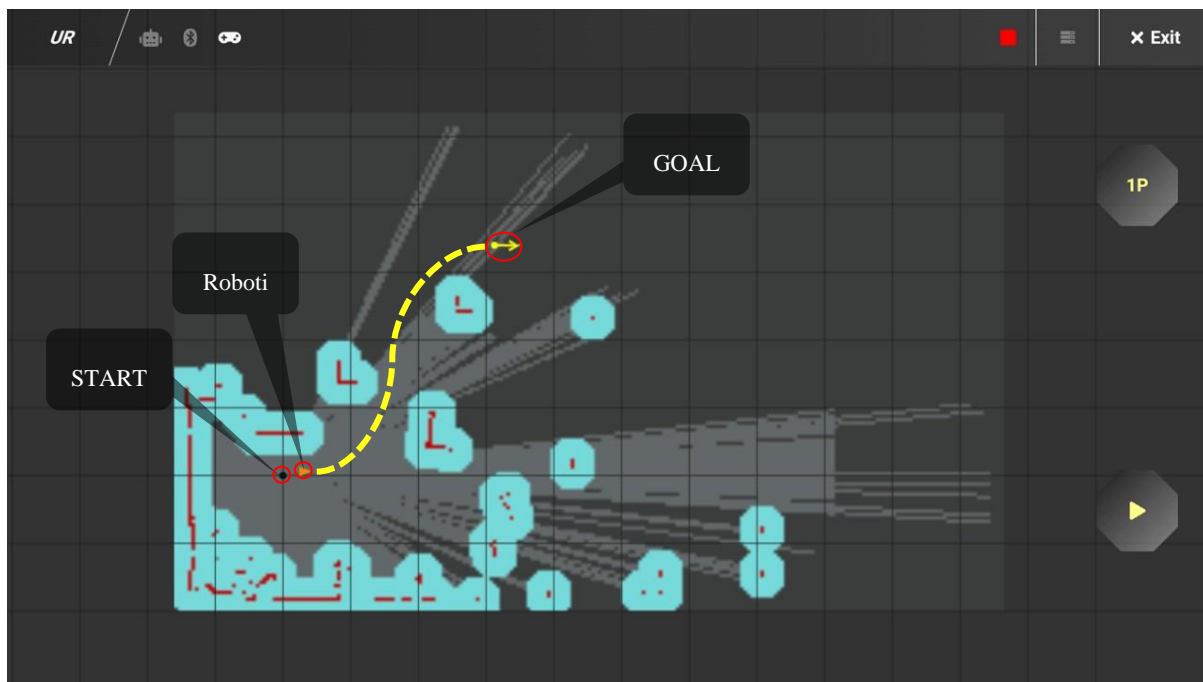


Figura 33: Përgaditja e robotit në pikën fillestare për navigim .

Kur roboti arrin në gjysmën e rrugës, ai tregon një përparim të konsiderueshëm. Përdorimi i SLAM-it i lejon atij të përditësojë vazhdimisht pozitat e tij dhe të përshtatë planin e levizjes në përputhje me pengesat që has në rrugë. Kjo fazë është kritike, pasi robotit i duhet të demonstrojë aftësitë e tij për të naviguar me saktësi në një ambient të ndërlikuar.

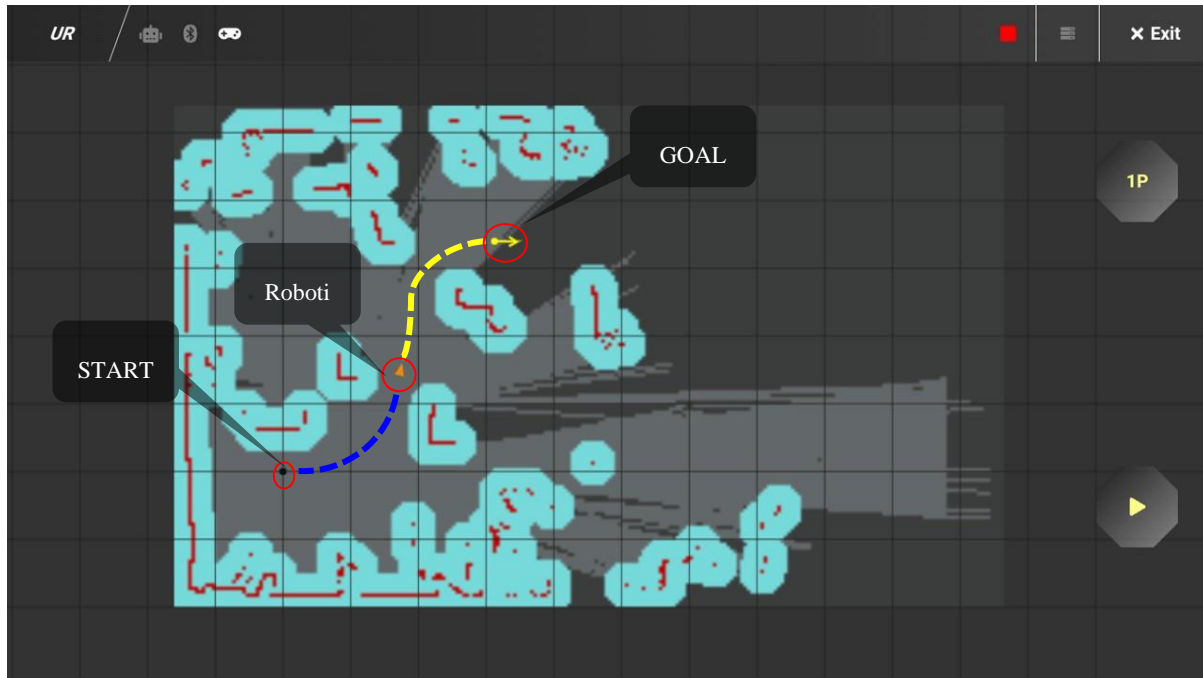


Figura 34: Roboti në gjysmën e rrugës, duke shmangur pengesat.

Kur roboti arrin në pikën përfundimtare, ai ka realizuar me sukses misionin e tij, duke shmangur të gjitha pengesat që ndodhën gjatë rrugës. Ky proces jo vetëm që tregon efektivitetin e RPLIDAR S1 dhe SLAM-it, por gjithashtu ilustron potencialin e teknologjive të avancuara

në robotikë për të përmirësuar autonominë dhe performancën e robotëve në ambientet e ndryshme.

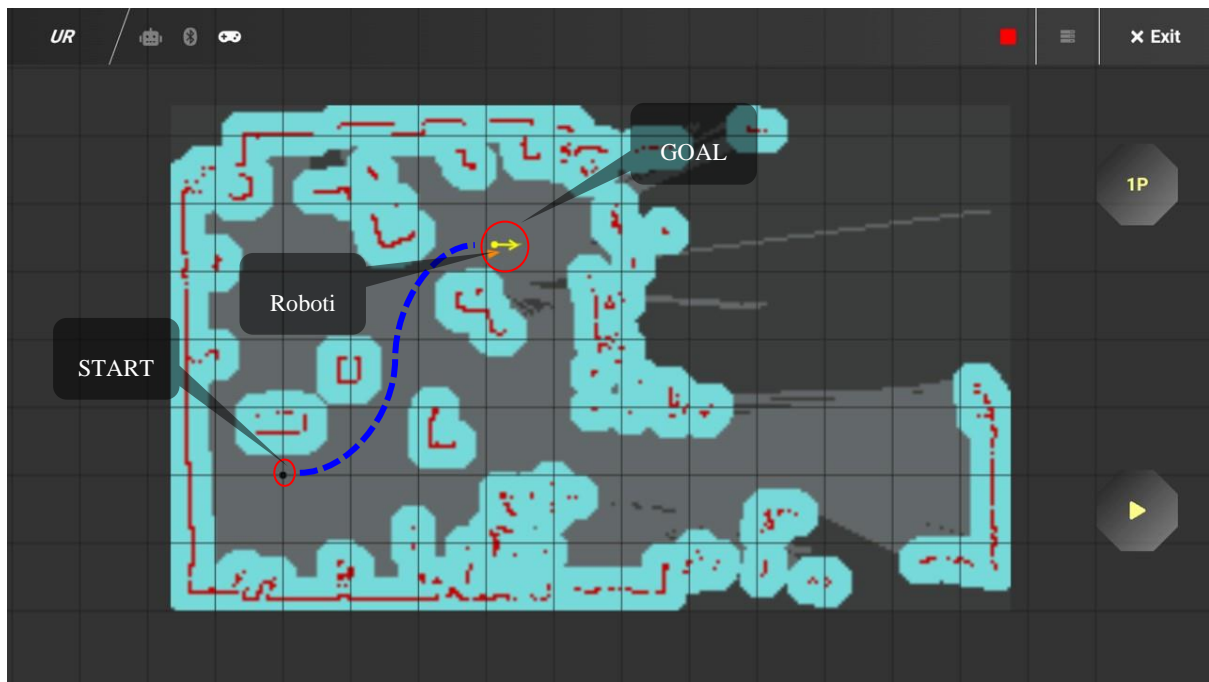


Figura 35: Roboti arrin në pikën përfundimtare me sukses pas përfundimit të navigimit.

## 8. Përfundimi

Në përfundim të këtij punimi, është theksuar rëndësia e integritit të robotit katërkëmbësh Unitree A1 me sistemin operativ ROS për të përmirësuar perceptimin dhe kontrollin në mjedise të ndërlikuara. Përmes testimeve dhe simulimeve, është vërtetuar se kjo teknologji ofron zgjidhje të suksesshme për navigim autonom, shmangie të pengesave dhe hartëzim të saktë duke përdorur algoritmet SLAM. Rezultatet kanë treguar përmirësime të dukshme në siguri, stabilitet dhe ndërveprim me mjedisin.

Integrimi i avancuar i sensorëve dhe koordinimi i tyre përmes ROS përfaqëson një hap të rëndësishëm drejt zhvillimit të robotëve më të sofistikuar dhe të besueshëm, me potencial për aplikime në sfidat e teknologjisë moderne. Ky punim trajton robotin Unitree A1 duke përdorur ROS, gjë që përmirëson ndjeshëm aftësitë e tij për perceptim dhe kontroll në mjedise komplekse. Përdorimi i RPLidarit dhe algoritmet SLAM ka dhënë rezultate premtuese në zgjerimin e aftësive të robotit për të lundruar dhe ndërvepruar në këto mjedise.

Rezultatet eksperimentale nga simulimet në platformën Gazebo dhe testet reale kanë demonstruar suksesin e kësaj metode, duke përmirësuar saktësinë e lokalizimit dhe aftësinë e robotit për të shmangur pengesat. Kjo qasje rrit sigurinë dhe stabilitetin e lëvizjes së robotit, ndërsa përmirëson aftësinë e tij për të marrë vendime autonome në kohë reale, të nevojshme për aplikime të avancuara si kërkim-shpëtimi dhe inspektimi industrial.

## Referencat

- [1]. Unitree A1: <https://www.unitree.com/a1/> (last check 28/04/2024)
- [2]. Bledt G, Powell MJ, Katz B, Di Carlo J, Wensing PM, Kim S. Mit cheetah 3: Design and control of a robust, dynamic quadruped robot. In 2018 IEEE. InRSJ International Conference on Intelligent Robots and Systems (IROS) 2019 (pp. 2245-2252).
- [3]. G. Bledt, P. M. Wensing, and S. Kim, "Policy-regularized model predictive control to stabilize diverse quadrupedal gaits for the MIT Cheetah," in IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2017.
- [4]. Wang L, Meng L, Kang R, Liu B, Gu S, Zhang Z, Meng F, Ming A. Design and dynamic locomotion control of quadruped robot with perception-less terrain adaptation. *Cyborg and Bionic Systems*. 2022 Feb 20.
- [5]. kih Muhamad F, Kim JS, Park JH. Learning Robust Perception Based Controller for Quadruped Robot. *IEEE Access*. 2023 Sep 1.
- [6]. F. Yang, C. Wang, C. Cadena, and M. Hutter, "iPlanner: Imperative path planning," 2023, arXiv:2302.11434.
- [7]. F. Kiani, A. Seyyedabbasi, R. Aliyev, M. U. Gulle, H. Basyildiz, and M. A. Shah, "Adapted-RRT: Novel hybrid method to solve three dimensional path planning problem using sampling and metaheuristicbased algorithms," *Neural Comput. Appl.*, vol. 33, pp. 15569–15599, Nov. 2021.
- [8]. A. Seyyedabbasi, R. Aliyev, F. Kiani, M. U. Gulle, H. Basyildiz, and M. A. Shah, "Hybrid algorithms based on combining reinforcement learning and metaheuristic methods to solve global optimization problems," *Knowl.-Based Syst.*, vol. 223, Jul. 2021, Art. no. 107044.
- [9]. J. Huang, J. Rojas, M. Zimmer, H. Wu, Y. Guan, and P. Weng, "Hyperparameter auto tuning in self-supervised robotic learning," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3537–3544, Apr. 2021.
- [10]. S. Nematzadeh, F. Kiani, M. Torkamanian-Afshar, and N. Aydin, "Tuning hyperparameters of machine learning algorithms and deep neural networks using metaheuristics: A bioinformatics study on biomedical and biological cases," *Comput. Biol. Chem.*, vol. 97, Apr. 2022, Art. no. 107619.
- [11]. Mehrizi, Kavan. "Quadrupedal robotic guide dog with vocal human-robot interaction." arXiv preprint arXiv:2111.03718 (2021).
- [12]. Cruz Ulloa C, Sánchez L, Del Cerro J, Barrientos A. Deep Learning Vision System for Quadruped Robot Gait Pattern Regulation. *Biomimetics*. 2023 Jul 3;8(3):289.
- [13]. Feng G, Zhang H, Li Z, Peng XB, Basireddy B, Yue L, Song Z, Yang L, Liu Y, Sreenath K, Levine S. Genloco: Generalized locomotion controllers for quadrupedal robots. In *Conference on Robot Learning* 2023 Mar 6 (pp. 1893-1903). PMLR.
- [14]. Muriga, Veronica. *Towards Reinforcement-Learning-based Robot Navigation with 3D Scene Graphs*. Diss. Massachusetts Institute of Technology, 2023.
- [15]. Cruz Ulloa C, Domínguez D, Del Cerro J, Barrientos A. A mixed-reality tele-operation method for high-level control of a legged-manipulator robot. *Sensors*. 2022 Oct 24;22(21):8146.

- [16]. Romero A, Delgado C, Zanzi L, Suárez R, Costa-Pérez X. Cellular-enabled Collaborative Robots Planning and Operations for Search-and-Rescue Scenarios. arXiv preprint arXiv:2403.09177. 2024 Mar 14.
- [17]. Liu J, Wang P, Li M, Guo W, Zha F, Sun L, Zheng P. A multiplicative noises and additive correlated noises cubature Kalman filter and its application in quadruped robot. IEEE Access. 2020 Sep 3;8:162290-301.
- [18]. Zhu X, Wang M, Ruan X, Chen L, Ji T, Liu X. Adaptive motion skill learning of quadruped robot on slopes based on augmented random search algorithm. Electronics. 2022 Mar 8;11(6):842.
- [19]. Zhu X, Wang M, Ruan X, Chen L, Ji T, Liu X. Adaptive motion skill learning of quadruped robot on slopes based on augmented random search algorithm. Electronics. 2022 Mar 8;11(6):842.
- [20]. Aractingi M, Léziart PA, Flayols T, Perez J, Silander T, Souères P. Controlling the Solo12 quadruped robot with deep reinforcement learning. scientific Reports. 2023 Jul 24;13(1):11945.
- [21]. Miki T, Lee J, Hwangbo J, Wellhausen L, Koltun V, Hutter M. Learning robust perceptive locomotion for quadrupedal robots in the wild. Science robotics. 2022 Jan 19;7(62):eabk2822.
- [22]. Zhang H, Yin J, Wang H. A Needs Learning Algorithm Applied to Stable Gait Generation of Quadruped Robot. Sensors. 2022 Sep 26;22(19):7302.
- [23]. Yan W, Pan Y, Che J, Yu J, Han Z. Whole-body kinematic and dynamic modeling for quadruped robot under different gaits and mechanism topologies. PeerJ Computer Science. 2021 Dec 16;7:e821.
- [24]. Meng J, Li Y, Li B. A dynamic balancing approach for a quadruped robot supported by diagonal legs. International Journal of Advanced Robotic Systems. 2015 Oct 14;12(10):142.
- [25]. You Y, Yang Z, Zou TA, Sui Y, Xu C, Zhang C, Xu H, Zhang Z, Han J. A new trajectory tracking control method for fully electrically driven quadruped robot. Machines. 2022 Apr 21;10(5):292.
- [26]. Li Z, Li B, Liang Q, Liu W, Hou L, Rong X. A quadruped robot obstacle avoidance and personnel following strategy based on ultra-wideband and three-dimensional laser radar. International Journal of Advanced Robotic Systems. 2022 Jul 28;19(4):17298806221114705.
- [27]. Shi Y, Li S, Guo M, Yang Y, Xia D, Luo X. Structural design, simulation and experiment of quadruped robot. Applied Sciences. 2021 Nov 12;11(22):10705.
- [28]. Spiceworks. (n.d.). What Is a Robot Operating System (ROS)? Meaning, Working, Applications, and Benefits. Retrieved March 3, 2024, from <https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-robot-operating-system/>
- [29]. LinkedIn. (2024, February 21). What are the best practices for ROS simulation and visualization skills in robotics? Retrieved July 18, 2024, from <https://www.linkedin.com/advice/1/what-best-practices-ros-simulation-visualization-skills-robotics-msgre#:~:text=using%20Gazebo%20APIs,->

[.4%20Use%20RViz%20and%20Gazebo,trajectories%2C%20markers%2C%20and%20transforms.](#)

- [30]. Elalawy, M. (2023, December 19). ROS architecture. *LinkedIn*. Retrieved July 18, 2024, from <https://www.linkedin.com/pulse/ros-architecture-muhamad-elalawy-osncf>
- [31]. ROS Wiki. (2018, August 8). ROS/Introduction. Retrieved July 18, 2024, from <https://wiki.ros.org/ROS/Introduction>
- [32]. macc-n. (n.d.). *ros\_unitree* [GitHub repository]. GitHub. Retrieved July 18, 2024, from [https://github.com/macc-n/ros\\_unitree](https://github.com/macc-n/ros_unitree)
- [33]. MathWorks. (n.d.). *Simultaneous Localization and Mapping (SLAM)*. Retrieved September 2, 2024, from <https://www.mathworks.com/discovery/slam.html>
- [34]. Ouster. (n.d.). *Introduction to SLAM (Simultaneous Localization and Mapping)*. Retrieved September 2, 2024, from <https://ouster.com/insights/blog/introduction-to-slam-simultaneous-localization-and-mapping>
- [35]. Unitree Robotics. (2020). *A1 software developer guide* (Version 1.0)
- [36]. Unitree Robotics. (2020). *A1 user manual* (Version 1.1). Unitree.