

UNIVERSITETI I PRISHTINËS
Fakulteti i Inxhinierisë Mekanike
Prishtinë

UNIVERSITETI I PRISHTINËS "HASAN PRISHTINA"
FAKULTETI I INXHINIERISË MEKANIKE
PRISHTINË

Pranuar me: 22-05-2026			
Nj. org.	Numër	Shtojca	Vlera
08	793	—	—

Në bazë të vendimit nr. 08/1124 të datës 04/07/2025, të Këshillit të Fakultetit të Inxhinierisë Mekanike në Prishtinë është formuar Komisioni në përbërje:

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

Për vlerësimin e punimit Master me titull: ” Zhvillimi i një manipuluesi robotik me 3-DOF për klasifikimin e objekteve duke përdorur Inteligjencën Artificiale dhe kinematikën inverse” të kandidatit Bachelor Altin Osmanaj.

Pas kontrollimit të punimit të lartpërmendur Komisioni jep këtë:

R A P O R T

Punimi Master me titull: “**Zhvillimi i një manipuluesi robotik me 3-DOF për klasifikimin e objekteve duke përdorur Inteligjencën Artificiale dhe kinematikën inverse**” është punuar në 9 (nëntë) kapituj, përfshirë kapitujt ndihmës Hyrjen, Përfundimin dhe Literaturën, të paraqitur me figura dhe ilustrime të nevojshme. Punimi ka 76 faqe, 55 figura dhe 5 tabela.

Në **Përmbledhje** kandidati **Bachelor Altin Osmanaj** ka paraqitur një hulumtim gjithëpërfshirës për zhvillimin e një manipuluesi robotik me 3-DOF për klasifikimin e objekteve duke përdorur Inteligjencën Artificiale dhe kinematikën inverse. Fillimisht është zhvilluar modeli mekanik i krahut robotik duke përcaktuar parametrat gjeometrikë dhe kufizimet kinematike të tij. Në vazhdim janë analizuar kinematika direkte dhe inverse për të mundësuar përcaktimin e saktë të pozicionit dhe orientimit të efektorit fundor. Për verifikimin e lëvizjeve dhe validimin e modeleve matematikore, sistemi është simuluar në platformën ROS2 duke përdorur mjediset Rviz dhe Gazebo. Pas përfundimit të simulimeve, janë prodhuar pjesët mekanike dhe janë zgjedhur komponentët elektronikë të nevojshëm për realizimin e prototipit fizik. Kontrolli i manipuluesit është mundësuar përmes një aplikacioni të dedikuar, i cili funksionon në mënyrë manuale dhe automatike, ku në modalitetin automatik Inteligenca Artificiale identifikon objekte me forma të ndryshme dhe përcakton koordinatat e tyre duke përdorur kinematikën inverse.

Kapitulli i parë përshkruan manipuluesit robotikë që janë sisteme të programueshme që imitojnë lëvizjet e krahut të njeriut për manipulimin e objekteve me saktësi dhe efikasitet. Të integruar me sensorë, kamera dhe Inteligjencë Artificiale, ata përdoren gjerësisht në industri të ndryshme për shkak të produktivitetit, sigurisë dhe fleksibilitetit të lartë.

Kapitulli i dytë shpjegon qëllimin e këtij punimi master që është projektimi dhe zhvillimi i një sistemi inteligjent mekatronik që integron një manipulues robotik 3-DOF me teknika të Inteligjencës Artificiale dhe vizionit kompjuterik për detektimin, klasifikimin dhe vendosjen automatike të objekteve sipas formës së tyre gjeometrike.

Kapitulli i tretë shpjegon një manipulues robotik që është një sistem mekanik që imiton lëvizjet e krahut të njeriut përmes komponentëve si nyjet, lidhjet, aktuatorët, sensorët dhe efektori fundor, duke mundësuar lëvizje të kontrolluara dhe të sakta në hapësirë. Përbërësit kryesorë si baza, lidhjet, nyjet, aktuatorët, sensorët dhe efektori fundor përcaktojnë stabilitetin, fleksibilitetin dhe performancën e përgjithshme të robotit. Kontrolli i tyre realizohet përmes sistemeve me qark të hapur ose të mbyllur, të mbështetura në teknika si PID, planifikimi i trajektores dhe kontrolli adaptiv, të cilat sigurojnë saktësi dhe stabilitet të lëvizjes. Sipas strukturës dhe shkallëve të lirisë, manipuluesit robotik klasifikohen në lloje të ndryshme si karteziante, cilindrike, sferike dhe të artikuluara, secili me karakteristika dhe aplikime të veçanta në industri.

Kapitulli i katërt paraqet manipuluesin robotik që është projektuar në SolidWorks, duke krijuar një model të saktë që përfshin kapësen dhe komponentët mekanikë, i optimizuar për stabilitet, funksionalitet dhe efikasitet gjatë manipulimit të objekteve, si dhe i verifikuar përmes simulimeve të ndryshme nën ngarkesa. Pas dizajnit, pjesët mekanike janë përpunuar me prerje dhe lakim me laser, ndërsa disa komponentë janë bashkuar me saldime me laser për të siguruar qëndrueshmëri strukturore, veçanërisht në folëtë e kushinetave. Sistemi elektrik përbëhet nga Arduino UNO si njësi kryesore e kontrollit, një motor stepper Nema 17 për bazën rrotulluese, tre servo motorë KST BLS825 për lëvizjet e krahut dhe kapësen, si dhe driveri TB6600 dhe konvertuesi DC-DC

për menaxhimin dhe stabilizimin e furnizimit me energji dhe kontrollit të motorëve. Pas përzgjedhjes dhe integritit të komponentëve mekanikë dhe elektrikë, është realizuar montimi i plotë i sistemit sipas vizatimeve teknike dhe skemës elektrike të projektuar.

Kapitulli i pestë shtjellon kinematikën në robotikë që është një nga fushat bazë që studion lëvizjen e manipuluesve robotik pa marrë parasysh forcat që e shkaktojnë atë, duke lidhur këndet e nyjeve me pozicionin dhe orientimin e efektorit fundor në hapësirë. Kinematika direkte përdoret për të përcaktuar pozicionin e efektorit fundor nga vlerat e njohura të këndeve të nyjeve, duke u modeluar zakonisht përmes metodës së Denavit–Hartenberg dhe matricave transformuese homogjene që përshkruajnë marrëdhëniet mes segmenteve të robotit. Në të kundërt, kinematika inverse shërben për të llogaritur këndet e nyjeve kur pozicioni i dëshiruar i efektorit fundor është i njohur, duke përdorur marrëdhënie gjeometrike dhe trigonometrike për të zgjidhur konfigurimin e robotit. Kombinimi i kinematikës direkte dhe inverse mundëson planifikimin dhe kontrollin e saktë të lëvizjes së robotit në hapësirë tredimensionale.

Në **Kapitullin e gjashtë** paraqitet simulimi në ROS2, i realizuar përmes RViz2 dhe Gazebo, ka luajtur një rol thelbësor në zhvillimin e sistemit duke mundësuar verifikimin e algoritmeve të kinematikës dhe lëvizjes përpara ndërtimit fizik të robotit. Përdorimi i ekosistemit ROS2 së bashku me MoveIt2 ka lejuar planifikimin dhe testimin e trajektoreve në një mjedis virtual të sigurt, duke reduktuar rrezikun e dëmtimeve dhe duke përshpejtuar procesin e iterimit të dizajnit. Modeli i robotit është ndërtuar në format XACRO/URDF, i integruar me Gazebo për simulim fizik dhe me RViz2 për vizualizim 3D në kohë reale të gjendjes së nyjeve dhe transformimeve. Në këtë kuadër, janë implementuar edhe algoritmet e kinematikës dhe kontrollit të lëvizjes, të cilat janë testuar dhe validuar para aplikimit në sistemin real të manipuluesit robotik.

Kapitulli i shtatë trajton detektimin dhe klasifikimin e objekteve duke përdorur teknika të Inteligjencës Artificiale, ku është aplikuar algoritmi YOLO (You Only Look Once) për njohjen e objekteve në kohë reale. YOLO është një qasje moderne e vizionit kompjuterik që e trajton detektimin si një problem të vetëm regresioni, duke parashikuar drejtpërdrejt kutitë rrethuese dhe klasat e objekteve nga një kalim i vetëm mbi imazh, gjë që e bën shumë më të shpejtë se metodat tradicionale me dy hapa.

Funksionimi i modelit bazohet në ndarjen e imazhit në një rrjet qelizash, ku secila qelizë parashikon një numër kutish rrethuese dhe probabilitetet përkatëse të klasave. Në këtë mënyrë, sistemi mund të identifikojë dhe lokalizojë objekte të ndryshme në mënyrë simultane, duke përdorur koordinatat e qendrës, dimensionet dhe vlerën e besueshmërisë për secilin objekt të detektuar.

Kapitulli i tetë përshkruan zhvillimin e aplikacionit të kontrollit të manipuluesit robotik të implementuar në Python me Tkinter për ndërfaqen grafike. Aplikacioni shërben si qendër e integritit të sistemit, duke mundësuar komunikimin me Arduino, kontrollin manual të nyjeve, integrimin me YOLOv5, kalibrimin e kamerës dhe modalitetin automatik pick-and-place. Për përdorim praktik, është pakëtuar si executable me PyInstaller.

Struktura e aplikacionit bazohet në klasën RoboticArmUI, e cila menaxhon ndërfaqen, komunikimin serial dhe funksionet kryesore të sistemit, përfshirë lidhjen me Arduino, kontrollin e servo-motorëve, sliderët për kontroll manual dhe modalitetin automatik. Ndërfaqja është ndërtuar në stil dark mode dhe është e ndarë në dy pjesë: kontrollet e robotit dhe Arduino në të majtë, si dhe kamerën, sistemin dhe monitorin serial në të djathtë.

Kontrolli manual realizohet përmes një protokolli serial M:base, shoulder, elbow, gripper, i cili dërgon direkt vlerat te Arduino për kontrollin e servo-motorëve dhe kapësit të robotit. Kapësi ka dy gjendje të kalibruara, të hapur dhe të mbyllur.

Në modalitetin automatik, sistemi përdor YOLOv5 për detektimin e objekteve në kohë reale dhe transformon koordinatat e kamerës në koordinata reale përmes matricës së kalibrimit. Aplikacioni dërgon komandat te Arduino dhe pret konfirmimin përpara se të vazhdojë ciklin e ri të detektimit.

Kalibrimi i kamerës realizohet me katër pika referuese për llogaritjen e matricës homogjene që lidh imazhin me hapësirën reale të robotit. Kjo siguron konvertim të saktë të koordinatave të pikselëve në koordinata fizike.

Komunikimi me Arduino bëhet përmes lidhjes seriale, duke mbështetur komandat manuale dhe ato automatike nga sistemi i vizionit, duke mundësuar integrim të plotë mes ndërfaqes, inteligjencës artificiale dhe kontrollit fizik të robotit.

Në **Kapitullin e nëntë** paraqiten testimet dhe analiza e performancës së sistemit, duke përfshirë gjithë fluksin nga detektimi vizual deri te vendosja fizike e objektit. Procesi fillon me kapjen e imazhit nga kamera dhe përpunimin përmes OpenCV, ku YOLOv5 identifikon objektet dhe klasat përkatëse në kohë reale. Më pas, koordinatat e detektuara transformohen në koordinata robotike përmes kalibrimit të kamerës dhe dërgohen te nodi C++ për llogaritjen e kinematikës inverse dhe gjenerimin e këndeve të nyjeve.

Sistemi vazhdon me planifikimin dhe ekzekutimin e trajektores përmes MoveIt2 dhe ros2_control, duke mundësuar kapjen, transportimin dhe vendosjen e objektit në destinacionin përkatës, pas së cilës roboti kthehet në pozicionin fillestar për ciklin e radhës. Krahasimi midis simulimit në Gazebo dhe robotit fizik tregon se simulimi ka saktësi më të lartë dhe kohë ekzekutimi më të shpejta, ndërsa roboti real ndikohet nga faktorët fizikë si fërkimi dhe vonesat mekanike.

Në sistemin real vërehen devijime më të mëdha në kinematikë dhe kohë më të gjata të ciklit Pick & Place, si dhe një përqindje e vogël e humbjes së objekteve gjatë transportit. Megjithatë, sistemi i vizionit YOLO ruan një saktësi të lartë në të gjitha klasat e objekteve, duke treguar stabilitet të mirë në kushte reale. Në përgjithësi, rezultatet konfirmojnë se sistemi i zhvilluar është i qëndrueshëm dhe i përshtatshëm për aplikime praktike.

Në fund është dhënë **Përfundimi**, duke theksuar se punimi i diplomës është realizuar me sukses projektimi, zhvillimi dhe testimi i një sistemi të integruar mekatronik, i cili bashkon një manipulues robotik me tre shkallë lirie (3-DOF) me metoda bashkëkohore të Inteligjencës Artificiale për identifikimin dhe vendosjen autonome të objekteve sipas formës së tyre gjeometrike.

Pastaj pason **Literatura** me listën e burimeve që janë përdorur gjatë përgatitjes së punimit, duke përfshirë libra, artikuj shkencorë, raporte, standarde, faqe interneti, dhe çdo material tjetër që ka shërbyer si bazë për informacionin, analizat dhe argumentet e paraqitura.

P Ë R F U N D I M

Punimi Master me titull: “**Zhvillimi i një manipuluesi robotik me 3-DOF për klasifikimin e objekteve duke përdorur Inteligjencën Artificiale dhe kinematikën inverse**” të kandidatit

Bachelor Altin Osmanaj, në bazë të rezultateve të paraqitura në këtë punim, mund të përfundohet se sistemi i zhvilluar përfaqëson një zgjidhje të plotë dhe funksionale në fushën e robotikës dhe inteligjencës artificiale. Integrimi i manipuluesit robotik 3-DOF me sistemin e vizionit YOLO dhe kontrollin përmes ROS2 ka mundësuar realizimin e suksesshëm të detektimit, klasifikimit dhe manipulimit autonom të objekteve në kohë reale.

Testimet në simulim dhe në sistemin fizik kanë treguar performancë të qëndrueshme, me saktësi të lartë në ekzekutimin e trajektoreve dhe klasifikimin e objekteve, pavarësisht devijimeve të vogla të pritura në kushte reale. Krahasimi mes Gazebo dhe robotit fizik konfirmon vlefshmërinë e modelit të zhvilluar dhe korrektësinë e implementimit të kinematikës dhe kontrollit.

Në përgjithësi, punimi vlerësohet si i suksesshëm, pasi arrin objektivat e përcaktuara dhe demonstroi një integrim të mirë të teorisë me implementimin praktik, duke ofruar një bazë të fortë për zhvillime të mëtejshme në sistemet autonome robotike.

Komisioni Vlerësues e konsideron këtë punim të përgatitur në mënyrë të shkëlqyer, të strukturuar mirë dhe të pasuruar me ilustrime, fotografi dhe figura që e bëjnë atë një tërësi të plotë dhe profesionale.

REKOMANDIM

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 dhe i plotëson në tërësi kriteret e kërkuara për një punim Master. Prandaj, Komisioni propozon, që punimi Master me titull: **“Zhvillimi i një manipuluesi robotik me 3-DOF për klasifikimin e objekteve duke përdorur Inteligjencën Artificiale dhe kinematikën inverse”** të kandidatit **Bachelor Altin Osmanaj**, ta paraqesë këtë raport për diskutim publik.

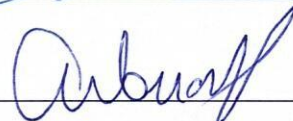
Prishtinë, Maj 2026

Komisioni:

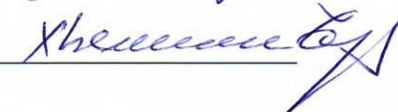
1. Prof. Dr. Ahmet Shala, kryetar



2. Prof. Dr. Arbnor Pajaziti, mentor/anëtar



3. Prof. Asoc. Dr. Xhevahir Bajrami, anëtar



UNIVERSITETI I PRISHTINËS “HASAN PRISHTINA”

FAKULTETI I INXHINIERISË MEKANIKE

PROGRAMI: MEKATRONIKË



PUNIM DIPLOME

TEMA: Zhvillimi i një manipuluesi robotik me 3-DOF për klasifikimin e objekteve duke përdorur Inteligjencën Artificiale dhe kinematikën inverse

Mentori:

Prof. Dr. Arbnor Pajaziti

Kandidati:

BSc. Altin Osmanaj

Prishtinë, Maj 2026

Përmbajtja

Lista e Figurave	3
Lista e Tabelave	4
Lista e Shkurtesave	4
Abstrakt	5
Abstract	7
1 Hyrje	8
2 Qëllimi i hulumtimit dhe objektivat	9
3 Baza teorike mbi sistemet robotike	11
3.1 Sensorët dhe sistemet e reagimit	16
3.2 Sistemet e kontrollit për manipuluesit robotikë	16
3.2.1 Sistemet e kontrollit me qark të hapur	16
3.2.2 Sistemet e kontrollit me qark të mbyllur	17
3.3 Teknikat kryesore të kontrollit për manipuluesit robotikë	18
3.3.1 Kontrolli PID (Kontrolli Proporcional-Integral-Derivativ).....	18
3.3.2 Planifikimi i Trajektores.....	19
3.3.3 Kontrolli i Impedancës (Kontrolli i Forcës).....	20
3.3.4 Kontrolli Adaptiv	20
3.4 Projektimi i një sistemi efektiv kontrolli	21
3.5 Llojet e manipuluesve robotik.....	21
3.5.1 Manipuluesit robotik karteziak	21
3.5.2 Manipuluesit robotik cilindrik	22
3.5.3 Manipuluesit robotik sferik	23
3.5.4 Manipuluesit robotik të artikuluar	24
4 Komponentët mekanik dhe elektrik	26
5 Kinematika direkte, inverse dhe lëvizja e robotit.....	31
5.1 Kinematika direkte	31
5.1.1 Metoda e Denavit Hartenberg-ut	31
5.1.2 Matrica transformuese homogjene.....	32
5.1.3 Llogaritja e kinematikës direkte për manipuluesin robotik me 3-DOF	33
5.2 Kinematika inverse	36

5.2.1	Llogaritja e kinematikës inverse për manipuluesin robotik me 3-DOF	37
6	Simulimi në ROS2 duke përdorur RViz dhe Gazebo	40
6.1	Ekosistemi ROS2 i përdorur	40
6.2	Modeli URDF/XACRO i manipuluesit robotik	41
6.3	Struktura kinematike: lidhjet dhe nyjet.....	41
6.3.1	Nyja e krahut — joint_2 dhe rotacioni i planit vertikal	44
6.4	Parametrat fizik të lidhjeve	46
6.5	Vizualizimi me RViz2	46
6.6	Simulimi fizik me Gazebo	47
6.7	Planifikimi i lëvizjes me MoveIt2 dhe rezultatet	48
6.7.1	Implementimi i Pick & Place në C++ me MoveIt2	49
7	Detektimi dhe klasifikimi objekteve me anë të Inteligjencës Artificiale	56
7.1	Si funksionon YOLO?	56
7.2	Vlerësimi i performancës	57
7.3	Non-Maximum Supression (NMS)	58
7.4	Trajnimi i një modeli YOLO	59
8	Zhvillimi i aplikacionit të kontrollit	65
8.1	Arkitektura e aplikacionit.....	65
8.2	Ndërfaqja grafike – pamja kryesore	66
8.3	Kontrolli manual i nyjeve dhe protokolli serial.....	67
8.4	Kontrolli automatik dhe integrimi me YOLO	67
8.5	Kalibrimi i kamerës	69
8.6	Komunikimi me Arduino	70
9	Testime dhe krahasime	71
9.1	Krahasimi i modelit të simuluar me robotin real.....	73
	Përfundimi	74
	Deklarata e origjinalitetit/ Autorësisë.....	Error! Bookmark not defined.
	Referencat.....	75
	Referencat e figurave	76

Lista e Figurave

Figura 1: Llojet e nyjeve të manipuluesve robotik.	13
Figura 2: Llojet e efektorëve fundor të manipuluesve robotik [1].	15
Figura 3: Sistemi i kontrollit me qark të hapur.	17
Figura 4: Sistemi i kontrollit me qark të mbyllur.	18
Figura 5: Kontrolli përmes PID.	19
Figura 6: Planifikimi i trajektores.	19
Figura 7: Kontrolli me anë të impedancës.	20
Figura 8: Kontrolli adaptiv.	20
Figura 9: Manipuluesi robotik kartezian.	22
Figura 10: Manipuluesi robotik cilindrik.	23
Figura 11: Manipuluesi robotik sferik.	24
Figura 12: Manipuluesi robotik i artikulluar.	25
Figura 13: Projektimi i manipuluesit robotik në SolidWorks [2].	26
Figura 14: Manipuluesi robotik i projektuar [3].	26
Figura 15: Lidhjet dhe efektori fundor të projektuara në SolidWorks.	27
Figura 16: Përgatitja dhe prerja me laser e pjesëve përbërëse të manipuluesit robotik.	27
Figura 17: Stepper motori Nema17 [4].	28
Figura 18: Servo motori KST BLS825 [5].	28
Figura 19: Ngasësi TB6600 [6].	29
Figura 20: Konvertues DC-DC me ulje tensioni [7].	29
Figura 21: Montimi i manipuluesit robotik.	30
Figura 22: Skema e lidhjeve elektrike.	30
Figura 23: Aprovimi i sistemeve koordinatave sipas D-H për manipuluesin robotik.	34
Figura 24: Pozicioni i efektorit fundor përgjatë x dhe y.	36
Figura 25: Pozicioni i efektorit fundor përgjatë x dhe z.	36
Figura 26: Distanca nga baza deri tek efektori fundor.	37
Figura 27: Këndi në rrafshin $x - y$.	37
Figura 28: Këndet në rrafshin $x^2 + y^2 - z$.	38
Figura 29: Sistemi Operativ për Robotë (Robot Operating System) [8].	40
Figura 30: Dëfinimi i të gjitha sistemeve koordinatave të nevojshme për eksport në URDF.	42
Figura 31: Përdorimi i veglës ExportToURDF.	42
Figura 32: Rrotullimi i sistemit koordinatave të bazës për 90° rreth x-it.	45
Figura 33: Vizualizimi me RViz2.	47
Figura 34: Simulimi i manipuluesit robotik në Gazebo para se të aktivizohen kontrollorët.	47
Figura 35: Simulimi pasi janë aktivizuar kontrollorët.	48
Figura 36: Pozita fillestare e manipuluesit robotik.	54
Figura 37: Testimi me komandën circle 10.5 15.2 0.0.	54
Figura 38: Pozita 20 -15 0.	55
Figura 39: Pozita e lirit të objektit katërkëndësh.	55

Figura 40: Metodologjia e funksionimit të YOLO.	57
Figura 41: Para dhe pas Non-Maximum Supression.	58
Figura 42: Ndarja e të dhënave në train, test dhe validation set.	59
Figura 43: Matrica e konfuzionit (Confusion Matrix) për modelin e trajnuar.	61
Figura 44: Lakorja Precision-Recall.	61
Figura 45: Lakorja F1.	62
Figura 46: Shembuj të objekteve të zbuluara.	63
Figura 47: Rezultatet përfundimtare të modelit të trajnuar.	64
Figura 48: Aplikacioni i kontrollit të manipuluesit robotik.	65
Figura 49: Dritarja kryesore e aplikacionit.	66
Figura 50: Lidhja me Arduino në COM5.	67
Figura 51: Dritarja e kamerës pasi klikohet në butonin Start Auto Mode.	68
Figura 52: Sistemet koordinatave.	68
Figura 53: Dritarja e kalibrimit të kamerës.	70
Figura 54: Gjenerimi dhe testimi i trajektoreve.	71
Figura 55: Testimi i manipuluesit robotik.	72

Lista e Tabelave

Tabela 1: Parametrat e Denavit Hartenberg-ut dhe përshkrimi i tyre.	32
Tabela 2: Parametrat e D-H për manipuluesin robotik.	34
Tabela 3: Parametrat e eksportuar nga SolidWorks për gjashtë nyjet e definuara.	44
Tabela 4: Parametrat fizik të lidhjeve të eksportuar nga SolidWorks.	46
Tabela 5: Krahasimi i modelit të simuluar me robotin real.	73

Lista e Shkurtesave

DoF – Degrees of Freedom (shkallët e lira të lëvizjes)
ROS – Robot Operating System (sistemi operativ i robotëve)
YOLO – You Only Look Once (algoritëm për detektimin e imazheve)
PCB – Printed Circuit Board (pllaka elektronike)
GUI – Graphical User Interface (ndërfaqe grafike e përdoruesit)
TCP - Transmission Control Protocol (protokolli i kontrollit të transmetimit të të dhënave)

Deklarata e origjinalitetit/ Autorësisë

Me anë të kësaj deklaratë, unë Altin Fadil Osmanaj me përgjegjësi të plotë deklaroj se ky punim është puna ime personale dhe nuk është prezantuar asnjëherë përpara këtij institucioni apo ndonjë institucioni tjetër. Ndërsa literatura dhe burimet e tjera që janë nevojitur rreth punimit janë të cituara dhe të listuara te referencat.

Prishtinë, Maj 2026

Studenti - nënshkrimi

Abstrakt

Në këtë punim diplome paraqitet zhvillimi i një manipuluesi robotik me tre shkallë lirie (3-DOF) i aftë të kryejë klasifikimin dhe vendosjen automatike të objekteve në vende të caktuara duke përdorur Inteligjencën Artificiale dhe kinematikën inverse.

Fillimisht është realizuar dizajnimi mekanik i krahut robotik, ku janë përcaktuar dimensionet gjeometrike dhe kufizimet kinematike. Më pas janë llogaritur kinematika direkte dhe kinematika inverse për përcaktimin e pozicionit dhe orientimit të efektorit fundor.

Sistemi është simuluar në ROS2 (Robot Operating System 2) duke përdorur mjedisin Rviz dhe Gazebo, me qëllim verifikimin e lëvizjeve dhe validimin e modeleve matematikore. Pas fazës së simulimit, është bërë prodhimi i pjesëve mekanike dhe përzgjedhja e komponentëve elektronike për ndërtimin e sistemit fizik.

Kontrolli i manipuluesit është realizuar përmes një aplikacioni të zhvilluar, i cili ofron dy mënyra funksioni: manual dhe automatik. Kontrolli automatik përmes Inteligjencës Artificiale detekton objektet e formave të ndryshme dhe përcakton koordinatat e tyre përmes kinematikës inverse.

Rezultatet e testeve fizike janë krahasuar me ato të simulimeve në ROS2, duke dëshmuar përputhje të mirë dhe efikasitet në klasifikimin e objekteve. Ky punim kontribuon në fushën e automatizimit dhe robotikës inteligjente duke demonstruar një zgjidhje të integruar, të thjeshtë dhe me kosto të ulët.

Abstract

This thesis presents the development of a three degrees-of-freedom (3-DOF) robotic manipulator capable of performing automatic classification and placement of objects in certain locations using Artificial Intelligence and inverse kinematics.

Initially, the mechanical design of the robotic arm was carried out, where the geometric dimensions and kinematic limitations were defined. Direct kinematics and inverse kinematics were then calculated to determine the position and orientation of the end effector.

The system is simulated in ROS2 (Robot Operating System 2) using the Rviz and Gazebo environment, in order to verify the movements and validate the mathematical models. After the simulation phase, the production of mechanical parts and the selection of electronic components for the construction of the physical system is done.

Manipulator control is realized through a developed application, which offers two modes of operation: manual and automatic. Automatic control through Artificial Intelligence detects objects of different shapes and determines their coordinates through inverse kinematics.

The results of physical tests have been compared with those of simulations in ROS2, providing good agreement and efficiency in object classification. This thesis contributes to the field of automation and intelligent robotics by demonstrating an integrated, simple and low-cost solution.

1 Hyrje

Robotika është bërë një nga teknologjitë më transformuese të epokës moderne, dhe në qendër të saj qëndron manipuluesi robotik. Një manipulues robotik është një krah mekanik i programueshëm që imiton lëvizjet e krahut të njeriut. Qëllimi i tij kryesor është të lëvizë mjete ose objekte me saktësi, shpejtësi dhe besueshmëri. Me kalimin e viteve, manipuluesit robotik janë zhvilluar nga pajisje të thjeshta mekanike në sisteme inteligjente, të pajisura me kamera, sensorë dhe inteligjencë artificiale. Sot, ata përdoren pothuajse në çdo industri kryesore nga automobilizimi dhe prodhimi, te shëndetësia, logjistika dhe madje edhe eksplorimi hapësinor. Një manipulues robotik përbëhet nga komponentë mekanikë dhe sisteme kontrolli që së bashku e bëjnë të funksionojë si një krah njeriu. Udhëhiqet nga kompjuterë ose mikrokontrollerë, që përkthejnë komandat në sinjale për motorët. Një nga përmirësimet moderne më të rëndësishme është integrimi me kamera dhe sisteme të vizionit kompjuterik. Roli i kamerave në vizionin kompjuterik është që të shërbejnë si sytë e robotit, duke ofruar informacion mbi pozicionin dhe orientimin e objekteve dhe zbulimi i tyre duke përdorur algoritmet e Inteligjencës Artificiale. Përfitimet kryesore të manipuluesve robotik në industri janë:

- Produktivitet më i lartë – punojnë pa ndërprerje.
- Saktësi dhe përsëritshmëri.
- Siguri – shmangin rrezikun për punëtorët.
- Kosto më të ulëta në plan afatgjatë.
- Fleksibilitet – ri programim për detyra të ndryshme.
- Integrim me AI – përshtatje në kohë reale.

2 Qëllimi i hulumtimit dhe objektivat

Ky punim diplome ka si qëllim kryesor projektimin, zhvillimin dhe vlerësimin e një sistemi të integruar mekatronik i cili kombinon një manipulues robotik me tre shkallë lirie (3-DOF) me teknikat moderne të Inteligjencës Artificiale për kryerjen automatike të detyrës së klasifikimit dhe vendosjes së objekteve sipas formës gjeometrike. Sistemi synon të demonstrojë se si robotika dhe perceptimi vizual i bazuar në algoritme të të nxënimit të thellë (deep learning) mund të integrohen në mënyrë efikase për të automatizuar procese industriale të renditjes.

Motivimi për zhvillimin e këtij sistemi rrjedh nga nevoja gjithnjë e në rritje e industrisë moderne për sisteme inteligjente dhe autonome, të cilat mund të kryejnë detyra të përsëritura me saktësi të lartë dhe kosto të ulët operacionale. Integrimi i vizionit kompjuterik me kontrollin robotik paraqet një sfidë inxhinierie shumëdimensionale, duke prekur fusha si mekanika, elektronika, algoritmet e kontrollit dhe inteligjenca artificiale.

Për të realizuar këtë qëllim, janë vendosur objektivat e mëposhtme specifike:

1. Projektimi dhe ndërtimi fizik i një manipuluesi robotik me 3-DOF, duke përdorur SolidWorks dhe teknikën e sheet metal për realizimin e komponentëve mekanikë.
2. Zhvillimi dhe verifikimi i modelit matematikor të kinematikës direkte dhe inverse duke aplikuar konventën e Denavit-Hartenberg-ut. Llogaritja e matricave transformuese homogjene dhe validimi numerik i algoritmeve me vlera konkrete të këndeve dhe pozicioneve.
3. Trajnimi dhe implementimi i një modeli të bazuar në algoritmin YOLO (You Only Look Once) për detektimin dhe klasifikimin e objekteve sipas formës gjeometrike: rreth, trekëndësh dhe drejtkëndësh. Saktësia e klasifikimit është vendosur si objektiv mbi 90%.
4. Zhvillimi i një pipeline-i të integruar vizion-kontroll: kamera kap imazhin, modeli YOLO identifikon objektin dhe llogarit koordinatat e tij, kinematika inverse llogarit këndet e njejve, dhe komanda dërgohet serialisht te Arduino për ekzekutim të lëvizjes fizike.
5. Zhvillimi i një aplikacioni kontrolli në Python me ndërfaqe grafike (Tkinter) që ofron dy mënyra funksionimi: mënyrën manuale për kontroll direkt të secilës nyje dhe mënyrën automatike të drejtuar nga Inteligjenca Artificiale për pick-and-place autonome.

6. Simulimi i plotë i sistemit robotik në ROS2 (Robot Operating System 2) duke përdorur MoveIt për planifikimin e lëvizjes dhe Gazebo për simulimin fizik, me qëllim verifikimin e algoritmeve para implementimit fizik.
7. Testimi eksperimental i sistemit të plotë dhe krahasimi i rezultateve fizike me ato të simulimit, duke matur parametrat kryesorë: gabimin e pozicionimit, saktësinë e klasifikimit dhe kohën e ciklit marrje dhe vendosje (pick-and-place).

Dizajni fillestar i krahut robotik dhe implementimi bazë i sistemit për detektimin dhe klasifikimin e objekteve janë prezantuar paraprakisht në punimin shkencor “Automated Robotic Arm for Object Detection and Classification”, të publikuar në konferencën MECO 2025. Kontributi kryesor i këtij punimi diplome konsiston në integrimin e plotë të dy teknologjive: vizionit kompjuterik të bazuar në YOLO dhe kontrollit robotik të bazuar në kinematikë inverse, duke realizuar një sistem funksional dhe të testueshëm fizikisht. Sistemi i propozuar synon të jetë i replikueshëm dhe i arritshëm me komponentë me kosto të ulët, duke e bërë atë të vlefshëm si mjet edukativ dhe si prototip për aplikime industriale të lehta.

3 Baza teorike mbi sistemet robotike

Një manipulues robotik është një sistem mekanik i projektuar për të imituar lëvizjen e një krahu të njeriut. Ai e arrin këtë nëpërmjet njeve, lidhjeve, sensorëve, aktuatorëve dhe efektorëve fundorë, që e ndihmojnë atë të kryejë lëvizje të kontrolluara në shumë akse. Këto sisteme janë kyçe në automatizim sepse ato mund të trajtojnë me saktësi përpunimin e materialeve, montimin dhe madje edhe kirurgjinë robotike. Ato gjithashtu janë të domosdoshme në detyra të përsëritura dhe të rrezikshme që kërkojnë saktësi përtej aftësive njerëzore. Ajo që i bën manipuluesit robotikë vërtet të rëndësishëm është shkathtësia e tyre. Me shkallë lirie të personalizueshme (DoF) dhe sisteme kontrolli të drejtuara me precizitet të lartë, ato mund të trajtojnë gjithçka. Për të arritur këtë precizitet, algoritmet dhe sistemet e kontrollit llogarisin këndet e njeve dhe trajektoret e lëvizjes në kohë reale. Nga ana tjetër, teknikat si kinematika inverse sigurojnë precizitet dhe lëvizje të qetë. Për të kuptuar më mirë funksionalitetin e manipuluesve robotikë, është thelbësore të shqyrtohen komponentët e tyre kryesore. Këto përfshijnë bazën, lidhjet, nyjet, aktuatorët, sensorët dhe efektorët fundorë. Do komponent luan një rol vendimtar në performancën dhe aftësitë e përgjithshme të robotit [1].

Baza

Baza është themeli i një manipuluesi robotik. Ajo e mban të gjithë sistemin të qëndrueshëm, duke siguruar që roboti të mos anohet, të mos vibrojë ose të mos zhvendoset gjatë funksionimit. Në sistemet stacionare, si ato të përdorura në prodhim, baza është e fiksuar në dyshe me ose në një platformë të fortë për stabilitet maksimal. Për robotët që duhet të lëvizin midis stacioneve – siç janë robotët e automatizimit të magazinave – baza mund të montohet në rrota ose shina për lëvizshmëri. Dizajni i bazës është thelbësor. Ajo duhet të përballojë peshën e robotit, të marrë parasysh gamën e lëvizjes së tij dhe të mbështesë forcat e gjeneruara gjatë lëvizjes – veçanërisht në robotët industrialë që bartin ngarkesa të rënda. Për shembull, një robot i projektuar për të ngritur pjesë automobilistike kërkon një bazë të fortë që mund të absorbojë vibrimet dhe të mbështesë peshën e manipuluesit. Nga ana tjetër, një robot bashkëpunues i lehtë që ndihmon njerëzit mund të përdorë një bazë më të vogël dhe më fleksibilitet për të manovruar lirshëm.

Lidhjet

Lidhjet janë segmentet e ngurta që formojnë strukturën e robotit, duke lidhur nyejt dhe duke përcaktuar formën e përgjithshme të robotit. Çdo lidhje luan një rol kyç në përcaktimin e hapësirës së punës, gamës së lëvizjes dhe forcës së robotit. Dizajni i këtyre lidhjeve është kritik. Lidhjet më të gjata i lejojnë robotit të zgjatet më tej, duke rritur hapësirën e tij të punës. Megjithatë, lidhjet më të gjata mund të sjellin fleksibilitet të padëshiruar, duke zvogëluar saktësinë. Anasjelltas, lidhjet më të shkurtra përmirësojnë stabilitetin dhe kontrollin, por kufizojnë shtrirjen e robotit. Zgjedhja e materialit gjithashtu ka rëndësi. Materialet e lehta si alumini ose fibra e karbonit përmirësojnë shpejtësinë dhe efikasitetin, ndërsa materialet më të forta si çeliku janë ideale për robotët e rëndë që trajtojnë ngarkesa të mëdha. Për shembull, një krah robotik i përdorur në trajtimin e materialeve mund të ketë lidhje të lehta për të përmirësuar shpejtësinë, ndërsa një robot saldimit në një konfigurim robotësh industrialë mund të përdorë lidhje më të rënda për stabilitet dhe forcë më të madhe.

Nyjet

Nyjet janë pjesët lëvizëse që lidhin lidhjet e robotit dhe lejojnë atë të përkulet, të rrotullohet ose të zgjatet. Lloji dhe numri i nyejve përcaktojnë shkallët e lirisë (DoF) të robotit – një faktor kyç në përcaktimin e fleksibilitetit dhe gamës së lëvizjes së tij. Ekzistojnë tre lloje kryesore të nyejve në manipulatorët robotik:

- Nyjet rrotulluese: Këto lejojnë rrotullimin rreth një boshti të vetëm, ngjashëm me një bërryl njerëzor. Ato janë të zakonshme në robotët e artikulluar dhe janë ideale për detyra që kërkojnë lëvizje precize, siç është saldimit ose montimi.
- Nyjet prizmatike: Këto mundësojnë lëvizjen lineare përgjatë një boshti të vetëm, ngjashëm me një sirtar që hapet me rrëshqitje. Ato shpesh gjenden në robotët kartezianë ose sistemet e marrjes dhe vendosjes (pick and place) që kërkojnë lëvizje në vijë të drejtë.
- Nyjet sferike (ball-and-socket joints): Këto sigurojnë lëvizje rrotulluese përgjatë boshteve të shumta, ngjashëm me një krah të njeriut. Ndërsa më pak të zakonshme, ato ofrojnë shkathtësi të shkëlqyer në robotët që kanë nevojë për lëvizje fleksibile dhe shumë drejtimesh.

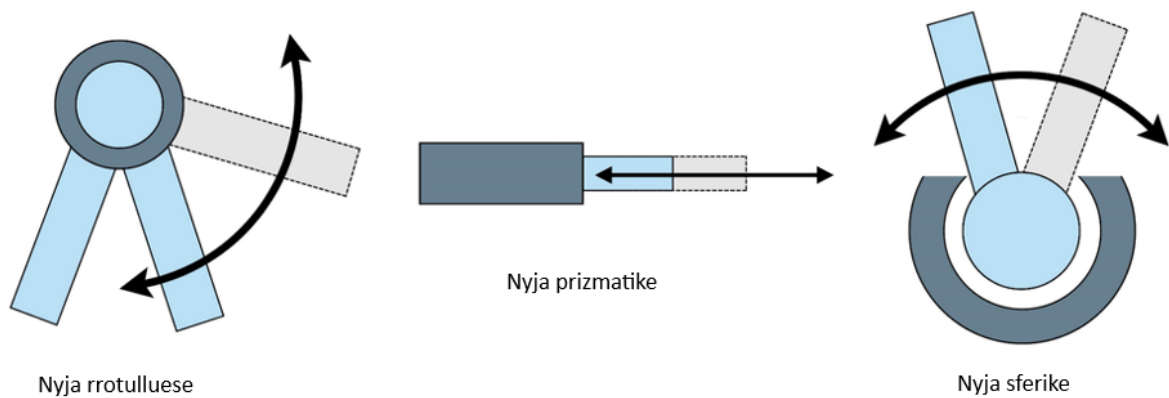


Figura 1: Llojet e nyjeve të manipuluesve robotik.

Numri i nyjeve ndikon drejtpërdrejt në DoF të robotit. Për shembull, një robot me gjashtë DoF mund të lëvizë me fleksibilitet mbresëlënës, duke imituar lëvizjen e një krahu njerëzor. Ndërkohë, një robot me vetëm tre nyje do të ketë një gamë lëvizjeje më të thjeshtë dhe më të kufizuar, por mund të jetë i dhurë në detyra përsëritëse dhe me shpejtësi të lartë. Zgjedhja e kombinimit të duhur të nyjeve është kritike për përmbushjen e objektivave të performancës – qoftë për montimin e elektronikës, trajtimin e materialeve delikate apo operimin në kirurgjinë robotike. Nyjet sferike ofrojnë lëvizje rrotulluese rreth boshteve të shumta, duke lejuar një shkallë të lartë fleksibiliteti dhe shkathhtësie. Këto nyje janë më pak të zakonshme në manipulorët robotik, por mund të gjenden në disa aplikime të specializuara, siç janë duart robotike ose robotët humanoid, ku është e nevojshme një gamë e gjerë lëvizjeje. Kur projektohen nyje për një manipulor robotik, duhet të merren në konsideratë faktorë të tillë si diapazoni i lëvizjes, kapaciteti i ngarkesës dhe rezistenca ndaj konsumit. Zgjedhja e llojit dhe dizajnit të nyjeve do të ndikojë drejtpërdrejt në performancën, fleksibilitetin dhe aftësitë e përgjithshme të robotit, duke e bërë atë një aspekt kritik të procesit të projektimit të robotit.

Aktuatorët

Aktuatorët janë muskujt e një manipulatori robotik – ata krijojnë lëvizje duke vënë në lëvizje nyjet e robotit. Lloje të ndryshme të aktuatorëve zgjidhen bazuar në shpejtësinë, precizitetin dhe kërkesat e ngarkesës së robotit. Llojet e zakonshme të aktuatorëve përfshijnë:

- Aktuatorët elektrik: Të njohur për precizitetin e tyre, ata janë idealë për detyra që kërkojnë pozicionim të saktë, siç është kontrolli i kapëseve ose puna e montimit të imët tek PCB (Printed Circuit Board).
- Aktuatorët hidraulikë: Të fuqishëm dhe të fortë, këta aktuatorë janë faktor kyç në robotët e industrive të rënda që bartin ngarkesa të mëdha.
- Aktuatorët pneumatikë: Të shpejtë dhe të lehtë, këta janë të njohur në detyra të thjeshta automatizimi përsëritëse si paketimi ose renditja.

Për shembull, një saldues robotik në një mjedis robotik industrial mund të përdorë aktuatorë të fuqishëm hidraulikë, ndërsa një robot që monton elektronikë delikate mund të mbështetet në aktuatorë të saktë elektrikë për pozicionim të kujdesshëm. Zgjedhja e aktuatorit të duhur siguron që roboti të arrijë shpejtësinë, fuqinë dhe precizitetin e dëshiruar për detyrën e synuar. Shumica e manipulatorëve robotikë modern përdorin servomotorë elektrikë në secilën nyje si aktuatorë. Këta motorë (shpesh motorë pa furça DC ose sinkronë AC) zgjidhen për raportin e tyre të lartë të çiftit rrotullues ndaj peshës dhe precizitetin. Çdo motor është i çiftuar me një kuti ingranazhesh (transmetuesit harmonikë dhe reduktuesit RV janë të zakonshëm në robotët industrialë) për të amplifikuar çiftin rrotullues dhe për të mbajtur pozicionin nën ngarkesë. Dalja e kutisë së ingranazhit vë në lëvizje nyjën. Pozicioni këndor i secilës nyje matet nga një enkoder (optik ose magnetik), duke ofruar reagime mbi këndin e nyjës ose shpejtësinë e rrotullimit.

Efektori fundor

Efektori fundor është përbërësi i një manipuluesi robotikë që bashkëvepron drejtpërdrejt me mjedisin ose objektet brenda hapësirës së tij të punës. Ai është përgjegjës për kryerjen e detyrave kryesore të robotit, të tilla si kapja, prerja ose saldimi. Dizajni i efektorit fundor është thelbësor, pasi duhet të jetë i pajtueshëm me strukturën e robotit dhe i aftë të trajtojë ngarkesën e kërkuar, saktësinë dhe kërkesat specifike të detyrës. Ekzistojnë disa lloje efektorësh fundorë që përdoren zakonisht në manipulatorët robotik, duke përfshirë kapëse, gota thithëse (suction

cups) dhe pishtarë saldimiti. Çdo lloj efektori fundor ofron aftësi dhe kompromise të ndryshme, duke i bërë ato të përshtatshme për aplikime specifike. Kapëset përdoren për kapjen dhe mbajtjen e objekteve dhe vijnë në dizajne të ndryshme, të tilla si kapëse paralele, këndore dhe adaptive. Kapëset mund të projektohen me forca, materiale dhe madhësi të ndryshme kapjeje për të akomoduar një gamë të gjerë objektsh dhe aplikimesh. Për shembull, një kapëse paralele mund të përdoret në një aplikim marrjeje dhe vendosjeje (pick and place), ndërsa një kapëse adaptive mund të përdoret për trajtimin e objekteve me forma të parregullta. Gotat thithëse përdoren për ngritjen dhe trajtimin e objekteve me sipërfaqe të lëmuara, siç janë qelqi ose fletët metalike. Ato mbështeten në presionin e vakumit për të krijuar një lidhje të sigurt midis efektorit fundor dhe objektit, duke lejuar një trajtim të saktë dhe të butë. Gotat thithëse (suction cups) mund të projektohen me madhësi, forma dhe materiale të ndryshme për t'iu përshtatur llojeve të ndryshme të objekteve dhe kushteve të sipërfaqes. Pishtarët e saldimit janë efektorë fundorë të specializuar që përdoren për kryerjen e detyrave të saldimit, siç është saldimiti me hark ose me pika. Ato janë projektuar për të ofruar kokën e kërkuar, rrymën elektrike dhe lidhjen e qëndrueshme midis materialeve që bashkohen. Pishtarët e saldimit duhet të jenë të pajtueshëm me sistemin e kontrollit të robotit dhe të aftë të përballojnë temperaturat e larta dhe rrymat elektrike të përfshira në procesin e saldimit.



Figura 2: Llojet e efektorëve fundor të manipuluesve robotik [1].

3.1 Sensorët dhe sistemet e reagimit

Sensorët dhe sistemet e reagimit janë komponentë integralë të manipulatorëve robotik, duke rritur aftësinë e tyre për të bashkëvepruar në mënyrë efektive me mjedisin e tyre. Këto sisteme ofrojnë të dhëna në kohë reale, duke u lejuar robotëve të rregullojnë veprimet e tyre në mënyrë dinamike. Sistemet e shikimit, për shembull, u mundësojnë robotëve të identifikojnë, sortojnë, dhe trajtojnë objektet, ndërsa sensorët forcë-moment ndihmojnë në aplikimin e sasisë së saktë të forcës gjatë operacioneve, duke parandaluar dëmtimin si të robotit ashtu edhe të objekteve që ai trajton. Sensorët e afërsisë dhe distancës i lejojnë robotit të lëvizë në mjedisin e tij, duke shmangur pengesat dhe duke siguruar funksionim të sigurt. Këto mekanizma reagimi janë thelbësorë për detyrat që kërkojnë saktësi dhe besueshmëri të lartë, pasi ato lejojnë monitorimin dhe rregullimin e vazhdueshëm të lëvizjeve dhe veprimeve të robotit [3].

3.2 Sistemet e kontrollit për manipuluesit robotik

Sistemet e kontrollit të robotëve luajnë një rol të rëndësishëm në funksionimin e manipulatorëve robotik, pasi ato rregullojnë lëvizjen e robotit dhe sigurojnë që ai të kryejë detyrat e tij me saktësi dhe efikasitet. Ekzistojnë dy lloje kryesore të sistemeve të kontrollit të përdorura në manipulatorët robotik: sistemet e kontrollit me qark të hapur dhe me qark të mbyllur. Secili ofron avantazhe dhe disavantazhe të ndryshme, duke i bërë ato të përshtatshme për aplikime dhe kërkesa specifike të performancës.

3.2.1 Sistemet e kontrollit me qark të hapur

Karakterizohen nga mungesa e reagimeve, që do të thotë se lëvizjet e robotit përcaktohen vetëm nga komandat hyrëse të ofruara nga kontrolluesi. Në një sistem me qark të hapur, kontrolluesi dërgon një seri komandash të aktuatorët e robotit, të cilët më pas ekzekutojnë lëvizjet pa asnjë reagim mbi lëvizjet, pa asnjë reagim mbi pozicionin ose performancën aktuale të robotit. Avantazhi i synuar i sistemeve të kontrollit me qark të hapur është thjeshtësia e tyre, pasi ato nuk kërkojnë mekanizma kompleksë reagimi ose sensorë për të funksionuar. Kjo mund të rezultojë në kosto më të ulëta dhe kompleksitet të reduktuar të sistemit, duke i bërë sistemet me qark të hapur të përshtatshme për aplikime ku saktësia dhe preciziteti nuk janë kritike. Megjithatë, sistemet e kontrollit me qark të hapur kanë disa disavantazhe të rëndësishme. Pa reagime, sistemi nuk mund të kompensojë gabimet ose shqetësimet që mund të ndodhin gjatë

funksionimit, siç janë konsumimi mekanik, ndryshmet në ngarkesë ose forcat e jashtme. Kjo mund të rezultojë në saktësi dhe përsëritshmëri të reduktuar, pasi pozicioni aktual i robotit mund të devijojë nga pozicioni i dëshiruar i specifikuar nga komandat hyrëse. Sistemet e kontrollit me qark të hapur përdoren zakonisht në aplikacione ku niveli i kërkuar i precizitetit dhe saktësisë është relativisht i ulët dhe rreziku i gabimeve ose shqetësimeve është minimal. Shembuj të aplikacioneve të tilla përfshijnë detyra të thjeshta të marrjes dhe vendosjes (pick and place), sisteme transportuesish ose operacione bazë të trajtimit të materialeve.

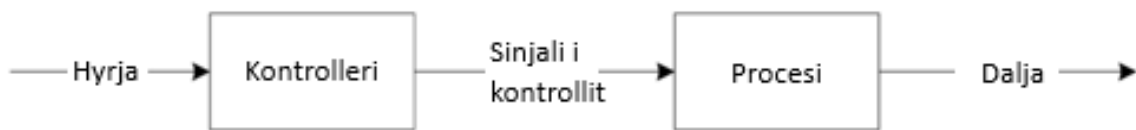


Figura 3: Sistemi i kontrollit me qark të hapur.

3.2.2 Sistemet e kontrollit me qark të mbyllur

Në dallim nga sistemet me qark të hapur, përfshijnë mekanizma reagimi dhe algoritme komplekse për të monitoruar dhe rregulluar lëvizjet e robotit në kohë reale. Në një sistem me qark të mbyllur, kontrolluesi dërgon komanda te aktuatorët e robotit, ndërsa sensorët matin vazhdimisht pozicionin dhe performancën aktuale të robotit. Ky reagim përdoret më pas nga kontrolluesit për të rregulluar komandat, duke siguruar që lëvizjet e robotit përputhen ngushtë me pozicionin dhe trajektoren e dëshiruar. Avantazhi kryesor i sistemeve të kontrollit me qark të mbyllur është aftësia e tyre për të kompensuar gabimet dhe shqetësimet që mund të ndodhin gjatë funksionimit. Duke monitoruar vazhdimisht pozicionin dhe performancën e robotit, sistemi mund të bëjë rregullime në komanda, duke rezultuar në saktësi dhe përsëritje të përmirësuar. Kjo i bën sistemet me qark të mbyllur të përshtatshme për aplikime ku preciziteti dhe saktësia e lartë janë kritike. Megjithatë, sistemet e kontrollit me qark të mbyllur kanë disa disavantazhe. Ato zakonisht kërkojnë harduerë dhe softuer më kompleks, pasi mbështeten në mekanizma reagimi dhe sensorë për të funksionuar. Kjo mund të rezultojë në rritje të kostove dhe kompleksitetit të sistemit, duke i bërë sistemet me qark të mbyllur më pak të përshtatshme për aplikime ku thjeshtësia dhe efektiviteti i kostos janë përparësi. Sistemet e kontrollit me qark të mbyllur përdoren zakonisht në aplikime ku kërkohet precizitet dhe saktësi e lartë, siç janë detyrat e montimit, inspektimit ose përpunimit mekanik. Për shembull, një sistem me qark të

mbyllur mund të përdoret në aplikimet e saldimit robotik, ku roboti duhet të mbajë një pozicion dhe trajektore të saktë për të siguruar një saldimit të fortë dhe të qëndrueshëm. Në këtë rast, sistemi me qark të mbyllur do të monitoronte vazhdimisht pozicionin e robotit dhe do të bënte rregullime sipas nevojës për të ruajtur rrugën e dëshiruar të saldimit.

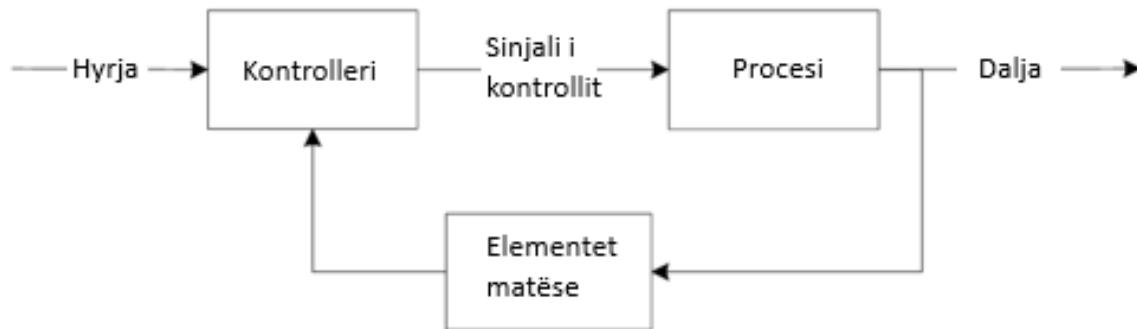


Figura 4: Sistemi i kontrollit me qark të mbyllur.

3.3 Teknikat kryesore të kontrollit për manipuluesit robotik

Sistemet e kontrollit robotik mbështeten në teknikat e kontrollit të vendosura për të siguruar lëvizje të qetë dhe të saktë. Metodatat më të zakonshme përfshijnë:

3.3.1 Kontrolli PID (Kontrolli Proporcional-Integral-Derivativ)

Është një nga metodatat më të përdorura në manipuluesit robotikë. Ai funksionon duke llogaritur tre rregullime kryesore për ta mbajtur robotin në rrugën e dëshiruar:

- Kontrolli Proporcional (P): Korrigojnë gabimet bazuar në ndryshimin midis pozicionit aktual të robotit dhe pozicionit të tij të synuar.
- Kontrolli Integral (I): Korrigojnë gabimet e kaluara që grumbullohen me kalimin e kohës.
- Kontrolli Derivativ (D): Parashikojnë gabimet e ardhshme bazuar në tendencat e lëvizjes së robotit.

Duke kombinuar këto tre rregullime, kontrolli PID minimizon tejkalimin, përmirëson stabilitetin dhe siguron pozicionim të saktë.

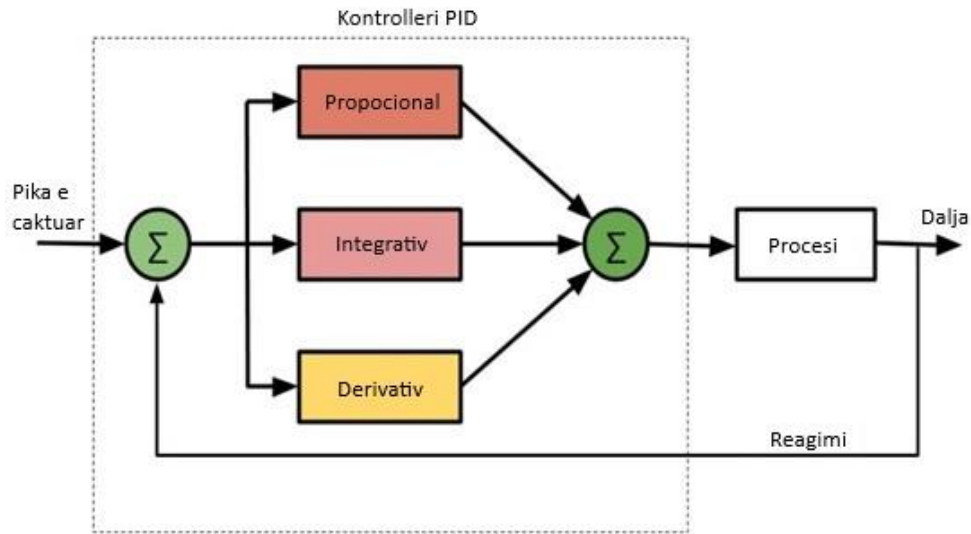


Figura 5: Kontrolli përmes PID.

3.3.2 Planifikimi i Trajektores

Përqendrohet në projektimin e rrugës së lëvizjes së robotit – duke siguruar tranzicion të qetë midis pikave. Në vend që të lëvizë drejtpërdrejt nga një pozicion në tjetrin (gjë që mund të rezultojë në lëvizje të lëkundur), planifikimi i trajektores llogarit kurbat e optimizuara të lëvizjes që parandalojnë ndryshimet e papritura të shpejtësisë ose këndet e mprehta. Kjo është veçanërisht e rëndësishme për robotët që trajtojnë materiale të brishta ose që veprojnë me shpejtësi të lartë.

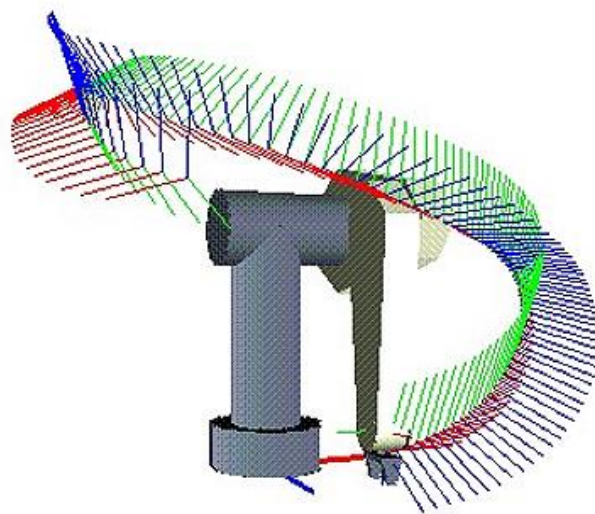


Figura 6: Planifikimi i trajektores.

3.3.3 Kontrolli i Impedancës (Kontrolli i Forcës)

Në detyrat që kërkojnë ndërveprim fizik me objekte, kontrolli i impedancës ndihmon robotin të përshtatet me forcat e papritura. Në vend që të ndjekë shtigje fikse të lëvizjes, kontrolli i impedancës rregullon lëvizjen e robotit bazuar në rezistencë.

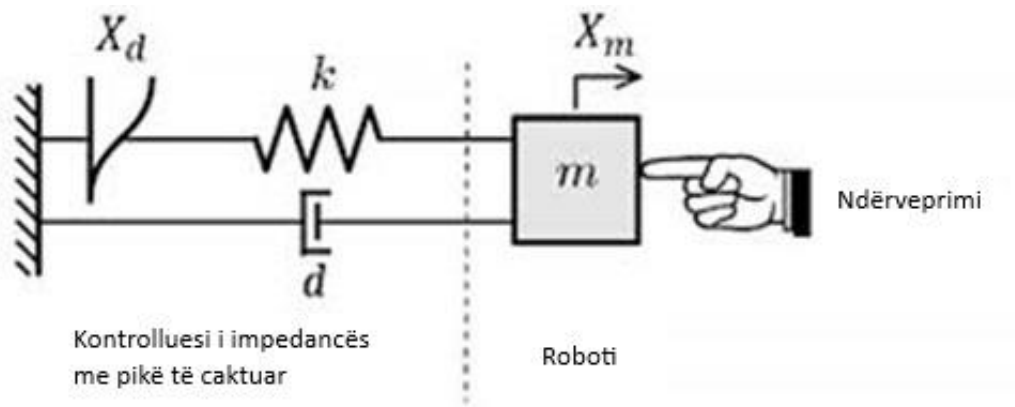


Figura 7: Kontrolli me anë të impedancës.

3.3.4 Kontrolli Adaptiv

Sistemet e kontrollit adaptiv modifikojnë sjelljen e robotit bazuar në kushtet në ndryshim. Për shembull, nëse ngarkesa e një roboti rritet papritur (si një objekt më i rëndë në një transportues), kontrolli adaptiv rillogarit fuqinë e motorit për të ruajtur lëvizjen e saktë pa ngadalësuar.

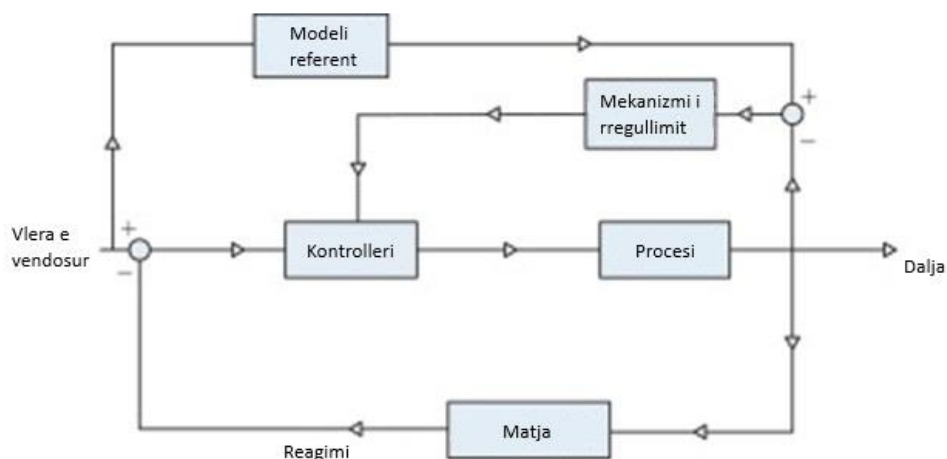


Figura 8: Kontrolli adaptiv.

3.4 Projektimi i një sistemi efektiv kontrolli

Për të projektuar një manipulator robotik si dhe për të bërë përzgjedhjen e duhur të sistemit të kontrollit duhet pasur në konsideratë:

1. Nevojat për precizitet: Detyrat si montimi i mikroçipave ose prerja me laser kërkojnë sisteme me qark të mbyllur me kontroll PID për saktësi maksimale.
2. Kërkesat për shpejtësi: Për renditje ose paketim me shpejtësi të lartë, kontrolli me qark të hapur mund të jetë më i shpejtë dhe i mjaftueshëm.
3. Faktorët mjedisorë: Në mjedise të paparashikueshme (p.sh., puna me njerëz ose materiale të brishta), kontrolli adaptiv dhe sensorët e forcës bëhen thelbësore.
4. Variacionet e ngarkesës: Robotët që trajtojnë objekte me pesha të ndryshme duhet të përfshijnë kontroll adaptiv ose kontroll të impedancës për lëvizje të qetë [4].

3.5 Llojet e manipuluesve robotik

Manipuluesit robotik zakonisht kanë nga tre deri në gjashtë shkallë lirie, të ofruara nga një sërë nyjesh të cilat lidhin lidhjet e ndryshme të pajisjes. Lidhjet e një manipuluesi zakonisht janë ose nyje rrotulluese ose nyje prizmatike. Lloji i krahut përcakton numrin e shkallëve të lirisë, si dhe hapësirën e tij të punës (sa larg mund të arrijë). Manipulatorët robotikë mund të klasifikohen në bazë të strukturës dhe aftësive të tyre të lëvizjes. Çdo lloj ka avantazhet dhe disavantazhet e tij, duke i bërë ato të përshtatshme për aplikime specifike.

3.5.1 Manipuluesit robotik kartezi

Manipuluesit robotikë kartezi kanë një dizajn linear dhe përbëhen nga tre boshte ortogonale (x , y dhe z). Këta robotë lëvizin në vija të drejta përgjatë secilit aks, duke lejuar pozicionimin dhe lëvizjen e saktë brenda një hapësire të caktuar pune. Struktura e robotëve kartezi zakonisht përfshin aktuatorët linearë ose rrëshqitës, të cilët i mundësojnë robotit të lëvizë përgjatë secilit aks në mënyrë të pavarur. Një nga avantazhet kryesore të manipuluesve robotik kartezi është saktësia dhe përsëritshmëria e tyre e lartë. Për shkak të dizajnit të tyre linear, këta robotë mund të arrijnë saktësi pozicionimi deri në 0.001mm. Për më tepër, robotët kartezi janë të njohur për shkallëzueshmërinë e tyre, pasi ato mund të personalizohen lehtësisht për të akomoduar madhësi të ndryshme të hapësirës së punës dhe kapaciteteve të ngarkesës. Sidoqoftë, manipulatorët robotik kartezi kanë disa kufizime. Lëvizja e tyre është

e kufizuar në shtigje lineare, të cilat mund të mos jenë të përshtatshme për aplikime që kërkojnë trajektore komplekse ose të lakuara. Për më tepër, madhësia dhe pesha e robotëve karteziianë mund të jenë thelbësore, veçanërisht për aplikime në shkallë të gjerë, të cilat mund të rezultojnë në rritjen e konsumit të energjisë dhe reduktimin e efikasitetit të përgjithshëm. Manipulatorët robotik karteziianë përdoren zakonisht në industri të tilla si prodhimi dhe automatizimi. Aftësitë e tyre të sakta të pozicionimit i bëjnë ato ideale për detyra të tilla si operacionet e marrjes dhe vendosjes (pick and place), montimi dhe inspektimi.

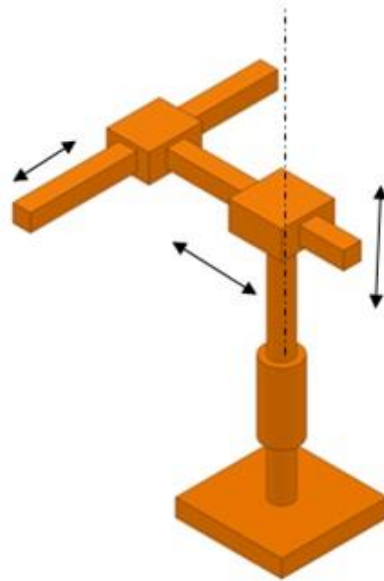


Figura 9: Manipuluesi robotik karteziianë.

3.5.2 Manipuluesit robotik cilindrik

Manipuluesit robotik cilindrik kanë hapësirë pune cilindrike dhe përbëhet nga një kombinim i lëvizjeve lineare dhe rrotulluese. Këta robotë zakonisht kanë një bosht linear vertikal (bosht z), një bosht linear radial (bosht r) dhe një bosht rrotulluese (bosht $teta$) rreth boshtit vertikal. Struktura e robotëve cilindrik shpesh përfshin një kolonë vertikale të montuar në një nyje rrotulluese në bazë, me një krah që mund të shtrihet në mënyrë radiale dhe të rrotullohet rreth boshtit vertikal. Një nga avantazhet kryesore të manipuluesve robotik cilindrik është aftësia e tyre për të kapërcyer pengesat brenda hapësirës së tyre cilindrike të punës. Kjo i bën ata të përshtatshme për aplikime ku qasja në objektin e synuar mund të pengohet. Për më tepër, robotët cilindrik mund të ofrojnë një zgjidhje kompakte, pasi boshti i tij vertikal dhe dizajni i krahut radial i lejojnë ata të zënë më pak hapësirën e bazës në krahasim me llojet e tjera të

robotëve. Sidoqoftë, manipuluesit robotik cilindrik kanë gjithashtu disa kufizime. Korniza e tyre e punës është e kufizuar në një formë cilindrike, e cila mund të mos jetë e përshtatshme për aplikime që kërkojnë një hapësirë pune më të madhe me forma të ndryshme. Për më tepër, saktësia dhe përshtatshmëria e robotëve cilindrik mund të jetë më e ulët se ajo e robotëve karteziqan, pasi kombinimi i lëvizjeve lineare dhe rrotulluese mund të sjellë një burim shtesë gabimi.



Figura 10: Manipuluesi robotik cilindrik.

3.5.3 Manipuluesit robotik sferik

Manipuluesit robotik sferik, të njohur gjithashtu si robot polarë, kanë një hapësirë pune sferike dhe përbëhen nga një kombinim i lëvizjeve lineare dhe rrotulluese. Këta manipulues robotik, njësoj si krahu i njeriut, shfaqin një gamë të jashtëzakonshme lëvizjeje dhe fleksibiliteti, duke i lejuar ata të lëvizin dhe kryejnë në mënyrë efikase detyra brenda një hapësire pune tre dimensionale. Këta robotë zakonisht kanë një bosht linar radial (bosht ρ), një bosht rrotullues polar (bosht θ) dhe një bosht rrotullues azimutal (bosht φ). Robotët sferik vijnë me dy nje rrotulluese dhe një prizmatike. Struktura e robotëve sferik shpesh përfshin një manipulues të montuar në bazë që mund të shtrihet në mënyrë radiale dhe të rrotullohet rreth dy boshteve pingul. Një nga avantazhet kryesore të manipuluesve robotik sferik është aftësia e tyre për të mbuluar një hapësirë të madhe pune me një gjurmë (footprint) relativisht të vogël. Kombinimi unik i lëvizjeve lineare dhe rrotulluese i lejon ata të mbulojnë një gamë të gjerë pozicionesh brenda hapësirës së tyre sferike të punës. Për më tepër, robotët sferik mund të ofrojnë një shkallë të lartë shkathtësie, duke i bërë ata të përshtatshëm për aplikime që kërkojnë lëvizje

komplekse dhe të sakta. Sidoqoftë, robotët manipulues sferik kanë gjithashtu disa kufizime. Saktësia dhe përshtatshmëria e tyre mund të jetë më e ulët se ajo e robotëve kartezian, pasi kombinimi i lëvizjeve lineare dhe rrotulluese mund të sjellë burime shtesë gabimi. Manipuluesit robotik sferik përdoren zakonisht në industri të tilla si trajtimi i materialeve dhe saldimit. Aftësia e tyre për të mbuluar një hapësirë të madhe pune dhe për të kryer lëvizje komplekse i bën ata të përshtatshëm për detyra të tilla si zhvendosja e objekteve midis vendndodhjeve të ndryshme ose kryerja e operacioneve të ndërlikuara të saldimit.

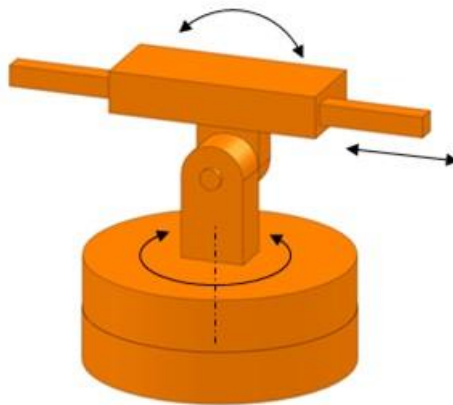


Figura 11: Manipuluesi robotik sferik.

3.5.4 Manipuluesit robotik të artikulluar

Manipuluesit robotik të artikulluar, kanë një strukturë komplekse të përbërë nga nyje dhe lidhje të shumta që lejojnë një gamë të gjerë lëvizjesh. Këta robotë zakonisht kanë një seri nyjesh rrotulluese të lidhura me lidhje të ngurtë, të cilat u mundësojnë atyre të kryejnë lëvizje në shkallë të shumta lirie (DoF). Struktura e robotëve të artikulluar shpesh përfshin një manipulues të montuar në një bazë me segmente të shumta, secili i lidhur nga një nyje që lejon lëvizjen rrotulluese. Një nga avantazhet kryesore të manipuluesit robotik të artikulluar është shkalla e lartë e fleksibilitetit dhe shkathhtësisë së tyre. Shkallët e tyre të shumta të lirisë u mundësojnë atyre të kryejnë detyra komplekse dhe të arrijnë një gamë të gjerë pozicionesh brenda hapësirës së tyre të punës. Për më tepër, robotët e artikulluar mund të dizajnohen me numër të ndryshëm nyjesh dhe lidhjesh, duke lejuar personalizimin për t'iu përshtatur kërkesave specifike të aplikimit. Sidoqoftë, kompleksiteti i tyre ka disa kufizime, struktura e tyre komplekse mund t'i bëjë ata më sfidues për programim dhe kontroll në krahasim me llojet e tjera të robotëve. Për më tepër, saktësia dhe përsëritshmëria e robotëve të artikulluar mund të jetë më e ulët se ajo e

robotëve karteziar, pasi nyjet dhe lidhjet e shumta mund të sjellin burim shtesë gabimi. Manipuluesit robotik të artikuluar përdoren zakonisht në industri të tilla si automobila dhe hapësira ajrore. Fleksibiliteti dhe shkathtësia e tyre i bëjnë ata të përshtatshëm për detyra që kërkojnë lëvizje komplekse dhe pozicionim të saktë, si montimi ljerja dhe inspektimi [2].

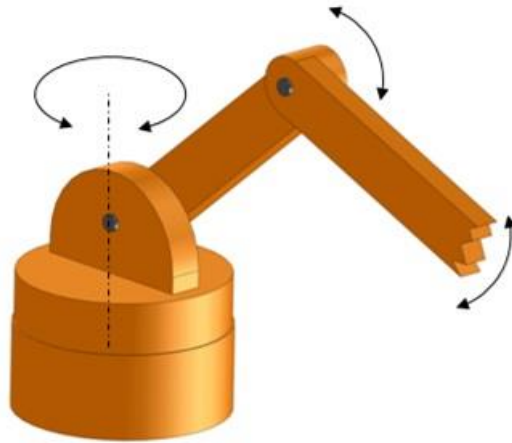


Figura 12: Manipuluesi robotik i artikuluar.

4 Komponentët mekanik dhe elektrik

Manipuluesi robotik është dizajnuar duke përdorur programin SolidWorks. Përmes tij është krijuar një model preciz i manipuluesit robotik, duke përfshirë kapësen (gripperin) dhe komponentët e tjerë mekanik. Dizajni është optimizuar për të siguruar funksionalitet, stabilitet dhe efikasitet gjatë manipulimit të objekteve. Gjithashtu, janë realizuar simulime për të analizuar lëvizjen dhe për të vlerësuar sjelljen strukturore të manipuluesit robotik nën ngarkesa të ndryshme, duke garantuar që të përmbushen kërkesat teknike dhe funksionale.

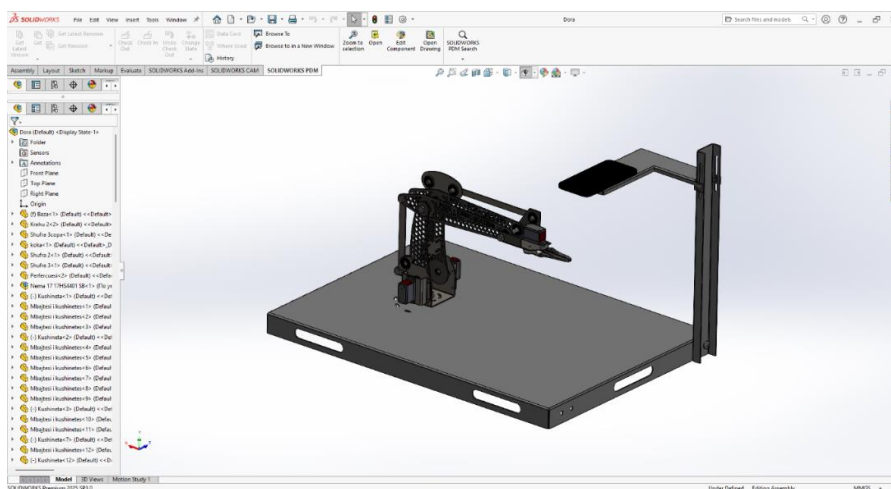


Figura 13: Projektimi i manipuluesit robotik në SolidWorks [2].

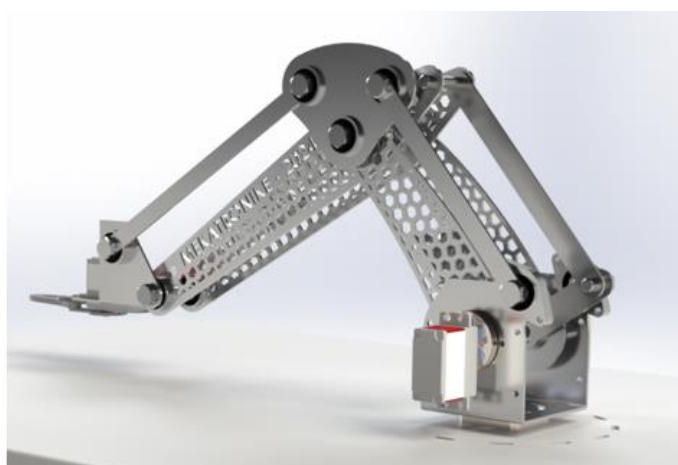


Figura 14: Manipuluesi robotik i projektuar [3].

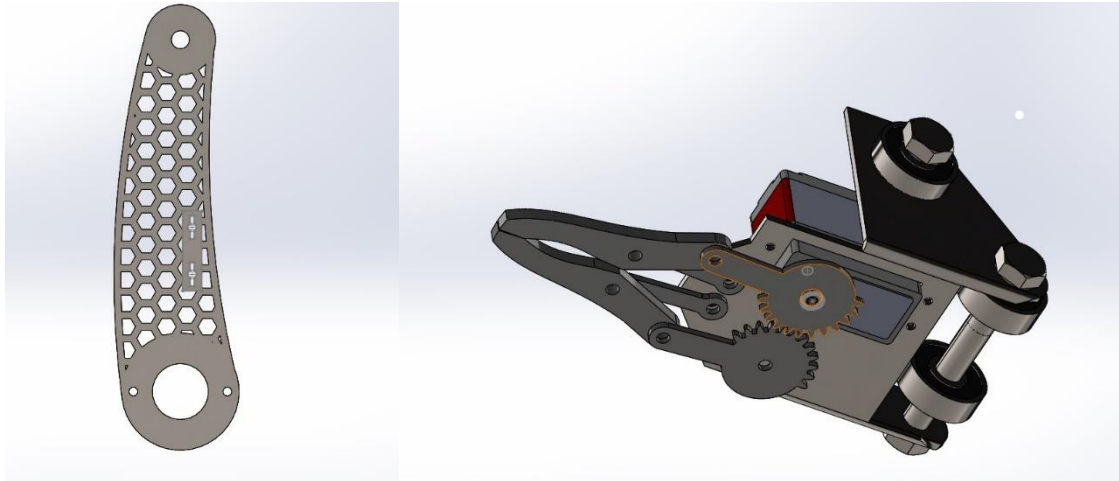


Figura 15: Lidhjet dhe efektori fundor të projektuar në SolidWorks.

Të gjitha pjesët përbërëse të manipuluesit robotik pas dizajnit janë bërë gati për përpunim me laser dhe lakim. Prerja është bërë me laser 3kW, ndërsa lakim është bërë me pressbrake. Disa nga pjesët tjera të manipuluesit janë të salduara me laser. Ky lloj saldimit është bërë kryesisht në foletë e kushinetave.



Figura 16: Përgatitja dhe prerja me laser e pjesëve përbërëse të manipuluesit robotik.

Sistemi elektrik i manipuluesit robotik është projektuar për të siguruar kontroll të saktë dhe të besueshëm të të gjitha lëvizjeve. Komponentët kryesore elektrike të përdorura janë:

Arduino UNO, shërben si njësi kryesore e kontrollit të sistemit. Ky mikrokontroller merr dhe përpunon komandat e lëvizjes dhe dërgon sinjalet e duhura drejt të gjithë aktuatorëve.

Stepper motori Nema 17 është përdorur për bazën rrotulluese të manipuluesit. Ky motor ofron pozicionim preciz dhe moment të lartë rrotullues, i cili është i nevojshëm për bartjen e ngarkesës së të gjithë strukturës. Mundëson lëvizje të kontrolluara në hapa të vegjël, duke garantuar saktësi të lartë pozicionimi.

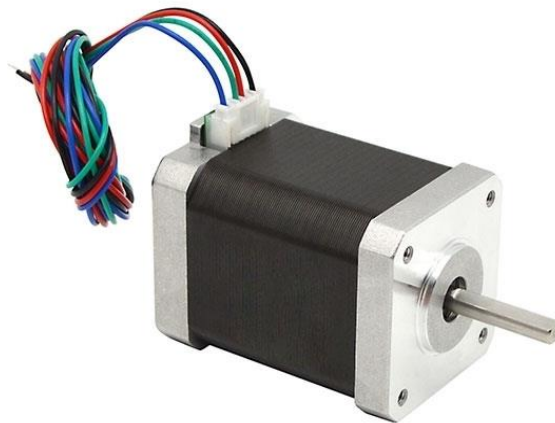


Figura 17: Stepper motori Nema17 [4].

Janë përdorur tre **servo motorë KST BLS825**, një servo motor është vendosur në krah, duke kontrolluar lëvizjen e ngritjes dhe uljes së krahut të parë. Një servo motor është vendosur në bërryl, duke mundësuar lakimin e krahut të dytë. Një servo motor është rezervuar për gripperin (kapësen), duke kontrolluar hapjen dhe mbylljen e gishtave të kapësit për të kapur dhe lëshuar objektet. Servo motorët kontrollohen direkt nga pinat PWM të Arduino UNO, duke siguruar pozicionim të saktë këndor në çdo aks.



Figura 18: Servo motori KST BLS825 [5].

Ngasësi TB6600 është përdorur për kontrollin e motor hap-pas-hapi (stepper motor). Ky ngasës mbështet rrymë deri në 4A dhe tensione pune nga 9V deri në 42V, duke ofruar mundësinë e rregullimit të mikrohapave për lëvizje edhe më të buta dhe precize. TB6600 komunikon me

Arduino UNO përmes sinjaleve STEP dhe DIR, duke lejuar kontroll të plotë të shpejtësisë dhe drejtimit të rrotullimit.

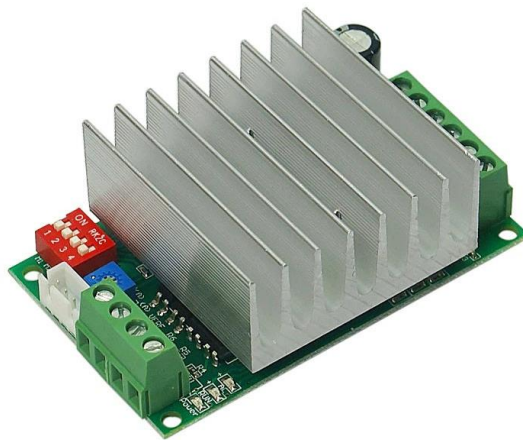


Figura 19: Ngasësi TB6600 [6].

Konvertues DC-DC me ulje tensioni (DC-DC Step Down Converter) është përdorur për të rregulluar dhe stabilizuar tensionin e furnizimit. Meqë servo motorët dhe motori stepper kërkojnë tensione dhe rryma të ndryshme nga ato që ofron drejtpërdrejt Arduino UNO, Konvertues me ulje tensioni ul tensionin e furnizimit kryesor në nivelin e kërkuar nga secili komponent, duke siguruar funksionim të sigurt dhe eficientë të të gjithë sistemit elektrik.



Figura 20: Konvertues DC-DC me ulje tensioni [7].

Pasi të gjitha pjesët kanë përfunduar procesin e përpunimit, dhe janë zgjedhur komponentët e duhur elektronik, është bërë montimi i tyre duke u bazuar në vizatimet teknike dhe skemën e lidhjes elektrike.

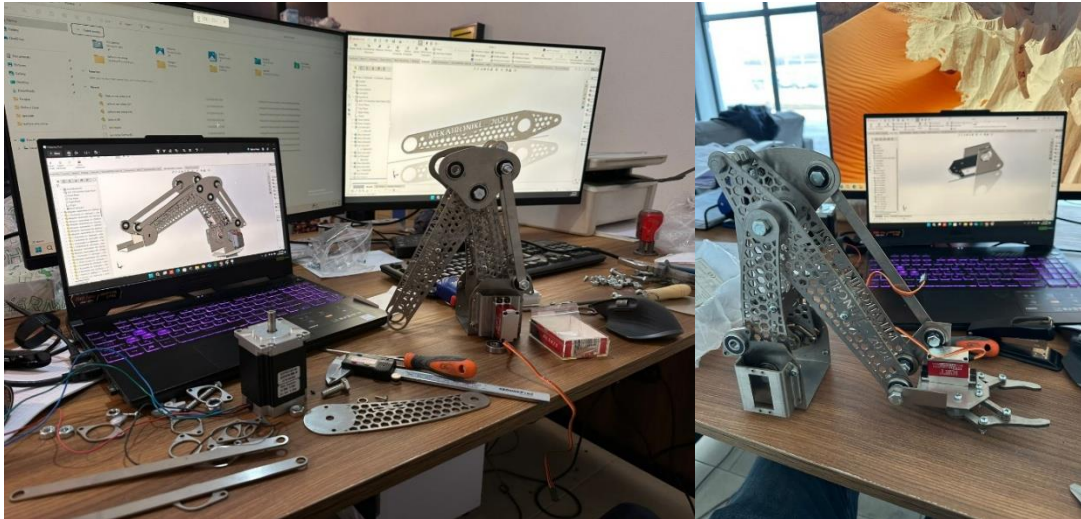


Figura 21: Montimi i manipuluesit robotik.

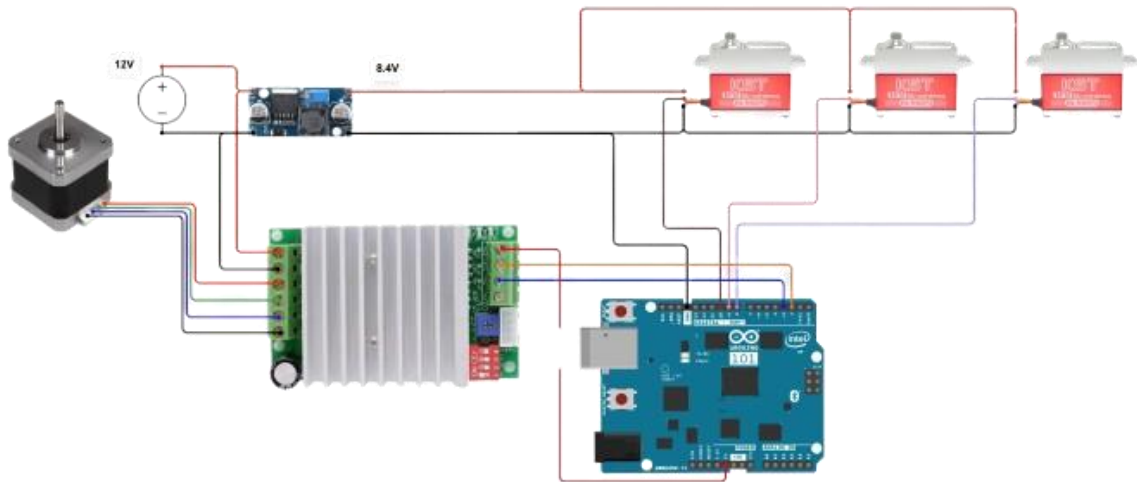


Figura 22: Skema e lidhjeve elektrike.

5 Kinematika direkte, inverse dhe lëvizja e robotit

Kinematika është një nga aspektet më të rëndësishme në projektimin dhe kontrollin e manipuluesve robotik. Ajo merret me studimin e lëvizjes së robotit pa marrë parasysh forcat që e shkaktojnë atë. Me fjalë të tjera, kinematika përshkruan marrëdhënien ndërmjet këndeve të nyjeve rrotulluese, në raport me pozicionin dhe orientimin e efektorit fundor (end-effector) në hapësirë. Ekzistojnë dy lloje kryesore të analizës kinematike: kinematika direkte dhe kinematika inverse [1].

5.1 Kinematika direkte

Kinematika direkte (ang. Forward kinematics) ka për qëllim të përcaktojë pozicionin dhe orientimin e efektorit fundor në hapësirë, duke pasur si të dhëna hyrëse këndet e nyjeve. Në një robot me n nyje, kinematika direkte merret me gjetjen e lidhjes matematikore ndërmjet vlerave të këndeve të nyjeve $\theta_1, \theta_2, \dots, \theta_n$ dhe pozicionit (x, y, z) të efektorit fundor.

5.1.1 Metoda e Denavit Hartenberg-ut

Metoda e Denavit Hartenberg-ut është mënyra më e përdorur për të përshkruar lëvizjet e robotit përmes matricave transformuese homogjene. Paraqet një mënyrë sistematike për aprovimin e sistemeve koordinatave për nyjet e manipuluesve, ashtu që të përshkruhet pozita dhe orientimi i secilës nyje. Ky përcaktim i sistemeve koordinatave bëhet duke u bazuar në disa rregulla elementare. Këto rregulla për përcaktimin e sistemeve koordinatave janë:

- Boshti z_i përcaktohet përgjatë boshtit të lëvizjes së nyjës i ,
- Boshti x_i përcaktohet normal ndaj boshtit z_{i-1} dhe z_i ,
- Boshti y_i përcaktohet sipas rregullës së dorës së djathtë për të plotësuar sistemin koordinativ (x, y, z) .

Pasi të jenë aprovuar sistemet koordinatave sipas notacionit të Denavit Hartenberg-ut (D-H), hapi tjetër është përcaktimi i parametrave D-H për secilën nyje/segment të manipulatorit. Kjo ngase qëllimi i kinematikës direkte sipas notacionit të D-H është formulimi i tabelës së manipuluesit robotik, prej të cilës më pas mund të gjenerohen matricat transformuese homogjene që përshkruajnë raportin në mes të nyjeve sekuenciale të manipulatorit robotik [2].

Çdo lidhje përshkruhet nga katër parametra:

Tabela 1: Parametrat e Denavit Hartenberg-ut dhe përshkrimi i tyre.

Simboli	Përshkrimi
θ_i	Këndi ndërmjet x_{i-1} dhe x_i për rrotullim ndaj z_{i-1}
d_i	Distanca nga origjina e sistemit $i - 1$ deri tek x_i përgjatë aksit z_{i-1}
a_i	Distanca në mes pikëpresjes së z_{i-1} dhe z_i përgjatë boshtit x_i
α_i	Këndi ndërmjet z_{i-1} dhe z_i për rrotullim ndaj x_i

5.1.2 Matrica transformuese homogjene

Për të përshkruar rrotullimin si dhe zhvendosjen në një operacion të vetëm. Në robotikë përdoren koordinatat homogjene dhe matrica transformuese 4×4 .

$$T = \begin{bmatrix} R_{3 \times 3} & p_{3 \times 1} \\ f_{1 \times 3} & 1 \times 1 \end{bmatrix} \quad (1)$$

$R_{3 \times 3}$ – matrica rrotulluese

$f_{1 \times 3}$ – transformimi i perspektives

$p_{3 \times 1}$ – vektori pozicional që tregon pozitën e pikës në raport me një sistem koordinatave referent

1×1 – shkallëzimi

Një matricë transformuese homogjene T_i^{i-1} zbaton transformimin nga sistemi koordinativ $i - 1$ në sistemin i dhe përcaktohet nga:

$$T_{i-1}^i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & \alpha_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & \alpha_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Kjo matricë fitohet duke përdorur katër transformime bazë, të aplikuara në këtë rend:

1. Rrotullimi rreth boshtit z_{i-1} për këndin θ_i :

$$Rot_{z,\theta_i} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

2. Zhvendosja përgjatë boshtit z_{i-1} për distancën d_i :

$$Trans_{z,d_i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

3. Zhvendosja përgjatë boshtit x_i për distancën a_i :

$$Trans_{x,a_i} = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

4. Rrotullimi rreth boshtit x_i për këndin α_i :

$$Rot_{x,\alpha_i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_i & -\sin\theta_i & 0 \\ 0 & \sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

Kështu, matrica T_{i-1}^i fitohet duke shumëzuar këto katër transformime:

$$T_{i-1}^i = Rot_{z,\theta_i} \cdot Trans_{z,d_i} \cdot Trans_{x,a_i} \cdot Rot_{x,\alpha_i} \quad (7)$$

Transformimi total nga baza në efektorin fundor arrihet duke shumëzuar të gjitha matricat përkatëse:

$$T_0^n = T_0^1 \cdot T_1^2 \cdot T_2^3 \cdot \dots \cdot T_{n-1}^n \quad (8)$$

5.1.3 Llogaritja e kinematikës direkte për manipuluesin robotik me 3-DOF

Në figurën më poshtë është paraqitur manipuluesi robotik me 3 nyje rrotulluese. Secila nyje është e modeluar sipas metodës D-H dhe sistemet koordinatave janë vendosur në përputhje me rregullat përkatëse.

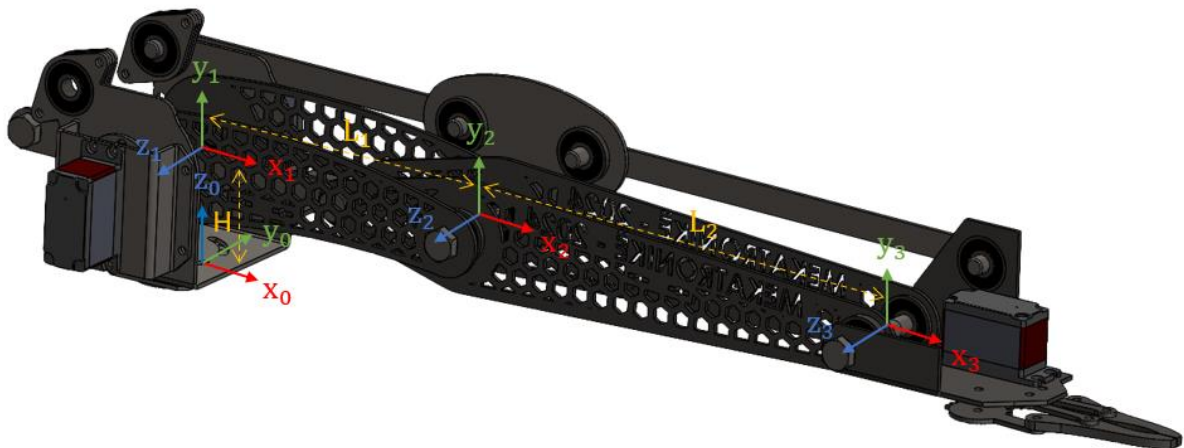


Figura 23: Aprovimi i sistemeve koordinatave sipas D-H për manipuluesin robotik.

Më pas është ndërtuar tabela me parametrat e Denavit Hartenberg-ut si më poshtë:

Tabela 2: Parametrat e D-H për manipuluesin robotik.

Nyja	θ_i	d_i	a_i	α_i
1	θ_1	H	0	α_1
2	θ_2	0	L_1	0
3	θ_3	0	L_2	0

Pas vendosjes së sistemeve koordinatave dhe ndarjes së parametrave D-H sipas tabelës, për këtë manipulator ndërtohen matricat transformuese si më poshtë:

$$T_0^1 = \begin{bmatrix} \cos\theta_1 & 0 & \sin\theta_1 & 0 \\ \sin\theta_1 & 0 & -\cos\theta_1 & 0 \\ 0 & 1 & 0 & H \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

$$T_1^2 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & L_1\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & L_1\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

$$T_2^3 = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & L_2\cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & L_2\sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

Me pas llogaritet matrica e përgjithshme:

$$T_0^3 = T_0^1 \cdot T_1^2 \cdot T_2^3 \quad (12)$$

$$T_0^2 = T_0^1 \cdot T_1^2 = \begin{bmatrix} \cos\theta_1 \cos\theta_2 & -\cos\theta_1 \sin\theta_2 & \sin\theta_1 & L_1 \cos\theta_2 \cos\theta_1 \\ \sin\theta_1 \cos\theta_2 & -\sin\theta_1 \sin\theta_2 & -\cos\theta_1 & L_1 \sin\theta_2 \sin\theta_1 \\ \sin\theta_2 & \cos\theta_2 & 0 & H + L_1 \sin\theta_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13)$$

$$T_0^3 = T_0^2 \cdot T_2^3 = \begin{bmatrix} \cos\theta_1 \cos(\theta_2 + \theta_3) & -\cos\theta_1 \sin(\theta_2 + \theta_3) & \sin\theta_1 & (L_1 \cos\theta_2 + L_2 \cos(\theta_2 + \theta_3)) \cos\theta_1 \\ \sin\theta_1 \cos(\theta_2 + \theta_3) & -\sin\theta_1 \sin(\theta_2 + \theta_3) & -\cos\theta_1 & (L_1 \cos\theta_2 + L_2 \cos(\theta_2 + \theta_3)) \sin\theta_1 \\ \sin(\theta_2 + \theta_3) & \cos(\theta_2 + \theta_3) & 0 & H + L_1 \sin\theta_2 + L_2 \sin(\theta_2 + \theta_3) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (14)$$

Kjo matricë përfaqëson pozicionin dhe orientimin e efektorit fundor në raport me bazën. Vlerat e fundit të kolonës së katërt të matricës T_0^3 tregojnë koordinatat (x, y, z) të efektorit fundor në hapësirë, në funksion të këndeve të nyjeve $\theta_1, \theta_2, \theta_3$ dhe gjatësive të segmenteve L_1, L_2 .

$$x = (L_1 \cos\theta_2 + L_2 \cos(\theta_2 + \theta_3)) \cos\theta_1 \quad (15)$$

$$y = (L_1 \cos\theta_2 + L_2 \cos(\theta_2 + \theta_3)) \sin\theta_1 \quad (16)$$

$$z = H + L_1 \sin\theta_2 + L_2 \sin(\theta_2 + \theta_3) \quad (17)$$

Për vlerat $L_1 = 15$, $L_2 = 20$, $H = 0$ dhe këndet $\theta_1 = 45$, $\theta_2 = 50$, $\theta_3 = -25$ kemi:

$$x = (15 \cdot 0.643 + 20 \cdot 0.906) \cdot 0.707 = 19.629$$

$$y = (15 \cdot 0.643 + 20 \cdot 0.906) \cdot 0.707 = 19.629$$

$$z = 0 + 15 \cdot 0.766 + 20 \cdot 0.423 = 19.95$$

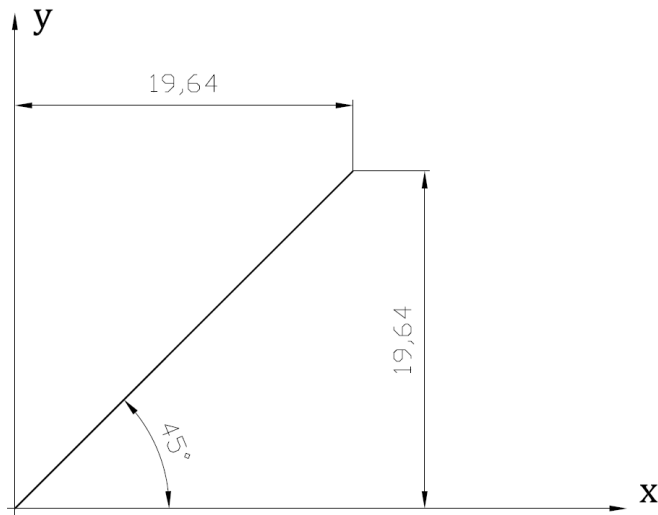


Figura 24: Pozicioni i efektorit fundor përgjatë x dhe y .

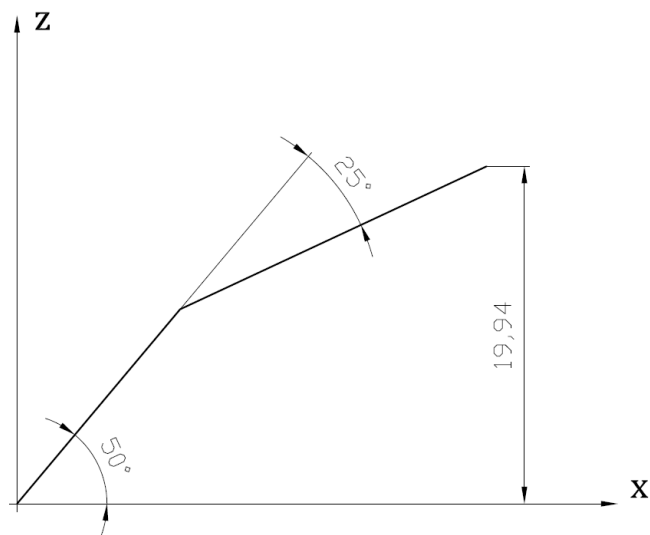


Figura 25: Pozicioni i efektorit fundor përgjatë x dhe z .

5.2 Kinematika inverse

Kinematika inverse (ang. Inverse kinematics) ka për qëllim të përcaktojë këndet e nyjeve të robotit, në mënyrë që efektori fundor të arrijë një pozicion të caktuar në hapësirë. Për dallim nga kinematika direkte, në kinematikën inverse pozicioni dhe orientimi i efektorit fundor janë të njohura, ndërsa duhen gjetur vlerat e këndeve të nyjeve $\theta_1, \theta_2, \dots, \theta_n$ që e bëjnë robotin të arrijë në atë pikë [1].

5.2.1 Llogaritja e kinematikës inverse për manipuluesin robotik me 3-DOF

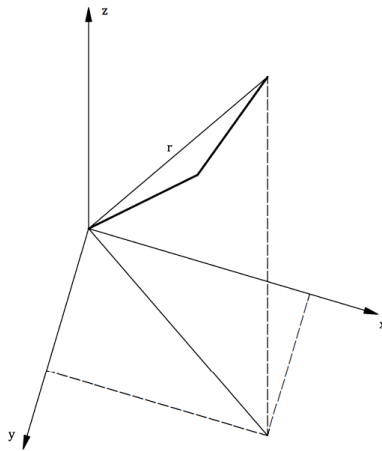


Figura 26: Distanca nga baza deri tek efektori fundor.

Distanca totale nga baza deri tek efektori fundor:

$$r = \sqrt{x^2 + y^2 + z^2} \quad (19)$$

Nëse $|L_1 - L_2| \leq r \leq L_1 + L_2$ pika është e arritshme.

Nëse manipuluesi robotik duhet të arrijë në pikën me koordinata (x, y, z) , atëherë ai fillimisht duhet të rrotullohet në drejtimin e duhur në rrafshin horizontal për këndin θ_1 :

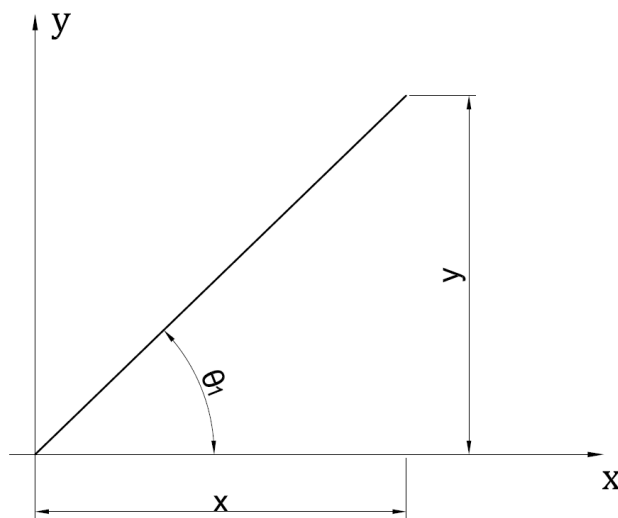


Figura 27: Këndi në rrafshin $x - y$.

$$\theta_1 = \tan^{-1}\left(\frac{y}{x}\right) \quad (20)$$

Pastaj duhet të bëhet rrotullimi i L_1 për këndin θ_2 dhe L_2 për këndin θ_3 në rrafshin vertikal:

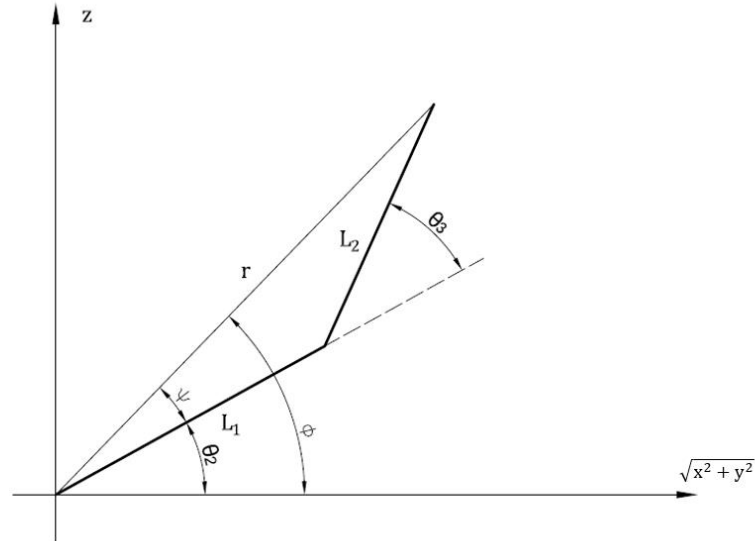


Figura 28: Këndet në rrafshin $\sqrt{x^2 + y^2} - z$.

Nga teorema e kosinuset:

$$r^2 = L_1^2 + L_2^2 - 2L_1L_2\cos(\pi - \theta_3) \quad (21)$$

Duke përdorur identitetin $\cos(\pi - \theta_3) = -\cos(\theta_3)$:

$$r^2 = L_1^2 + L_2^2 + 2L_1L_2\cos(\theta_3) \quad (23)$$

Nga këtu fitohet:

$$\theta_3 = -\cos^{-1}\left(\frac{r^2 - L_1^2 - L_2^2}{2L_1L_2}\right) \quad (24)$$

Shenja negativ tregon konfigurimin “elbow-down”.

$$\Phi = \tan^{-1}\left(\frac{z}{\sqrt{x^2 + y^2}}\right) \quad (25)$$

$$\Psi = \tan^{-1}\left(\frac{L_2\sin(\theta_3)}{L_1 + L_2\cos(\theta_3)}\right) \quad (26)$$

$$\theta_2 = \Phi - \Psi \quad (27)$$

Per vlerat $x = 19.64$, $y = 19.64$ dhe $z = 19.94$ kemi:

$$r = \sqrt{19.64^2 + 19.64^2 + 19.94^2} = \sqrt{1169.0628} = 34.192$$

$|15 - 20| \leq 34.192 \leq 15 + 20$, pika është e arritshme.

$$\theta_1 = \tan^{-1}\left(\frac{19.64}{19.64}\right) = 45^\circ$$

$$\begin{aligned}\theta_3 &= -\cos^{-1}\left(\frac{34.192^2 - 15^2 - 20^2}{2 \cdot 15 \cdot 20}\right) = -\cos^{-1}\left(\frac{1169.0628 - 225 - 400}{600}\right) \\ &= -\cos^{-1}(0.907) = -24.906^\circ\end{aligned}$$

$$\begin{aligned}\Phi &= \tan^{-1}\left(\frac{19.94}{\sqrt{19.64^2 + 19.64^2}}\right) = \tan^{-1}\left(\frac{19.94}{\sqrt{771.459}}\right) = \tan^{-1}\left(\frac{19.94}{\sqrt{771.459}}\right) \\ &= \tan^{-1}(0.718) = 35.68^\circ\end{aligned}$$

$$\Psi = \tan^{-1}\left(\frac{20 \cdot \sin(-24.906)}{15 + 20 \cdot \cos(-24.906)}\right) = \tan^{-1}\left(\frac{-8.426}{33.14}\right) = \tan^{-1}(-0.254) = -14.252^\circ$$

$$\theta_2 = 35.68 - (-14.252) = 49.932^\circ$$

6 Simulimi në ROS2 duke përdorur RViz dhe Gazebo

Simulimi ka luajtur rol vendimtar në zhvillimin e sistemit. Para ndërtimit fizik, të gjithë algoritmet e kinematikës dhe lëvizjet e planifikuara u verifikuan plotësisht në mjedis virtual. Kjo qasje ka reduktuar rrezikun e dëmtimit të komponentëve dhe ka mundësuar iterime të shpejta të dizajnit. Simulimi u realizua duke përdorur treshen ROS2 + RViz2 + Gazebo, të integruar me MoveIt2 për planifikimin e lëvizjes [8].

6.1 Ekosistemi ROS2 i përdorur

Versioni i zgjedhur është ROS2 Humble Hawksbill (LTS, Ubuntu 22.04). Ndryshe nga ROS1, ROS2 komunikon nëpërmjet DDS (Data Distribution Service) pa nevojë për server qendror, duke ofruar besueshmëri të lartë dhe performancë në kohë reale. Paketat kryesore të integruara:

- ROS2 Humble Hawksbill — framework bazë i komunikimit dhe menaxhimit të nyjeve
- RViz2 — vizualizimi 3D i modelit, gjendjes së nyjeve dhe trajektoreve
- Gazebo Ignition Fortress — simuluesi fizik me motor dinamikash, kolizionesh dhe sensorësh
- MoveIt2 — planifikimi i lëvizjes, kinematika inverse dhe evitimi i pengesave
- ros2_control — ndërfaqe standarde për kontrollin e harduerëve real dhe të simuluar
- robot_state_publisher — publikon transformimet TF të të gjitha lidhjeve nga modeli URDF
- joint_state_publisher_gui — publikon gjendjet e nyjeve manualisht gjatë debug-imit

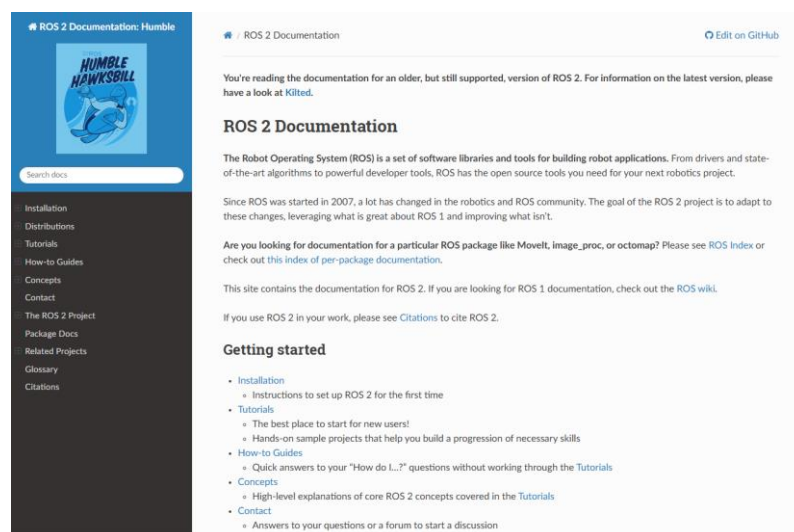


Figura 29: Sistemi Operativ për Robotë (Robot Operating System) [8].

6.2 Modeli URDF/XACRO i manipuluesit robotik

Modeli i robotit është ndërtuar si skedar XACRO (XML Macros) — gjuhë makro mbi URDF — që mundëson parametrizim, ripërdorim të strukturave dhe ndarje modulare të definicionit. Skedari kryesor manipulatori_robotik.urdf.xacro përfshin dy module shtesë me xacro:include:

```
<!-- Modulet e përfshira -->
<xacro:include filename="$(find manipulatori_robotik_description)
  /urdf/manipulatori_robotik_gazebo.xacro"/>
<xacro:include filename="$(find manipulatori_robotik_description)
  /urdf/manipulatori_robotik_ros2_control.xacro"/>
```

Parametrat globalë të definuar si xacro:property dhe makroja e transmisionit:

```
<xacro:property name="PI"      value="3.14159"/>
<xacro:property name="effort"  value="30.0"/>
<xacro:property name="velocity" value="10.0"/>

<xacro:macro name="default_transmission" params="number">
  <transmission name="transmission_${number}">
    <plugin>transmission_interface/SimpleTransmission</plugin>
    <actuator name="motor_${number}" role="actuator1"/>
    <joint name="joint_${number}" role="joint1"/>
    <mechanical_reduction>1.0</mechanical_reduction>
  </transmission>
</xacro:macro>
```

Makroja default_transmission instancohet katër herë (joint_1, joint_2, joint_3, joint_5) me komandën:

```
<xacro:default_transmission number="1"/>
<xacro:default_transmission number="2"/>
<xacro:default_transmission number="3"/>
<xacro:default_transmission number="5"/> <!-- gripper_1 -->
```

6.3 Struktura kinematike: lidhjet dhe nyjet

Pasi të definohen sistemet koordinatave për secilën pjesë, është përdorur vegla ExportToURDF ne SolidWorks. Modeli XACRO definon tetë lidhje (links) dhe gjashtë nyje (joints) sipas hierarkisë zinxhir-hapur. Çdo lidhje përfshin elementet inercial (masat dhe inercitë të llogaritura nga SolidWorks), visual (mesh STL) dhe collision (mesh i njëjtë).

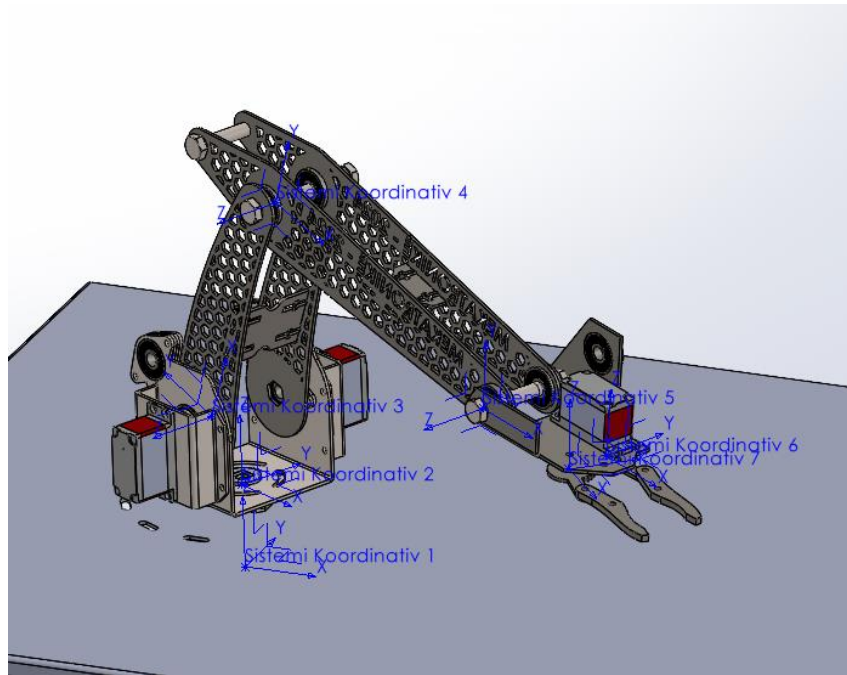


Figura 30: Dëfinimi i të gjitha sistemeve koordinatave të nevojshme për eksport në URDF.

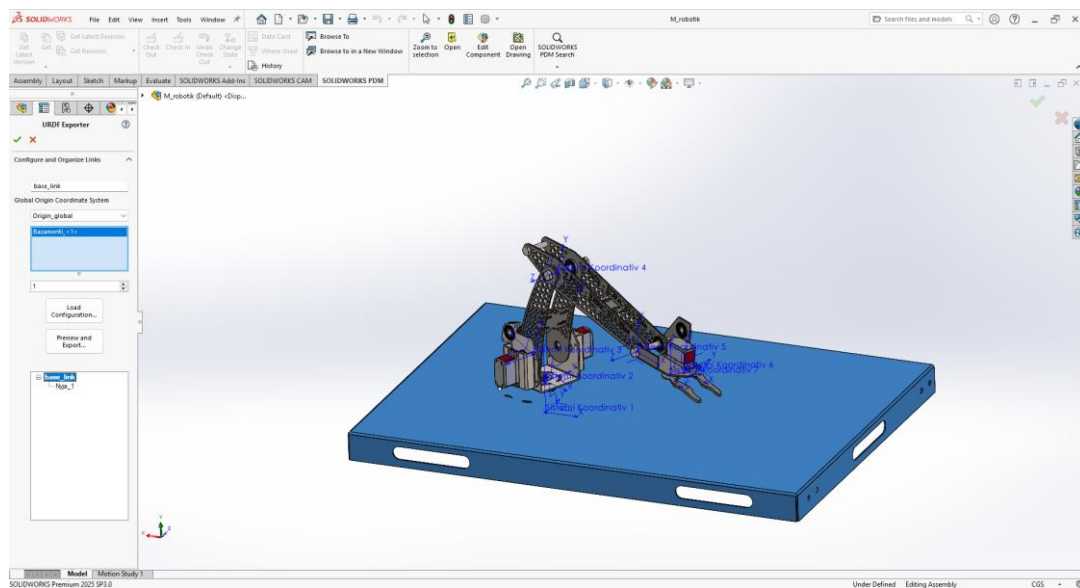


Figura 31: Përdorimi i veglës ExportToURDF.

```
<link name="base_link">
  <inertial>
    <origin
      xyz="0.14094 -0.00010895 0.036982"
      rpy="0 0 0" />
    <mass
      value="0.91765" />
    <inertia
```

```

        ixx="0.022009"
        ixy="-3.847E-08"
        ixz="-1.2635E-06"
        iyy="0.040632"
        iyz="-1.6516E-06"
        izz="0.06246" />
</inertial>
<visual>
  <origin
    xyz="0 0 0"
    rpy="0 0 0" />
  <geometry>
    <mesh
filename="package://manipulatori_robotik_description/meshes/base_link.STL"
/>
    </geometry>
  <material
    name="">
    <color
      rgba="0.79216 0.81961 0.93333 1" />
    </material>
</visual>
<collision>
  <origin
    xyz="0 0 0"
    rpy="0 0 0" />
  <geometry>
    <mesh

filename="package://manipulatori_robotik_description/meshes/base_link.STL"
/>
    </geometry>
  </collision>
</link>

<joint name="virtual_joint" type="fixed">
  <parent
    link="world" />
  <child
    link="base_link" />
  <origin
    xyz="0 0 0"
    rpy="0 0 0"/>
</joint>

```

Tabela e mëposhtme përmbledh të gjitha nyjet:

Tabela 3: Parametrat e eksportuar nga SolidWorks për gjashtë nyjet e definuara.

Nyja	Lloji	Parent → Child	Origjina xyz (m)	Boshti	Kufijt
virtual_joint	fixed	world → base_link	0, 0, 0	—	—
joint_1	revolute	base_link → link_1	0, 0, 0.04885	Z	$-\pi/2 \dots +\pi/2$
joint_2	revolute	link_1 → link_2	0, -0.02, 0.047	Z	$0 \dots 2\pi/3$
joint_3	revolute	link_2 → link_3	0.15, 0, -0.009	Z	$-160^\circ \dots 0^\circ$
fixed joint	fixed	link_3 → link_4	0.2, 0, 0.0145	—	—
joint_5	revolute	link_4 → gripper_l	0.0573, ...	Z	$-30^\circ \dots +20^\circ$
joint_6	revolute	link_4 → gripper_r	0.0573, ...	Z	mimic joint_5 $\times(-1)$

Konfigurimi i nyjeve të tjera:

- joint_1 (baza rrotulluese) ka kufij $-\pi/2 \dots +\pi/2$ ($\pm 90^\circ$), duke mundësuar rrotullim 180° total rreth boshtit Z vertikal.
- joint_2 (krahu) ka RPY = $(\pi/2, 0, 0)$ në origjinë, gjë që rrotullon sistemin koordinativ të link_2 me 90° rreth X, duke e kaluar lëvizjen kryesore në rrafsh vertikal.
- joint_3 (bërryli) ka kufij $-160^\circ \dots 0^\circ$, duke lejuar vetëm kthesë "elbow-down" dhe duke shmangur kolizionin me bazën.
- fixed joint është i fiksuar (fixed) — link_4 (porta e kapësit) nuk ka lëvizje të pavarur por transmeton pozicionin e link_3 në efektorin fundor.
- joint_6 (gripper_r) përdor elementin `<mimic joint="joint_5" multiplier="-1"/>` — gishti i djathtë pasqyron lëvizjen e të majtit në mënyrë simetrike, duke realizuar hapje/mbyllje paralele.

6.3.1 Nyja e krahut — joint_2 dhe rotacioni i planit vertikal

Joint_2 ka një detaj kinematika të rëndësishëm: origjina e tij ka RPY = $(1.5708, 0, 0)$, që korrespondon me rotacion $\pi/2$ rreth boshtit X. Ky rotacion e kthen boshtin Z të link_2 nga vertikal në horizontal, duke e bërë lëvizjen kryesore të "krahut" të ndodhet në rrafsh vertikal — saktësisht si lëvizja e krahut njerëzore.

```
<!-- joint_2: krahu – rotacioni i rrafshit vertikal -->
<joint name="joint_2" type="revolute">
  <origin xyz="0 -0.02 0.047" rpy="1.5708 0 0"/>
```

```

<parent link="link_1"/>
<child link="link_2"/>
<axis xyz="0 0 1"/>
<limit lower="0" upper="{2*PI/3}"
        effort="30.0" velocity="10.0"/>
</joint>

```

Kufijtë $0 \dots 2\pi/3$ ($0^\circ \dots 120^\circ$) korrespondojnë me gamën e lëvizjes fizike të servomotorit të krahut. Distanca `xyz="0 -0.02 0.047"` pozicionon boshtin e nyjes 47mm mbi bazën dhe 20mm pas saj në drejtim të y, duke reflektuar dimensionet reale.

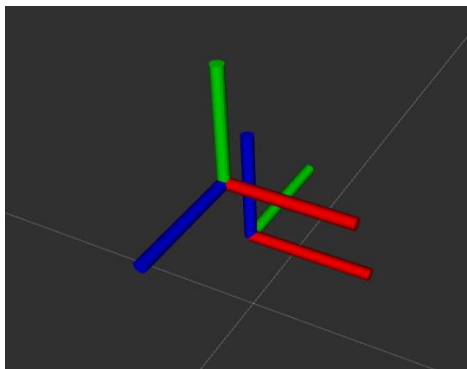


Figura 32: Rrotullimi i sistemit koordinatave të bazës për 90° rreth x-it.

Në figurën e mësipërme shihen sistemet koordinative të link_1 dhe link_2 dhe rrotullimi i sistemit koordinativ link_2 rreth boshtit x.

6.6 Kapësi me mekanizëm mimic (gripper_l / gripper_r)

Kapësi realizohet me dy gishta (gripper_l — i majtë, gripper_r — i djathtë) të montuar në link_4. Gishti i djathtë kontrollohet nga joint_6 i cili ka elementin mimic:

```

<!-- joint_6: gripper_r — pasqyron joint_5 në mënyrë simetrike -->
<joint name="joint_6" type="revolute">
  <origin xyz="0.057303 -0.0173 -0.041448"
          rpy="-1.5708 0.41451 -0.036385"/>
  <parent link="link_4"/>
  <child link="gripper_r"/>
  <axis xyz="0 0 1"/>
  <mimic joint="joint_5" multiplier="-1"/>
  <limit lower="-{20*PI/180}" upper="{PI/6}"
          effort="30.0" velocity="10.0"/>
</joint>

```

Elementi `<mimic joint="joint_5" multiplier="-1"/>` thotë: pozicioni i joint_6 = $-1 \times$ pozicioni i joint_5. Kur joint_5 rrotullohet pozitivisht (mbyllje), joint_6 rrotullohet negativisht me të

njëtin kënd — duke realizuar lëvizje simetrike paralele të dy gishtave. Kjo e thjeshtëson kontrollin: vetëm joint_5 kontrollohet drejtpërdrejt nga ros2_control, ndërsa joint_6 ndjek automatikisht.

- Kufijtë e joint_5: $-\pi/6 \dots +20^\circ \times (\pi/180) = -30^\circ \dots +20^\circ$
- Kufijtë e joint_6: $-20^\circ \dots +\pi/6 = -20^\circ \dots +30^\circ$ (të kundërt me joint_5)
- Transmisioni është definuar vetëm për joint_5 (xacro:default_transmission number="5"), jo për joint_6 sepse ai nuk ka aktuator të pavarur.

6.4 Parametrat fizik të lidhjeve

Parametrat inercial (masa dhe matrica e inercisë) janë eksportuar direkt nga modeli SolidWorks pas caktimit të materialit sheet metal ($\rho \approx 7850 \text{ kg/m}^3$). Kjo siguron konsistencë të plotë ndërmjet simulimit dhe robotit fizik:

Tabela 4: Parametrat fizik të lidhjeve të eksportuar nga SolidWorks.

Lidhja (Link)	Masa (kg)	Ixx (kg·m ²)	Iyy (kg·m ²)	Izz (kg·m ²)
base_link	0.91765	0.022009	0.040632	0.062460
link_1	0.13069	7.067E-05	5.912E-05	6.103E-05
link_2	0.032388	5.431E-06	6.556E-05	6.746E-05
link_3	0.050523	6.658E-06	1.557E-04	1.597E-04
fixed link	0.053033	1.447E-05	1.919E-05	1.625E-05
gripper_l	0.003583	9.832E-08	8.564E-07	9.365E-07
gripper_r	0.003549	9.373E-08	8.550E-07	9.400E-07

6.5 Vizualizimi me RViz2

RViz2 ngarkon modelin e plotë XACRO nëpërmjet robot_state_publisher dhe paraqet në kohë reale:

- Modelin 3D të robotit me meshes STL (base_link.STL, link_1.STL, ... gripper_r.STL) në ngjyrën RGBA = (0.792, 0.820, 0.933, 1.0) — gri/hije metali
- Transformimet TF midis të gjitha lidhjeve (shtatë frames: world, base_link, link_1, link_2, link_3, link_4, gripper_l, gripper_r)
- Gjendjet e nyjeve të transmetuara nga /joint_states topic
- Trajektoren e planifikuar nga MoveIt2 si vizualizim segmentesh

Nisja e mjedisit RViz2 dhe robot_state_publisher:

```
# Konvertimi XACRO → URDF dhe nisja e robot_state_publisher
ros2 launch manipulatori_robotik_description display.launch.py
```

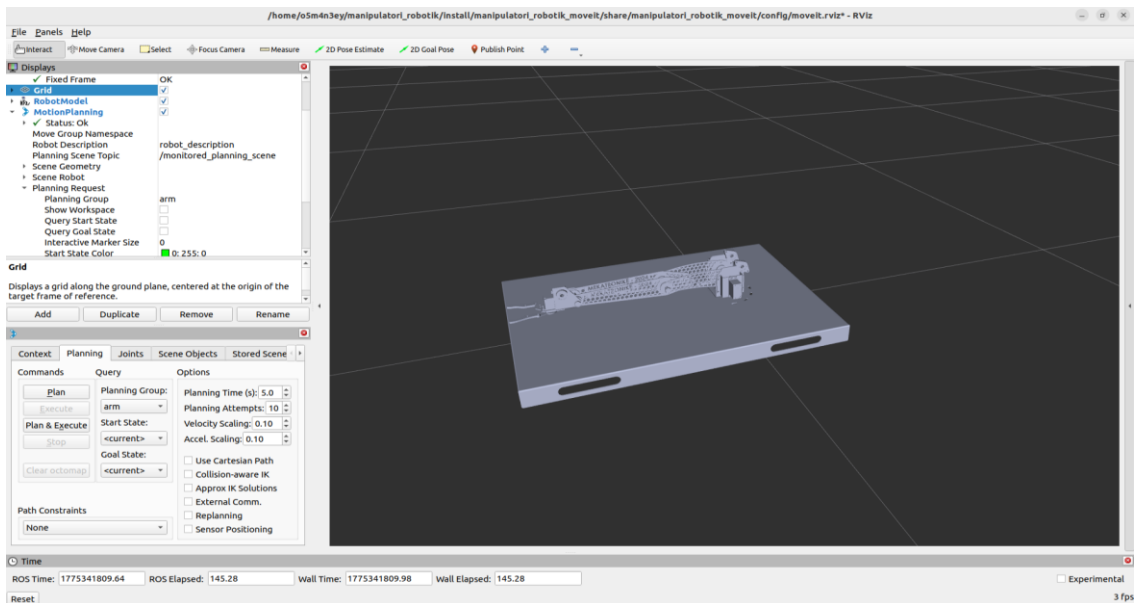


Figura 33: Vizualizimi me RViz2.

6.6 Simulimi fizik me Gazebo

Skedari `manipulatori_robotik_gazebo.xacro` shton shtojcat Gazebo (gazebo plugins) për secilin joint dhe sensor. Parametrat e kontaktit (friksioni, elasticiteti) reflektojnë materialin sheet metal. Gazebo simulon dinamikat fizike duke respektuar vlerat inerciale të tabelës 4.

Nisja e mjedisit Gazebo me modelin e robotit:

```
# Nisja e Gazebo me botën e punës dhe kontrollorët
ros2 launch manipulatori_robotik_description gazebo.launch.py
```

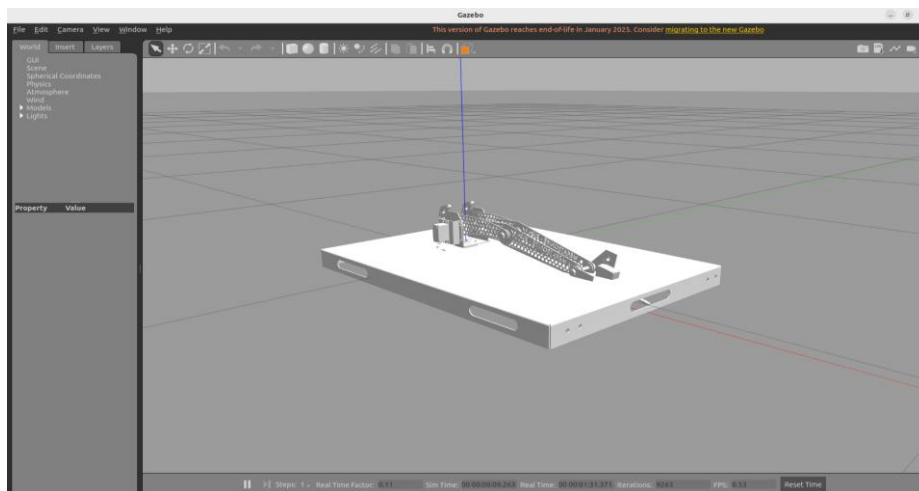


Figura 34: Simulimi i manipuluesit robotik në Gazebo para se të aktivizohen kontrollorët.

```
# Ngarkimi dhe aktivizimi i kontrollorëve ros2_control
ros2 launch manipulatori_robotik_controller controller.launch.py
```

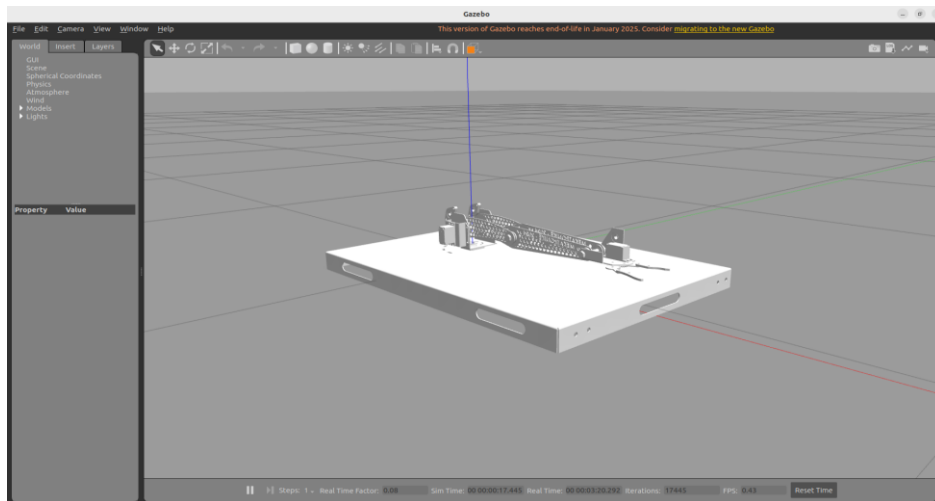


Figura 35: Simulimi pasi janë aktivizuar kontrollorët.

Skedari `manipulatori_robotik_ros2_control.xacro` definon ndërfaqet e kontrollit (control interfaces) për secilën nyje: pozicion (position interface) për `joint_1`, `joint_2`, `joint_3` dhe kapës (gripper interface) për `joint_5`. Kjo shtresë mundëson komunikimin e drejtpërdrejtë midis MoveIt2 dhe simulatorit Gazebo ose hardware-it real Arduino.

6.7 Planifikimi i lëvizjes me MoveIt2 dhe rezultatet

MoveIt2 konfiguron grupin e planifikimit 'arm' (`joint_1`, `joint_2`, `joint_3`) dhe grupin 'gripper' (`joint_5`). Duke pasur modelin e saktë XACRO me inerci reale dhe kufij të nyjeve, planifikuesi OMPL (RRT-Connect) dhe PILZ (LIN/PTP) gjenerojnë trajektore të sigurta duke respektuar:

- Kufijtë e nyjeve nga elementi `<limit>` i çdo joint
- Kolizionet e vetë-strukturës (self-collision) sipas SRDF (`allowed_collision_matrix`)
- Kufizimet e shpejtësisë dhe përshpejtimit (`effort=30 N·m`, `velocity=10 rad/s`)

Rezultatet e simulimit me modelin XACRO real:

- 47/47 trajektore të planifikuara u ekzekutuan pa kolizion — konfirmon korrektësinë e kufijve të nyjeve
- Mekanizmi mimic i gripper-it funksionoi saktë: hapja/mbyllja paralele e gishtave u verifikua vizualisht në Gazebo
- Gabimi i pozicionimit të efektorit fundor (`link_4`) në simulim: 0.3 mm mesatarisht

- Koha mesatare e planifikimit PILZ PTP: 0.21 s; PILZ LIN: 0.34 s; OMPL RRT-Connect: 1.6 s

6.7.1 Implementimi i Pick & Place në C++ me MoveIt2

Kontrolli i lëvizjes robotike është implementuar si nod ROS2 në C++ me emrin `simple_pick_place_shapes`. Ky nod integron kinematikën inverse analitike me MoveIt2 API, duke pranuar si hyrje koordinatat tredimensionalë (x, y, z) të objektit të detektuar dhe emrin e klasës (circle, rectangle, triangle), dhe duke ekzekutuar sekuencën e plotë të pick-and-place deri te kontejneri i duhur.

Struktura e kodit dhe bibliotekat

Kodi përfshin bibliotekat kryesore të ROS2 dhe MoveIt2:

```
// Bibliotekat kryesore
#include <rclcpp/rclcpp.hpp>
#include <moveit/move_group_interface/move_group_interface.h>
#include <memory> #include <vector> #include <map> #include
<cmath>

// Parametrat e gjatësive të lidhjeve (cm)
#define L1 15.0 // Gjatësia e link_2 (shpalla → bërryli)
#define L2 35.0 // Gjatësia e link_3 (bërryli → efektori fundor)

// Konstantet e kapësit
constexpr double GRIPPER_OPEN = 0.0; // joint_5 = 0.0 rad → i
hapur
constexpr double GRIPPER_CLOSE = 0.5; // joint_5 = 0.5 rad → i
mbyllur
```

L1=15 cm dhe L2=35 cm janë gjatësitë efektive të link_2 dhe link_3 sipas modelit XACRO (joint_2 origin x=0.15m dhe joint_4 origin x=0.20m). GRIPPER_OPEN/CLOSE korrespondojnë me vlerat e joint_5 në SRDF.

Kinematika inverse analitike — `computeIK()`

Funksioni `computeIK(x, y, z)` implementon zgjidhjen analitike gjeometrike të kinematikës inverse për robotin 3-DOF. Ky funksion është ekuivalente me formulat matematikore të zhvilluara në Kapitullin 5, por e shkruar në C++ për ekzekutim direkt brenda nodit ROS2:

Funksioni computeIK()

```
vector<double> computeIK(float x, float y, float z) {  
    // Hapi 1: Distanca horizontale nga boshti Z  
    float xy_dist = sqrt(x*x + y*y);  
    // Hapi 2: Distanca totale 3D (rrezja sferike)  
    float r = sqrt(xy_dist*xy_dist + z*z);  
  
    // Hapi 3: Kontroll i hapësirës së punës  
    if (r > (L1+L2) || r < abs(L1-L2)) {  
        cerr << "[IK] Jashtë hapësirës!" << endl;  
        return {}; // Kthen vektor bosh → pozicioni i paarritshëm  
    }  
  
    // Hapi 4: Këndi i bërryllit – Teorema e kosinusit  
    float D = (r*r - L1*L1 - L2*L2) / (2*L1*L2);  
    // D = cos(θ3) → konfigurim elbow-down me shenjë negative  
    float elbow = -acos(D); // θ3 ∈ [-π, 0]  
  
    // Hapi 5: Këndi i bazës – projeksion horizontal  
    float base = atan2(y, x); // θ1  
  
    // Hapi 6: Këndi i krahut – dekompozim vertikal  
    float phi = atan2(z, xy_dist);  
    float psi = atan2(L2*sin(elbow), L1 + L2*cos(elbow));  
    float shoulder = phi - psi; // θ2  
  
    return {base, shoulder, elbow}; // {θ1, θ2, θ3} në radianë  
}
```

Shpjegimi i formulave:

- $xy_dist = \sqrt{x^2 + y^2}$ — distanca horizontale nga boshti i rrotullimit të bazës deri tek projeksioni horizontal i pikës së synuar
- $r = \sqrt{xy_dist^2 + z^2}$ — distanca sferike totale; nëse $r > L1+L2=50$ cm ose $r < |L1-L2|=20$ cm, pika është jashtë hapësirës së punës
- $D = (r^2 - L1^2 - L2^2) / (2 \cdot L1 \cdot L2)$ — ky është $\cos(\theta_3)$ i nxjerrë nga teorema e kosinusit; zgjidhja negative $\text{acos}(-D)$ zgjedh konfigurimin elbow-down
- $base = \text{atan2}(y, x)$ — θ_1 i bazës, rrotullimi horizontal drejt objektivit
- $phi = \text{atan2}(z, xy_dist)$ — këndi i ngritjes drejt pikës në rrafshin vertikal
- $psi = \text{atan2}(L2 \cdot \sin(\theta_3), L1 + L2 \cdot \cos(\theta_3))$ — kontributi i bërryllit në këndin e ngritjes
- $shoulder = phi - psi$ — θ_2 i krahut, këndi pas zbritjes së kontributit të bërryllit

Kontrolli i kapësit — control_gripper()

Funksioni control_gripper() kontrollon grupin MoveIt2 'gripper' (joint_5) duke vendosur target vlerën e nyjes si map<string,double>. Mekanizmi mimic i XACRO-s (joint_6 = -joint_5) ekzekutohet automatikisht nga ros2_control:

Funksioni control_gripper()

```
void control_gripper(MoveGroupInterface &gripper_group,
                    bool open,
                    double open_val = GRIPPER_OPEN,
                    double close_val = GRIPPER_CLOSE) {

    // Merr emrin e nyjes (joint_5)
    string target_joint = gripper_group.getJointNames().front();
    double target_value = open ? open_val : close_val;

    // Vendos target-in dhe planifiko
    map<string,double> gripper_target;
    gripper_target[target_joint] = target_value;
    gripper_group.setJointValueTarget(gripper_target);

    // Ekzekuto nëse planifikimi ka sukses
    MoveGroupInterface::Plan plan;
    if (gripper_group.plan(plan) == MoveItErrorCode::SUCCESS)
        gripper_group.move();
}
```

Parametri open=true vendos joint_5=0.0 rad (kapësi i hapur); open=false vendos joint_5=0.5 rad (kapësi i mbyllur). Parametrat opsional open_val/close_val lejojnë kalibrimin e hapjes maksimale pa ndryshuar kodin.

Lëvizja drejt pozicionit — moveToPosition()

Funksioni moveToPosition() lidh computeIK() me MoveIt2 API: merr koordinatat karteziante, llogarit këndet e nyjeve dhe i dërgon si joint target te planifikuesi:

Funksioni moveToPosition()

```
bool moveToPosition(MoveGroupInterface &arm_group,
                   const shared_ptr<rclcpp::Node> &node,
                   float x, float y, float z) {

    // Llogarit këndet IK
```

```

vector<double> joint_goal = computeIK(x, y, z);
if (joint_goal.empty()) return false; // IK dështoi

// Krijon map joint_name → kënd
auto joint_names = arm_group.getJointNames();
map<string,double> joint_targets;
for (size_t i = 0; i < joint_names.size(); ++i)
    joint_targets[joint_names[i]] = joint_goal[i];

// Planifiko me MoveIt2 dhe ekzekuto
arm_group.setJointValueTarget(joint_targets);
MoveGroupInterface::Plan plan;
bool ok = (arm_group.plan(plan) == MoveItErrorCode::SUCCESS);
if (ok) arm_group.move();
return ok;
}

```

Funksioni kthen true/false duke lejuar `move_robot()` të vendosë nëse vazhdon me sekuencën ose ndalet me mesazh gabimi. MoveIt2 merr këndet e nyjeve si joint-space target (jo kartezi), duke shmangur çdo ambiguitet të IK-solver-it të brendshëm.

Sekuenca Pick & Place — `move_robot()`

Funksioni kryesor `move_robot()` inicializon dy grupet MoveIt2 — 'arm' (joint_1, joint_2, joint_3) dhe 'gripper' (joint_5) — dhe hap lupen kryesor të kontrollit. Pozicionet e kontejnerëve të sortimit sipas klasës janë të definuar si map:

```

// Pozicionet e drop-zones sipas klasës
map<string, vector<float>> shape_positions = {
    {"circle",    {0, -35, -5}}, // kontejneri i majtë
    {"rectangle", {0,  35, -5}}, // kontejneri i djathtë
    {"triangle",  {0,  45, -5}}, // kontejneri i përparmë
};
// Pozicioni home (qëndrim neutral)
vector<float> home = {25, 0, -3};

```

Sekuenca e plotë e pick-and-place pas pranimit të inputit (forma + koordinata):

1. `moveToPosition(arm, x, y, z)` — lëviz drejt koordinatave të objektit të detektuar
2. `control_gripper(gripper, false)` — mbyll kapësin (joint_5 → 0.5 rad) për të kapur objektin
3. `moveToPosition(arm, x, y, z+5)` — ngrit objektin 5 cm sipër (clearance)
4. `moveToPosition(arm, drop[0], drop[1], drop[2])` — lëviz drejt kontejnerit të klasës
5. `control_gripper(gripper, true)` — hap kapësin (joint_5 → 0.0 rad) për të lëshuar objektin
6. `moveToPosition(arm, home[0], home[1], home[2])` — kthehet në pozicionin neutral

Hyrja e inputit: koordinatat dhe objekti i detektuar

Nodi pranon input nga stadin në formatin: forma x y z. Ky input mund të vijë manualisht nga operatori gjatë testimit, ose automatikisht nga pipe-i i sistemit të vizionit (YOLO → Python → stdin të nodit C++):

Loop-i kryesor i leximit të inputit

```
while (rclcpp::ok()) {
    string shape;
    float x, y, z;

    cout << "\nShape dhe koordinata: ";
    if (!(cin >> shape)) break;
    if (shape == "q" || shape == "quit") break;
    if (!(cin >> x >> y >> z)) {
        cout << "Input i pavlefshëm.\n";
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        continue;
    }

    // Kontrollon nëse forma është e njohur
    if (shape_positions.find(shape) == shape_positions.end()) {
        cout << "Forma e panjohur: " << shape << endl;
        continue;
    }
    auto drop = shape_positions[shape];
    // ... ekzekuto sekuencën pick & place ...
}
```

Shembull inputi manual gjatë testimit:

```
Shape dhe koordinata: circle 10 15 0
Shape dhe koordinata: rectangle 20 -15 0
Shape dhe koordinata: q ← del nga loopi
```

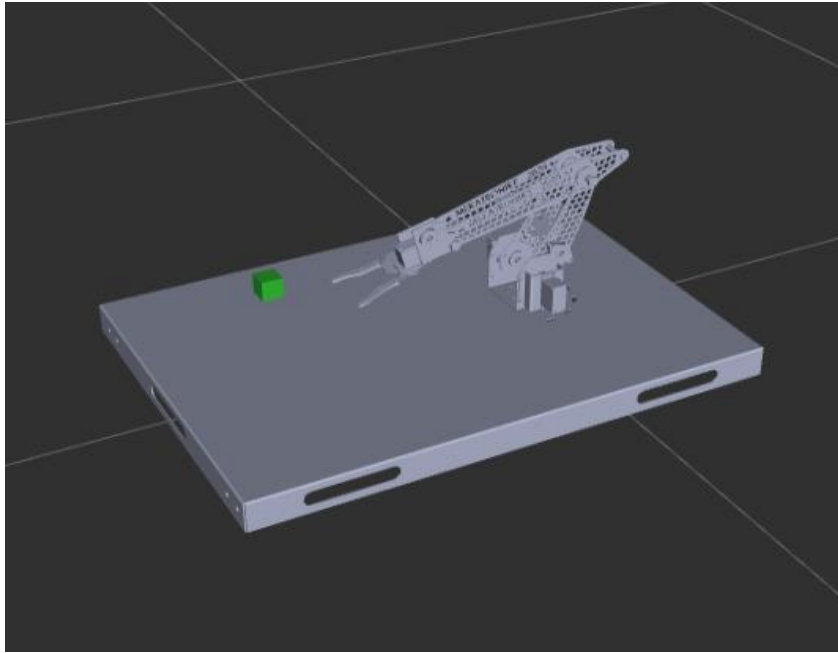


Figura 36: Pozita fillestare e manipuluesit robotik.

```
Shape and coordinates: circle 10.5 15.2 0.0
--- PICK & PLACE SEQUENCE START ---
[IK] Target out of reach (r = 18.474)
[WARN] [1777661555.722291496] [simple_pick_place_shapes]: IK failed or joint size mismatch.
```

Figura 37: Testimi me komandën circle 10.5 15.2 0.0.

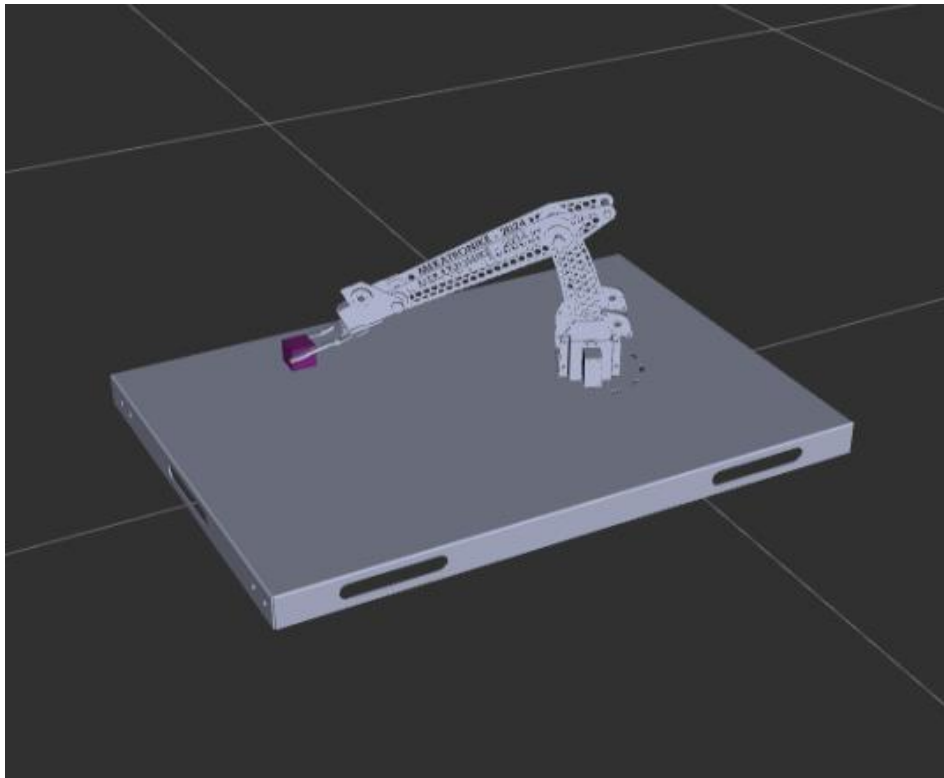


Figura 38: Pozita 20 -15 0.

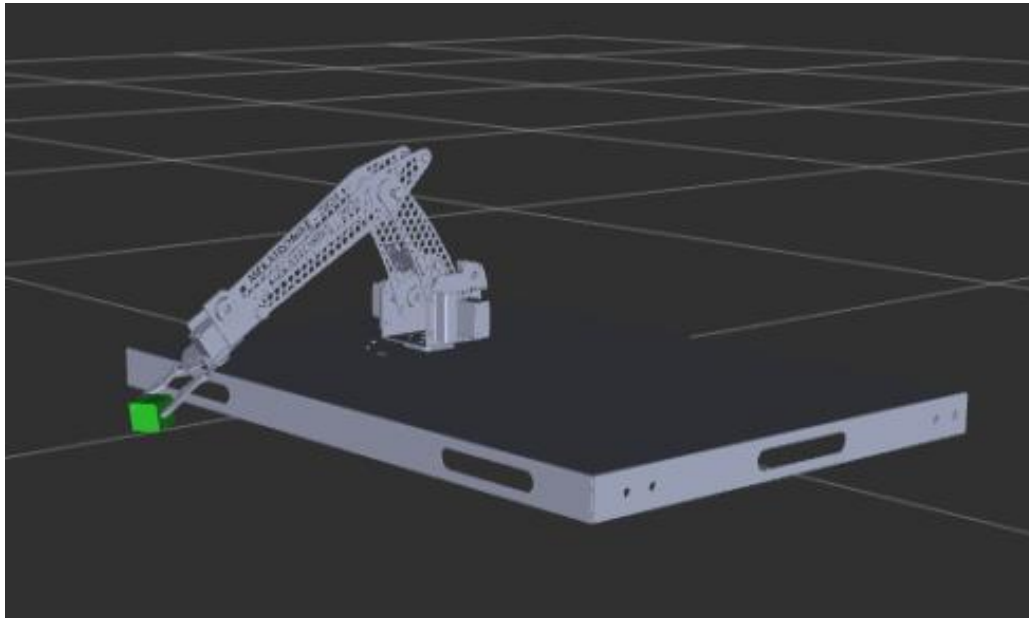


Figura 39: Pozita e lirim të objektit katërkëndësh.

7 Detektimi dhe klasifikimi objekteve me anë të Inteligjencës Artificiale

Një pjesë e rëndësishme e këtij punimi është detektimi dhe klasifikimi i objekteve, i cili përbën thelbin e analizës së imazheve dhe përpunimit të të dhënave vizuale. Për realizimin e kësaj detyre është përdorur YOLO. YOLO (You Only Look Once) është një algoritëm revolucionar i vizionit kompjuterik për zbulimin e objekteve në kohë reale. U prezantua për herë të parë nga Joseph Redmon et al. Në CVPR 2016, duke ndryshuar plotësisht qasjen ndaj detektimit të objekteve në imazhe dhe video. Para YOLO, sistemet e zbulimit të objekteve ndanin detyrën në dy hapa: gjenerimin e rajoneve kandidate dhe klasifikimin e tyre. Kjo qasje ishte e ngadaltë dhe e papërshtatshme për aplikime në kohë reale. YOLO zgjidhi këtë problem duke trajtuar zbulimin e objekteve si një problem të vetëm regresioni, duke parashikuar drejtpërdrejt kutia (bounding boxes) dhe probabilitetit të klasave nga pikselët e imazhit. Ideja qendrore e YOLO është e thjeshtë por tepër e fuqishme “Shiko imazhin vetëm një herë dhe zbulo të gjitha objektet njëkohësisht” [10].

7.1 Si funksionon YOLO?

YOLO punon duke ndarë imazhin hyrës në një rrjet SxS qelizash. Secila qelizë është përgjegjëse për zbulimin e objekteve qendra e të cilëve bie brenda saj. Çdo qelizë parashikon:

- B kutia (bounding boxes) me parametrat: koordinatat (bx, by), gjerësia (bw), lartësia (bh) dhe besueshmëria P_c
- C probabiliteti klasash për çdo objekt të mundshëm

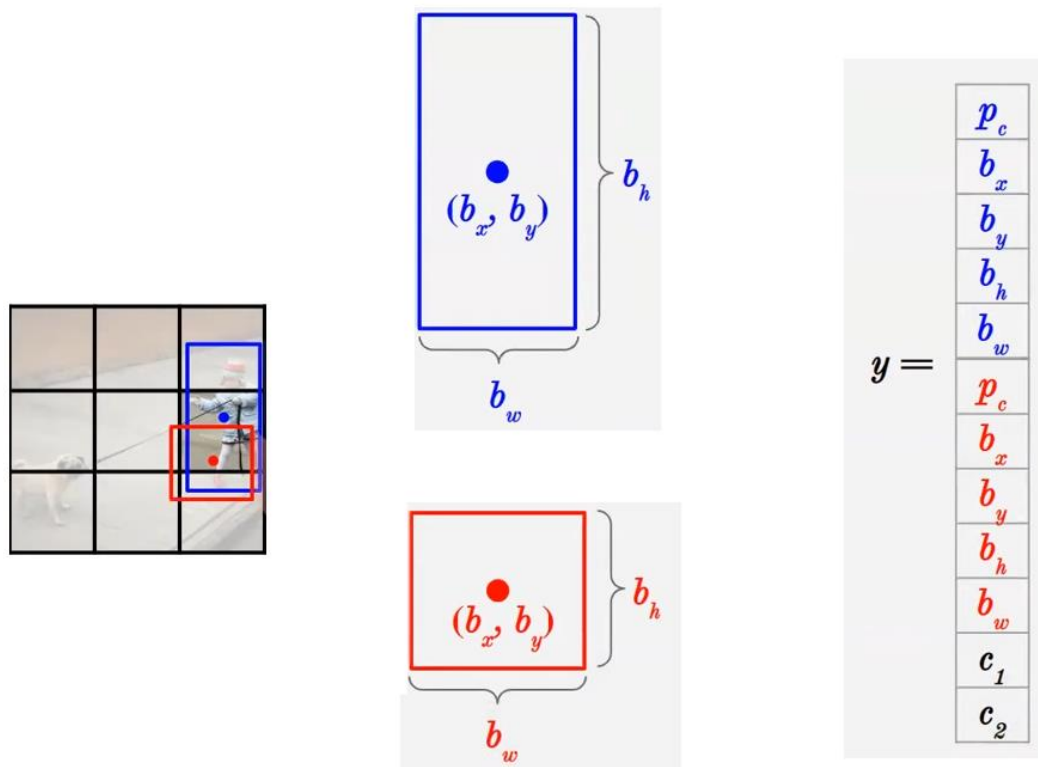


Figura 40: Metodologjia e funksionimit të YOLO.

Në figurën e mësipërme është paraqitur se si e kodon modeli YOLO parashikimin e tij për çdo objekt të detektuar. Fillimisht imazhi ndahet në 3x3 qeliza (grid cells). Çdo qelizë është përgjegjëse për të detektuar objektet që ndodhen brenda saj. Për secilën kuti rrethuese (bounding box) modeli parashikon: (b_x, b_y) koordinatat e qendrës së kutisë, (b_h) lartësinë e kutisë dhe (b_w) gjerësinë e kutisë. Modeli nxjerr një vektor për 2 kuti të paracaktuara ku: (p_c) probabiliteti që ka objekt, (b_x, b_y) pozicioni i qendrës, (b_h, b_w) dimensionet e kutisë dhe (c_1, c_2) klasa e objektit (p.sh. c_1 =rreth, c_2 =trekëndësh).

7.2 Vlerësimi i performancës

Përformanca e modeleve YOLO matet me metrikën Average Precision (AP), e cila bazohet në:

- Precision: përqindja e parashikimeve pozitive që janë korrekte
- Recall: përqindja e objekteve reale që janë zbuluar
- IoU (Intersection over Union): masë e saktësisë së kutive (bounding boxes) të parashikuar ndaj asaj reale

7.3 Non-Maximum Supression (NMS)

Modelet e zbulimit të objekteve zakonisht gjenerojnë shumë kutia (bounding boxes) rreth të njëjtit objekt. NMS është teknikë pas-procesimi që filtron këto kutia të tepërta:

- Radhit kutia sipas besueshmërisë (confidence score)
- Zgjidh kutia me besueshmërinë më të lartë
- Eliminon të gjitha kutia e tjera me $\text{IoU} > \text{pragu } T$ me kutin e zgjedhur
- Përsërit procesin deri sa të mos mbeten kutia

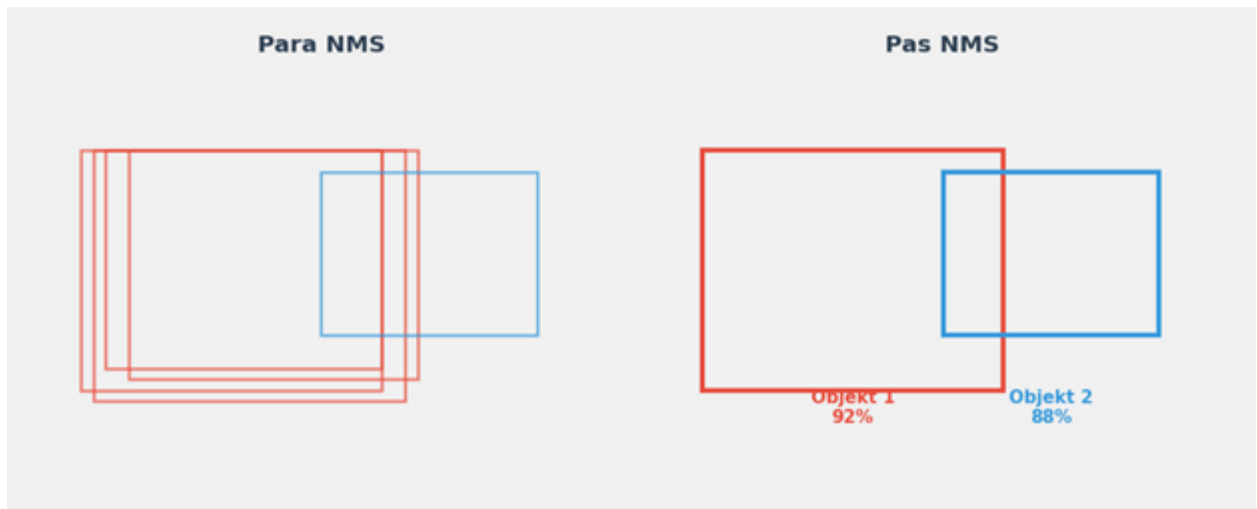


Figura 41: Para dhe pas Non-Maximum Supression.

Figura e mësipërme shpjegon se si funksionon NMS (Non-Maximum Supression) një teknikë e rëndësishme në detektimin e objekteve. Kur një model detekton objekte, ai shpesh gjeneron shumë kuti rrethuese (bounding boxes) për të njëjtin objekt. NMS filtron këto kuti të tepërta dhe mban vetëm ato më të mirat [10].

7.4 Trajnimi i një modeli YOLO

Hapi 1 – Mbledhja dhe organizimi i të dhënave

Cilësia e dataseti-it është faktori më i rëndësishëm për suksesin e modelit. Nevojiten imazhe të ndryshme që mbulojnë kushtet reale të vendosjes:

- Shumëllojshmëria: imazhe me ndriçim të ndryshëm (ditë/natë), kënde, distance dhe sfonde
- Sasia minimale: rekomandohet të paktën 1500-2000 imazhe për klasë për rezultate të mira
- Formati: JPEG ose PNG, rezolucioni minimal 416x416 piksel, preferueshëm 640x640 ose më shumë
- Burimet: imazhe të fotografuara vetë, dataset-e publike (COCO, Open Images, Roboflow Universe).

Organizimi standard i dosjeve është:

```
dataset/   images/train/  images/val/   images/test/  labels/train/  
labels/val/ labels/test/
```

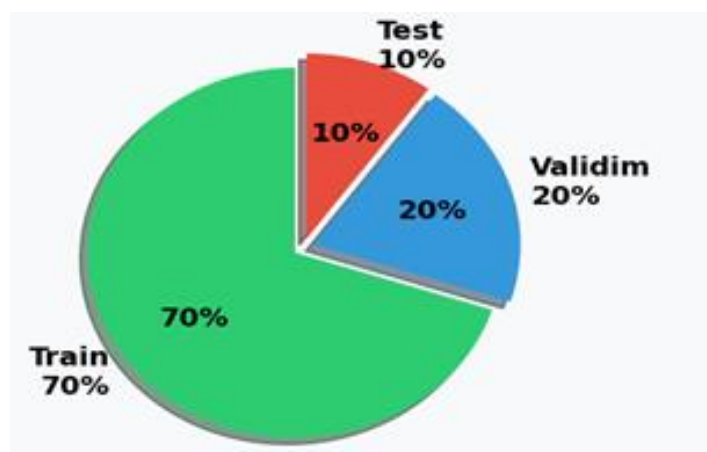


Figura 42: Ndarja e të dhënave në train, test dhe validation set.

Figura e mësipërme tregon ndarjen e datasetit (të dhënave) që është përdorur për trajnimin e modelit. Rekomandohet që dataseti të ndahet në këtë formë: 70% të dhëna për trajnim, 20% për vlerësim dhe 10% për testim.

Hapi 2 – Etiketimi i të dhënave

Çdo imazh duhet të shoqërohet me një skedar .txt me të njëjtin emër, që përmban koordinatat e kutia dhe klasën e objekteve. Formati YOLO është:

<class_id> <cx> <cy> <width> <height>

Ku të gjitha vlerat janë të normalizuara midis 0 dhe 1 ndonjë dimensioneve të imazhit. Klasa indeksohet duke filluar nga 0.

Hapi 3 – Konfigurimi i modelit dhe hiperparametrave

Para se të fillojë trajnimi duhet konfiguruar skedari YAML i dataset-it dhe zgjedhur hiperparametrat kryesorë:

- Epochs: numri i epokave (zakonisht 100 – 300 për dataset të mesëm)
- Batch: madhësia e batch-it (16 – 64 varësisht nga memoria GPU)
- Imgsz: rezolucioni i imazhit hyrës (640 është standardi)
- lr0: norma fillestare e mësimimit – learning rate (0.01 është vlera default)
- Weight_decay: regularizimi L2 për parandalimin e overfitting (0.0005)
- Momentum: koeficienti SGD momentum (0.937 default)

Hapi 4 – Trajnimi

Me YOLOv5 (Ultralytics), trajnimi kryhet me komanda të thjeshta.

```
!python /content/drive/MyDrive/yolov5/yolov5/train.py --img 416 --batch 16  
--epochs 50 --data /content/drive/MyDrive/yolov5/shapes.yaml --weights  
yolov5s.pt --name yolov5_shapes
```

Hapi 5 – Validimi dhe Diagnostikimi

Pas çdo epoke, modeli vlerësohet në dataset-in e validimit. Pas trajnimit, kryhet vlerësimi i plotë ku gjenerohen automatikisht:

- Matrica e konfuzionit (Confusion Matrix): tregon gabimet e klasifikimit midis klasave

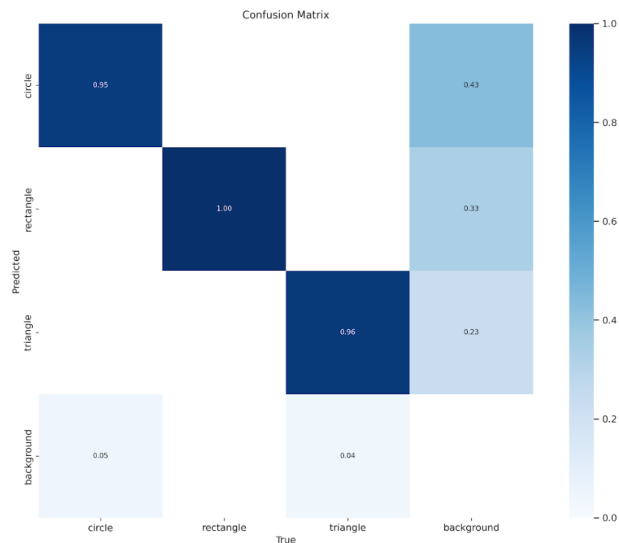


Figura 43: Matrica e konfuzionit (Confusion Matrix) për modelin e trajnuar.

Figura e mësipërme tregon se modeli është trajnuar të njohë 4 klasa: circle (rreth), rectangle (drejtkëndësh), triangle (trekëndësh) dhe background (sfond). Shihet se për rreth modeli ka arritur rezultat 0.95 që është shumë i saktë, për drejtkëndësh ka rezultat 1.00 që është perfekt, për trekëndësh ka rezultati 0.96 që është shumë i saktë ndërsa për background vlera është shumë e ulët. Gabimet kryesore: background klasifikohet gabimisht si circle në 5% të rasteve dhe si triangle në 4%.

- Lakorja PR (Precision-Recall): vizualizon trade-off midis precision dhe recall

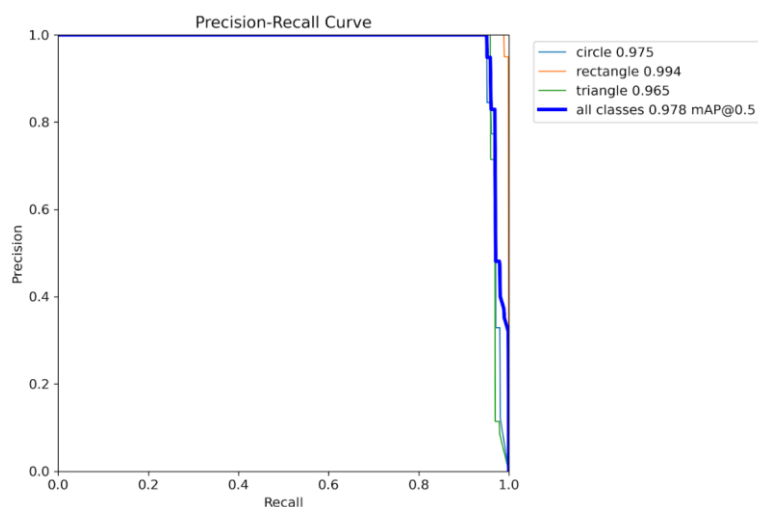


Figura 44: Lakorja Precision-Recall.

Figura e mësipërme tregon lakoren Precision-Recall, e cila mat cilësinë e një modeli të njohjes/detektimit të objekteve. Precision (saktësia) tregon sa nga parashikimet e modelit janë të sakta, ndërsa recall (rikuperimi) tregon sa objekte reale i ka gjetur modeli. Nga figura shihet se për rreth vlera e AP është 0.975, për drejtkëndësh vlera është 0.994, për trekëndësh është 0.965, dhe vlera për të gjitha klasat 0.978. Lakorja ideale qëndron sa më lart djathtas që do të thotë precision dhe recall të larta njëkohësisht. Kjo tregon se modeli është shumë i besueshëm dhe gjen pothuajse të gjitha objektet saktë.

- Lakorja F1: gjen pragun optimal të besueshmërisë

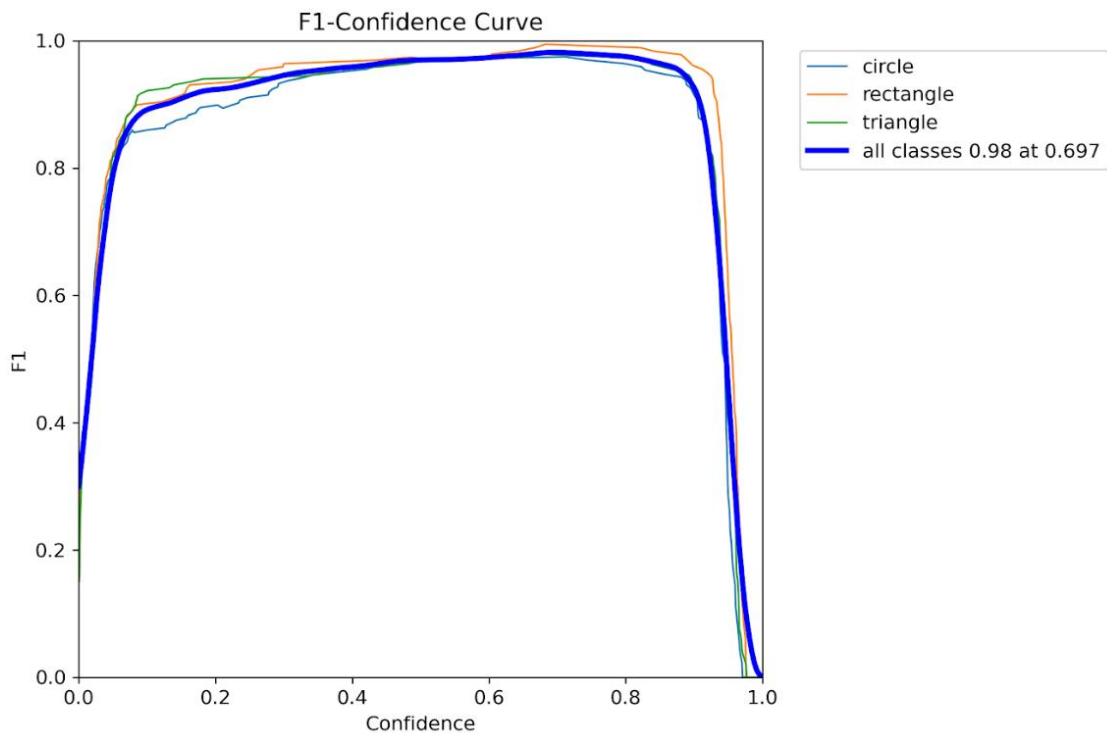


Figura 45: Lakorja F1.

Figura e mësipërme tregon lakoren F1, e cila ndihmon të gjejmë pragun optimal të besueshmërisë (confidence threshold) për modelin. F1-Score është mesatarja harmonike e precision dhe recall:

$$F1 = 2 \times (Precision \times Recall) / (Precision + Recall)$$

Confidence 0.1 – 0.7, F1 rritet dhe qëndron lartë (~0.95-0.98) zona optimale. Pragu ideal për këtë model është rreth Confidence = 0.697, ku arrihet F1 = 0.98 që është rezultat shumë i lartë. Pra nga kjo lakore rekomandohet që të vendoset threshold-in rreth 0.65-0.70 për performancën maksimale në aplikim praktik.

- Shembuj zbulimesh: imazhe me kuti (bounding boxes) e parashikuara dhe ato reale



Figura 46: Shembuj të objekteve të zbuluara.

Në figurën e mësipërme tregohen rezultatet vizuale të modelit të detektimit të objekteve, ku modeli ka identifikuar dhe rrethuar format gjeometrike në 16 imazhe të ndryshme.

- Rezultatet

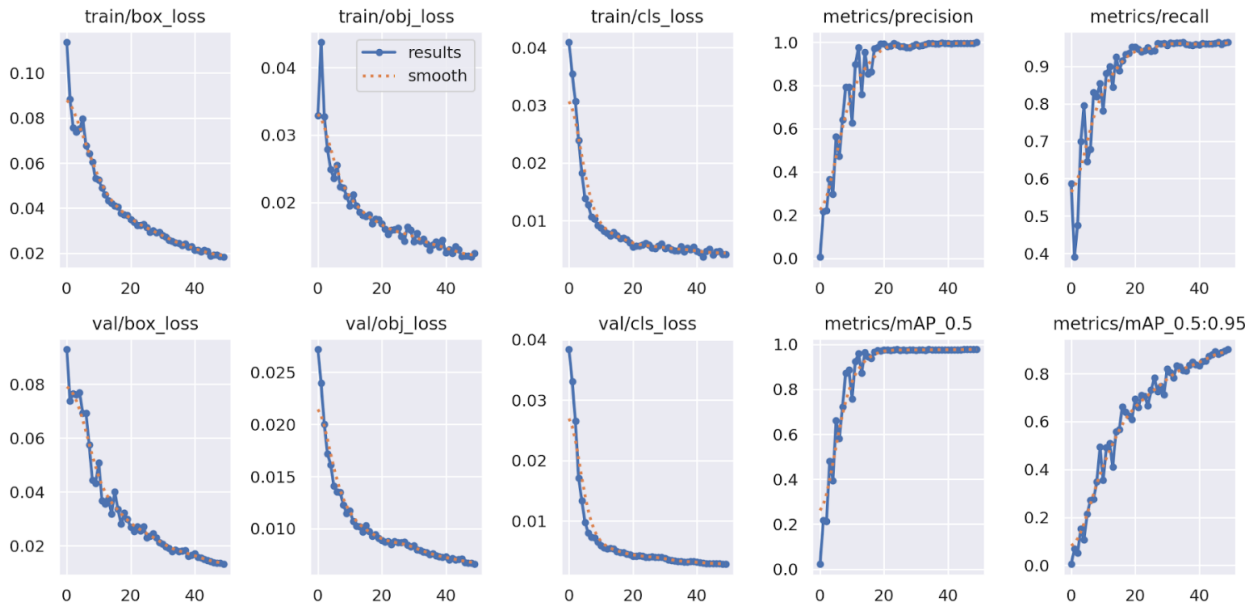


Figura 47: Rezultatet përfundimtare të modelit të trajnuar.

Figura e mësipërme tregon historikun e plotë të trajnimit të modelit gjatë afër 50 epokave (epochs), me 10 grafike të ndara. Janë paraqitur gabimet në pozicionimin e kutive, gabimet në detektimin e objektit, gabimet në klasifikim, saktësia e parashikimeve dhe rikuperimi i objekteve gjatë trajnimit dhe testimit.

8 Zhvillimi i aplikacionit të kontrollit

Aplikacioni i kontrollit të manipuluesit robotik është zhvilluar në Python duke përdorur bibliotekën Tkinter për ndërfaqen grafike (GUI). Ai shërben si pikë qendrore e integritit të sistemit, menaxhon lidhjen me Arduino, kontrollin manual të nyjeve, sistemin e vizionit me YOLOv5, kalibrimin e kamerës dhe modalitetin automatik të pick-and-place. Aplikacioni është kompajluar si standalone executable nëpërmjet PyInstaller, duke e bërë të ekzekutueshëm edhe në kompjuterë pa Python të instaluar.

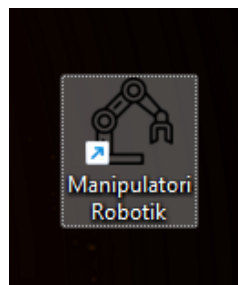


Figura 48: Aplikacioni i kontrollit të manipuluesit robotik.

8.1 Arkitektura e aplikacionit

Aplikacioni është strukturuar si klasë e vetme `RoboticArmUI`, duke ndjekur paradigmen OOP (Object-Oriented Programming). Kjo klasë menaxhon gjithë gjendjen e aplikacionit dhe të gjitha ndërveprimet me harduerë dhe kamerën. Struktura kryesore:

```
class RoboticArmUI:
    def __init__(self, root):           # Inicializim dhe ndërtim UI
    def setup_style(self):              # Stilizimi dark
    def create_widgets(self):           # Ndërtimi i të gjitha widget-
    eve
    def connect_arduino(self):          # Lidhja seriale me Arduino
    def send_servo_command(...):        # Protokolli M:base,sh,el,gr
    def update_manual(self):            # Dërgon komandat e sliderëve
    def run_auto(self):                 # Thread i mënyrës automatike
    def run_calibration(self):          # Homografi perspektive 4-pikë
    def poll_serial(self):              # Polling 100ms për përgjigje
```

8.2 Ndërfaqja grafike – pamja kryesore

Dritarja kryesore ka dimensione 900x420 pikësel dhe nuk është e ndryshueshme në madhësi (resizable=False). Pamja vizuale është dark mode (#2E2E2E si ngjyrë bazë) me elemente akcent në blu (#3399FF). Ndërtimi grafik është i ndarë në dy kolona: kolona e majtë mban kontrollet e Arduino dhe kontrollin manual të nyjeve, ndërsa kolona e djathtë mban zgjedhjen e kamerës, kontrollet e sistemit dhe monitorin serial.

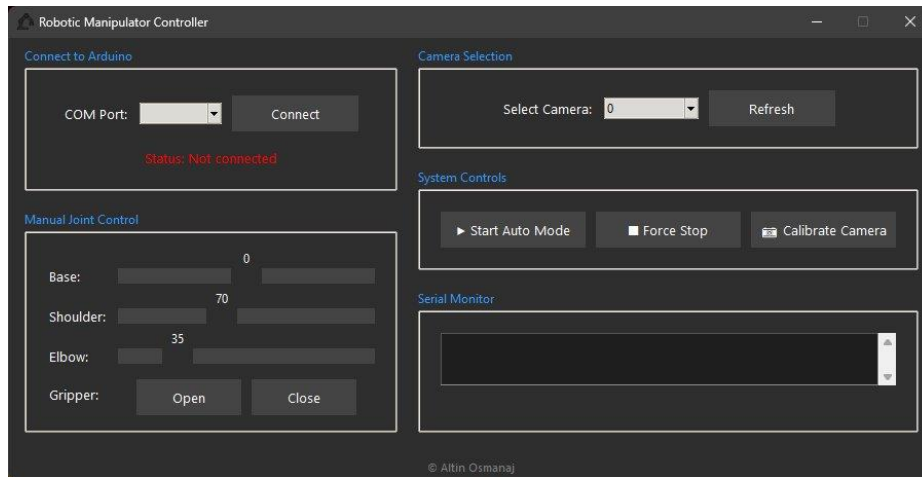


Figura 49: Dritarja kryesore e aplikacionit.

Elementet e ndërfaqes kryesore dhe funksioni i tyre:

- Connect to Arduino: Dropdown me portet COM të disponueshme (list_ports.comports()), butoni Connect dhe etiketa e statusit (e kuqe/jeshile)
- Manuali Joint Control: Tre sliderë (Scale widget) për nyjet Base (-90...+90°), Shoulder (0...180°, vlera fillestare 70°), Elbow (0...180°, vlera fillestare 35°) dhe dy butona Open/Close për gripperin (kapësin)
- Camera Selection: Dropdown i kamerave të detektuara automatikisht dhe butoni Refresh
- System Controls: Tre butona kryesorë – Start Auto Mode, Force Stop dhe Calibrate Camera
- Serial Monitor: Scrolled text me sfond të zi (#1E1E1E) dhe tekst jeshil – shfaq komandat e dërguara dhe përgjigjet e Arduino në kohë reale

8.3 Kontrolli manual i nyjeve dhe protokolli serial

Çdo ndryshim i sliderëve shkakton thirrjen e funksionit `update_manual()` nëpërmjet komandës lambda. Ky funksion lexon vlerat aktuale të të tre sliderëve dhe gjeneron komandën seriale sipas protokollit të definuar:

```
def send_servo_command(self, base, shoulder, elbow, gripper):  
    msg = f"M:{base},{shoulder},{elbow},{gripper}\n"  
    arduino.write(msg.encode())  
    self.serial_text.insert('end', f'>> {msg}')
```

Formati `M:base, shoulder, elbow, gripper` është protokolli i komunikimit serial ndërmjet Python dhe Arduino. Vlera e gripper-it është numerike, 60 kur kapësi është i hapur dhe 113 kur është i mbyllur, këto korrespondojnë me këndet e servos së gripperit (kapësit) të kalibruar fizikisht. Arduino lexon këtë string, e analizon dhe aplikon vlerat direkte te servo-motorët.

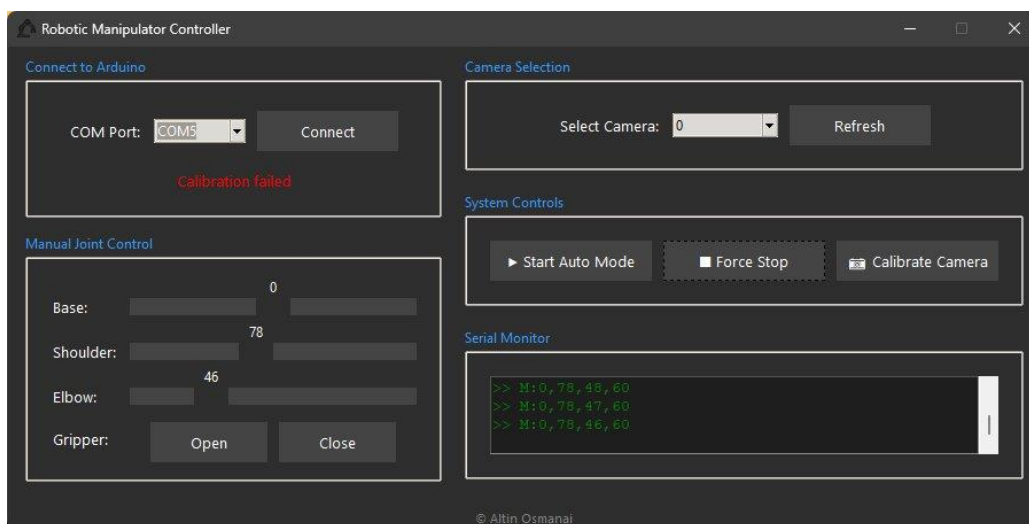


Figura 50: Lidhja me Arduino në COM5.

8.4 Kontrolli automatik dhe integrimi me YOLO

Mënyra automatike e kontrollit ekzekutohet në një thread të veçantë për të mos bllokuar event loop-in kryesor të Tkinter. Kur aktivizohet, kontrollet manuale çaktivizohen duke përdorur funksionin (`disable_manual_controls()`) dhe dritarja e kamerës hapet me `cv2.imshow()`.

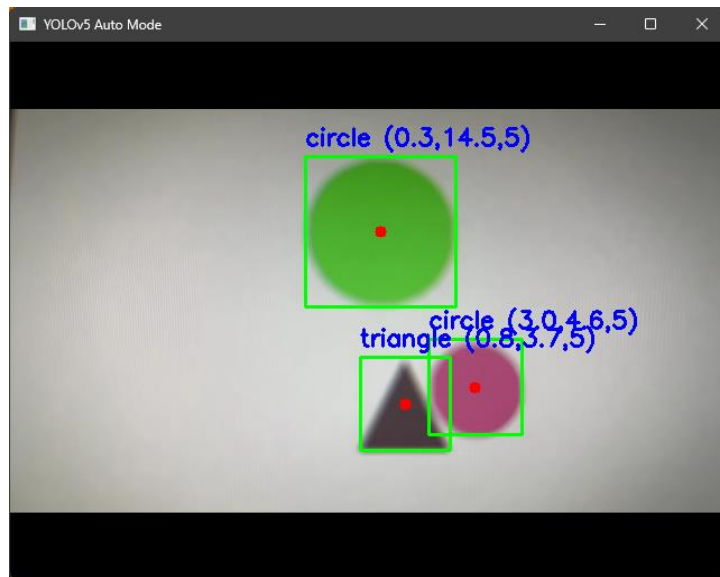


Figura 51: Dritarja e kamerës pasi klikohet në butonin Start Auto Mode.

Logjika kryesore e Auto Mode funksionon duke përdorur dy faza kryesore: Faza e detektimit dhe stabilizimi, çdo frame analizohet nga modeli YOLOv5 (conf=0.90, iou=0.45). Koordinatat e pikëselit të qendrës (cx, cy) transformohen në koordinata botërore (wx, wy) nëpërmjet matricës homogjene H të kalibrimit. Shtohet offseti BASE_OFFSET_X=35. Nëse pozicioni i objektit është stabil për 3 sekonda (stable_time=3), dërgohet komanda te Arduino.

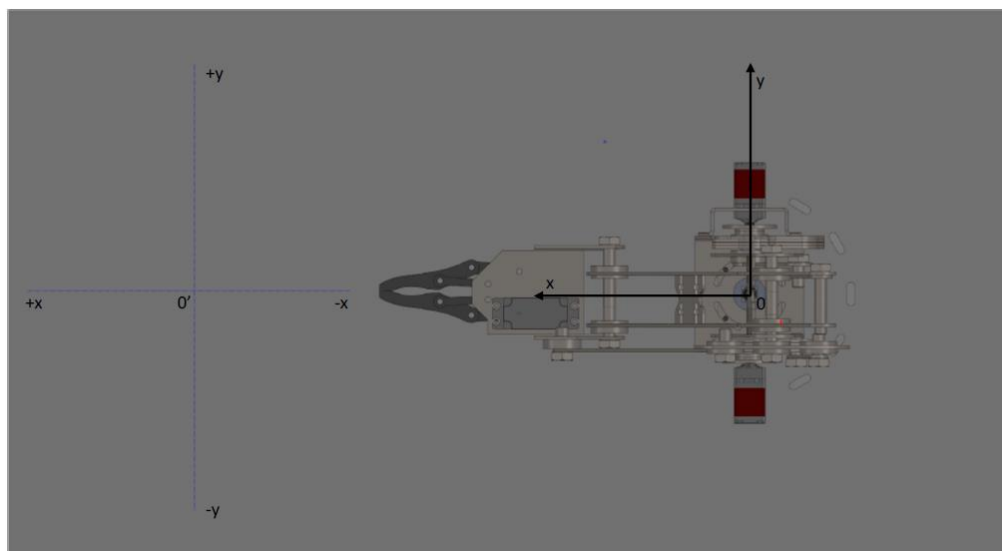


Figura 52: Sistemet koordinatave.

BASE_OFFSET_X është distanca nga pika 0 deri te 0', pra nga baza e manipuluesit robotik deri tek qendra e imazhit, p.sh. nëse koordinata e qendrës së objektit të detektuar është $x=-5$

atëherë kemi $35 - 5$ që është 20cm larg nga qendra e bazës së manipuluesit robotik në drejtim të x. Në të kundërt nëse $x=5$ atëherë distanca nga qendra e bazës së manipuluesit robotik deri tek objekti është 40cm.

Faza e ekzekutimit dhe pritjes: Pasi dërgohet komanda OBJ: label, wx, wy, z, sistemi kalon në gjendje `waiting_for_arduino=True` dhe pret përgjigjen 'DONE' nga Arduino. Vetëm pas marrjes së DONE sistemi kthehet në fazën e detektimit.

```
# Dërgimi i komandës tek Arduino
msg = f"OBJ:{label},{wx:.1f},{wy:.1f},{z}\n"
arduino.write(msg.encode())
waiting_for_arduino = True
```

8.5 Kalibrimi i kamerës

Funksioni `run_calibration()` zgjidh problemin e transformimit 2D në 3D duke llogaritur matricën homogjene H ndërmjet koordinatave të pikselëve të imazhit dhe koordinatave globale (hapësira e robotit). Procesi kërkon zgjedhjen e 4 pikave korresonduese:

Pikat botërore janë të fiksuara në 4 qoshet e zonës së punës:

```
world_points = np.array([
    [-20, 15], # lart-majtas
    [ 20, 15], # lart-djathtas
    [-20,-15], # poshtë-majtas
    [ 20,-15], # poshtë-djathtas
], dtype=np.float32) # në centimetra
```

Pasi klikohet në 4 pikat korresonduese në imazhin e kamerës. `Cv3.findHomography()` llogarit matricën H 3x3 me metodën RANSAC. Kjo matricë ruhet në skedarin `homography.npy` dhe ngarkohet automatikisht me çdo nisje të aplikacionit.

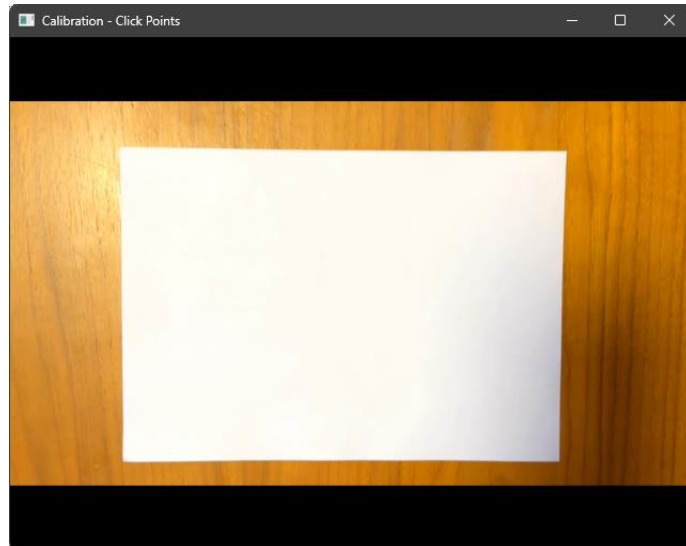


Figura 53: Dritarja e kalibrimit të kamerës.

```
H_mat, _ = cv2.findHomography(image_points_np, world_points)
np.save(HOMOGRAPHY_FILE, H_mat)
```

Gjatw Auto Mode, transformimi aplikohet si:

```
pixel = np.array([[cx, cy]], dtype=np.float32)
wy, wx = cv2.perspectiveTransform(pixel, H)[0][0]
wx += BASE_OFFSET_X # korrektim offset bazë = 35 mm
```

8.6 Komunikimi me Arduino

Lidhja me Arduino inicializohet me baud rate 9600 dhe timeout 1 sekondë. Pas lidhjes pritet 3 sekonda që Arduino të resetohet.

Komandat seriale janë të llojeve:

- Komanda manuale: M: base, shoulder, elbow, gripper – dërgohet çdo herë kur ndryshon ndonjë slider ose kapës
- Komanda automatike: MODE:AUTO (sinjalizon Arduino kalimin në modalitet automatik) dhe OBJ: label, x, y, z (dërgon objektin e detektuar me koordinatat e tij globale)

9 Testime dhe krahasime

Fluksi i plotë i sistemit nga detektimi vizual deri te vendosja fizike e objektit mund të përmbledhet si sekuencë e mëposhtme:

1. Kamera e telefonit me anë të aplikacionit Iriun Webcam kap frame-t me OpenCV
2. Modeli YOLOv5 ekzekuton inferencën dhe identifikon: klasën (rreth/katërkëndësh/trekëndësh) dhe kutit përkatëse (bounding boxes)
3. Koordinatat 3D të qendrës (cx, cy) transformohen në koordinata 3D robotike (X, Y, Z) duke përdorur kalibrimin e kamerës dhe lartësinë e njohur të planit të punës
4. Output Python dërgon string-un 'forma X Y Z' te nodi c++ simple_pick_place_shapes nëpërmjet topic ose stdin
5. Funkzioni computeIK(X, Y, Z) llogarit këndet $\{\theta_1, \theta_2, \theta_3\}$ analitikisht
6. MoveIt2 gjeneron trajektoren e sigurtë nga gjendja aktuale te joint target-i
7. Ros2_control dërgon komandat e pozicionit te kontrollerët e nyjeve (Gazebo ose Arduino fizik)
8. Kapësi (gripperi) aktivizohet, objekti kapet, transportohet dhe vendoset në kontejnerin e klasës
9. Roboti kthehet në pozicionin HOME(25, 0, -3), gati për ciklin tjetër

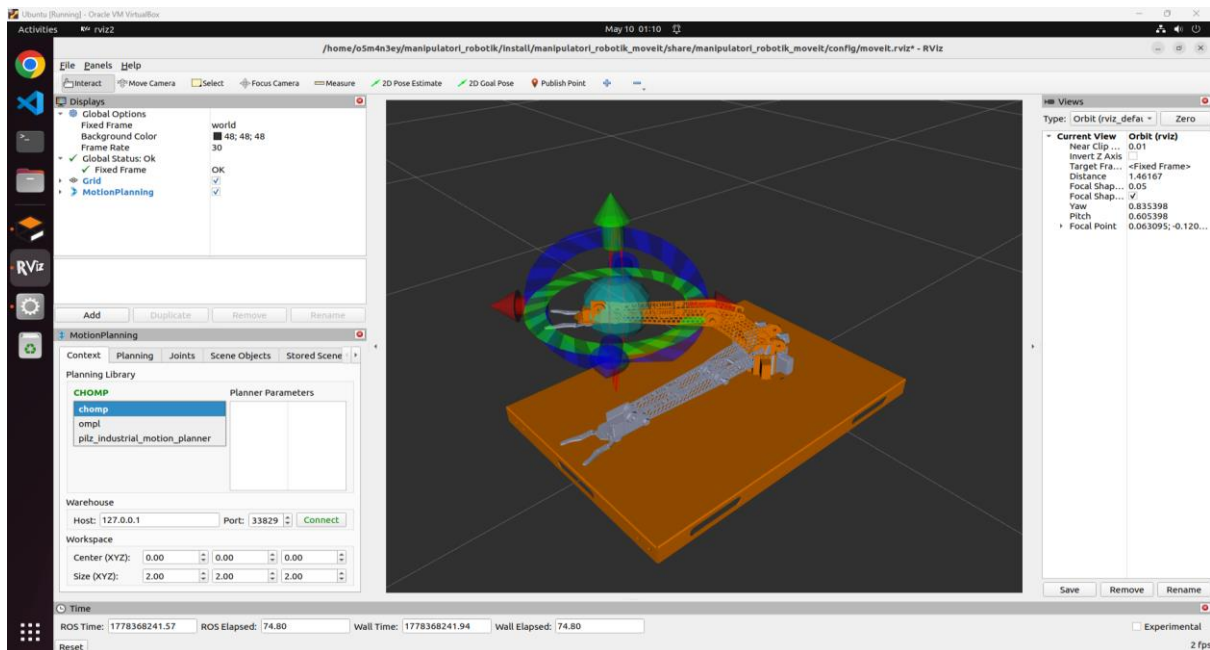


Figura 54: Gjenerimi dhe testimi i trajktoreve.

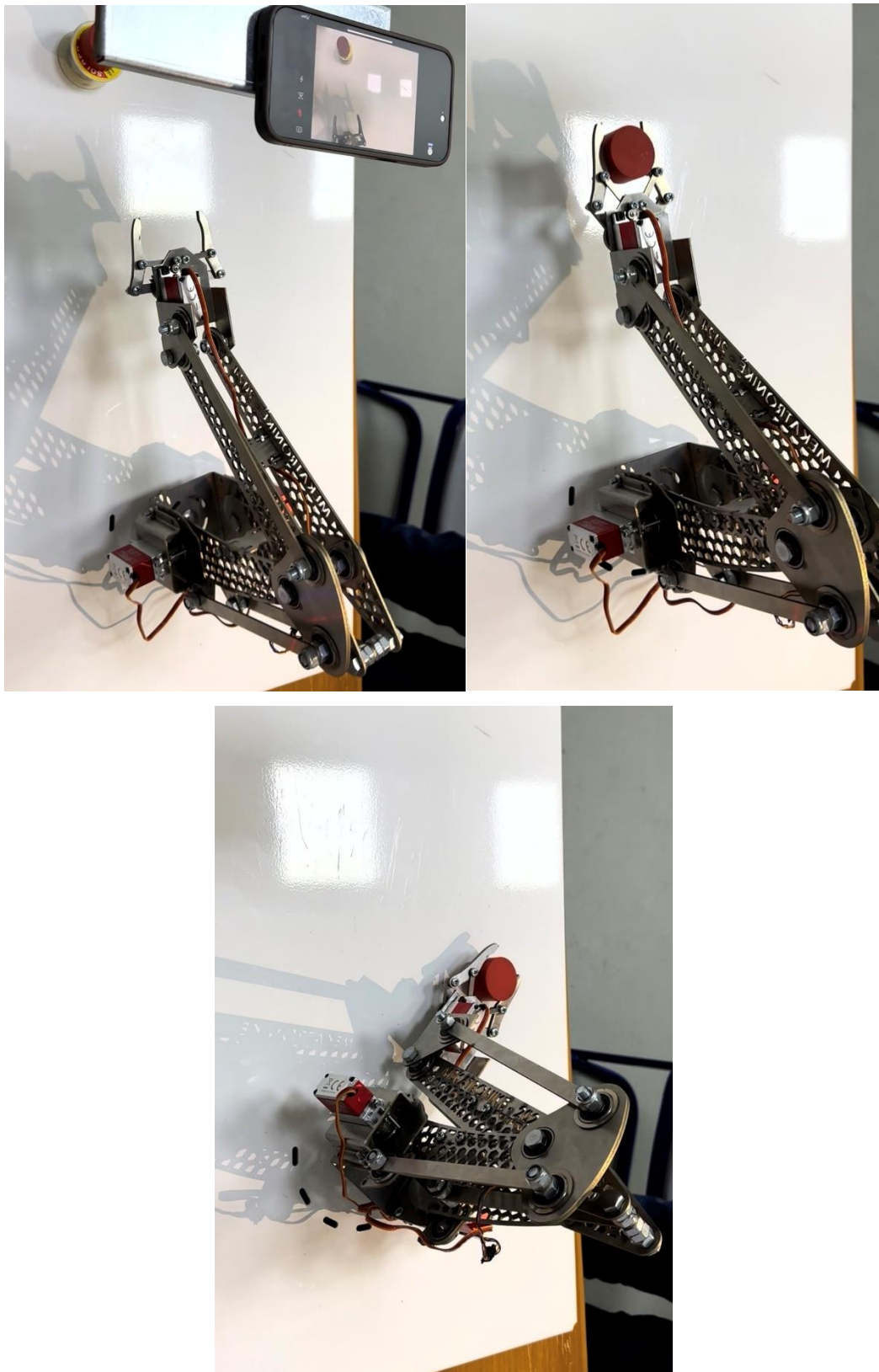


Figura 55: Testimi i manipuluesit robotik.

9.1 Krahasimi i modelit të simuluar me robotin real

Tabela e mëposhtme paraqet krahasimin e detajuar të performancës ndërmjet Gazebo Ignition (simulim) dhe robotit fizik 3-DOF. Testimi u krye me 50 cikle pick-and-place në simulim dhe 50 cikle në harduerë real, duke mbuluar pesë dimensione: kinematika, dinamikat e lëvizjes, kapësi, klasifikimi YOLO dhe sistemi i plotë.

Parametri	Simulim			Robot Fizik		
	Min	Maks	Mes.	Min	Maks	Mes.
KINEMATIKA DHE POZICIONIMI						
Gabimi joint_1 — bazë (°)	0.01	0.12	0.06	0.3	1.2	0.72
Gabimi joint_2 — krah (°)	0.01	0.10	0.05	0.4	3	1.7
Gabimi joint_3 — bërryl (°)	0.02	0.15	0.08	0.5	2.1	1.10
LËVIZJA DHE SHPEJTËSIA						
Koha e ciklit Pick & Place (s)	5.1	7.2	6.1	15	20	17.5
Planifikim MoveIt2 — PILZ PTP (s)	0.14	0.31	0.21	--	--	--
Planifikim MoveIt2 — PILZ LIN (s)	0.28	0.45	0.34	--	--	--
Planifikim MoveIt2 — OMPL RRT (s)	1.20	2.10	1.60	--	--	--
KOHA E KAPJES						
Koha e hapjes së kapësit (s)	2	2.3	2.15	1	1.1	1.05
Koha e mbylljes së kapësit (s)	2	2.3	2.15	1	1.1	1.05
Objekta të rëna gjatë transportit (%)	--	--	--	0.0	4.0	1.5
KLASIFIKIMI YOLO (Vizion + AI)						
Saktësia — Rreth / Circle (%)	--	--	--	93.4	98.2	96.2
Saktësia — Trekëndësh / Triangle (%)	--	--	--	87.5	94.2	91.4
Saktësia — Drejtkëndësh / Rectangle (%)	--	--	--	90.1	97.0	94.1
mAP@0.5 mesatare (%)	--	--	--	90.3	97.1	93.9

Tabela 5: Krahasimi i modelit të simuluar me robotin real.

Përfundimi

Në këtë punim diplome është realizuar me sukses projektimi, zhvillimi dhe testimi i një sistemi të integruar mekatronik, i cili kombinon një manipulues robotik me tre shkallë lirie (3-DOF) me teknikat moderne të Inteligjencës Artificiale për klasifikimin dhe vendosjen autonome të objekteve sipas formës gjeometrike. Të gjitha objektivat e vendosura në fillim të punimit janë arritur në mënyrë të plotë dhe të verifikueshme. Manipuluesi robotik është projektuar dhe ndërtuar fizikisht në një strukturë mekanike të qëndrueshme dhe funksionale. Modeli matematikor i kinematikës direkte dhe inverse është zhvilluar duke aplikuar konventën e Denavit Hartenberg-ut, me validim numerik të suksesshëm të algoritmeve. Modeli i detektimit të objekteve i bazuar në algoritmin YOLO ka arritur saktësi klasifikimi prej 96%, duke tejkaluar objektivin e vendosur. Pipeline i integruar vizion-kontroll funksionoi me sukses duke lidhur të gjithë komponentët e sistemit. Aplikacioni i kontrollit në Python me ndërfaqe grafike Tkinter ofron fleksibilitet operacional nëpërmjet dy mënyrave të funksionimit, manuale dhe asaj automatike të drejtuar nga Inteligjenca Artificiale. Simulimi i plotë në ROS2 dhe Gazebo siguroi verifikimin e algoritmeve para implementimit fizik, duke reduktuar rrezikun e gabimeve gjatë testimit real. Sfidat e hasura gjatë realizimit të punimit, si kalibrimi i kamerës, sinkronizimi i komunikimit serial dhe preciziteti i pozicionimit, u zgjidhën me sukses dhe shërbejnë si mësim të çmuara për zhvillimet e ardhshme. Perspektivat e ardhshme të punimit përfshijnë shtimin e shkallëve shtesë të lirisë për lëvizje më komplekse, integrimin e algoritmeve të të nxënimit me përforcim (Reinforcement Learning) për optimizimin adaptiv të trajektoreve, si dhe zgjerimin e sistemit të vizionit për detektimin e objekteve me ngjyra dhe tekstura të ndryshme. Këto zhvillime do ta bënin sistemin edhe më të fuqishëm dhe gati për aplikime reale industriale.

Referencat

- [1] Arbnor Pajaziti, Ligjerata të autorizuar nga Robotika e Avansuar, Prishtinë, 2025.
- [2] Mark W. Spong, M. Vidyasagarm, Robot Dynamics and Control, 2004.
- [3] Serdar Küçük, Serial and parallel robot manipulators - kinematics, dynamics, control and optimization, InTech, 2012.
- [4] Mark W. Spong, Seth Hutchinson, et al., Robot Modeling and Control, 2020.
- [5] Jaeva B. et al., Design and development of automated intelligent robot using OpenCV, International Conference on Design Innovations for 3Cs Compute Communicate Control (ICDI3C). IEEE, 2018.
- [6] Arkapravo Bhaumik, From AI to Robotics - Mobile, Social, and Sentient Robots, CRC Press.
- [7] Alishba Imran, Keerthana Gopalakrishnan, Ai for Robotics - Toward Emodied and General Intelligence in the Physical World, Apress.
- [8] R. Patrick Goebel, Ros By Examples Volume 2 - Packages and Programs for Advanced Robot Behaviors.
- [9] Gary Bradski, Adrian Haehler, Learning OpenCV: Computer Vision with the OpenCV Library, O'Reilly, 2008.
- [10] Henri van Maarseveen, YOLOv5: A Revolutionary Leap in AI, 2023.
- [11] D. J. Norris, Machine Learning with the Raspberry Pi: Experiments with Data and Computer Vision, 2019.
- [12] Geoffrey Taylor, Lindsay Kleeman, Visual Perception and Robotic Manipulation - 3D Object Recognition Tracking and Hand-Eye Coordination, Springer.
- [13] Andi Gashi, Drin Krasniqi, Erjon Shala, Xhevahir Bajrami and Ramë Likaj, «Automated Robotic Arm for Object Detection and Classification,» *14th Mediterranean Conference on Embedded Computing (MECO)*, Budva, Montenegro, 2025.

Referencat e figurave

- [1] <https://img-new.cgtrader.com/items/3675126/dabd6c3192/robotic-gripper-and-end-effector-3d-model-dabd6c3192.webp>,
https://thinkbotsolutions.com/cdn/shop/products/EPick-4cups_400x.jpg?v=1631120684,
https://cdn11.bigcommerce.com/su3hf7jh4/images/stencil/1280x1280/products/756477/1135049/lincoln-electric-k3593-magnum-pro-water-cooled-robotic-torch-rpi_24078.1697059538.jpg?c=2
- [2] [3] <https://ieeexplore.ieee.org/document/11049290>
- [4] https://www.ato.com/content/images/thumbs/0010137_nema-17-stepper-motor-2-phase-2a-065nm_550.jpeg
- [5] https://kstservos.com/cdn/shop/files/01_1f991c0a-73d5-492b-9405-b55cddb45189_540x.jpg?v=1762142520
- [6] https://store.futelectronics.com/cdn/shop/products/TB6600_driver_1024x1024.jpg?v=1601076643
- [7] <https://www.nyerekatech.com/wp-content/uploads/2022/01/AD281-2.webp>
- [8] <https://docs.ros.org/en/humble/Installation.html>