

**UNIVERSITETI I PRISHTINËS "HASAN PRISHTINA"**  
**FAKULTETI I INXHINIERISË ELEKTRIKE DHE KOMPJUTERIKE**  
**DEPARTAMENTI AUTOMATIKË E KOMPJUTERIZUAR DHE ROBOTIKË**



## **Punim Diplome (BSc.)**

Zhvillimi dhe simulimi i robotit për veprim  
në sipërfaqe të rrafshët

**Mentori:** Lavdim Kurtaj

**Studenti:** Rrezon Pllana

Prishtinë, Maj 2024

## **Përmbledhja**

Kjo tezë paraqet dizajnin dhe simulimin e një roboti shkrimi të thjeshtë por të gjithanshëm me 4 shkallë lirie (DOF) të kontrolluar nga rregullatorët PID për secilën nyje. Një mjedis gjithëpërfshirës simulimi që përfshin planifikimin e trajektores, kinematikën inverse dhe direkte, dinamikën dhe kontrollin me rregullator PID që u zbatuan përmes MATLAB/Simulink. Rezultatet konfirmojnë llogaritjet e sakta, duke demonstruar potencialin për ndërtimin e robotit fizik.

Një kontribut kryesor qëndron në përshkrimin e procesit për integrimin e modeleve CAD me simulimin dinamik. Kjo qasje krijon një strukturë të vlefshme të zbatueshme për sistemet e tjera mekanike, duke promovuar dizajnimin dhe zhvillimin efikas të sistemeve robotike.

Dizajni 4DOF thjeshton ndërtimin, kontrollin dhe përshtatshmërinë, duke e bërë atë një zgjidhje ideale për detyra të ndryshme shkrimi në industri, arsim dhe teknologji ndihmëse.

Simulimi i suksesshëm i funksionaliteteve thelbësore, duke përfshirë kontrollin PID për lëvizje të saktë të nyjeve, hap rrugën për eksplorimin e ardhshëm. Kjo përfshin ndërtimin fizik dhe testimin e robotit, së bashku me zhvillimin e algoritmeve të avancuara të kontrollit për performancë të përmirësuar. Për më tepër, metodologjia e përshkruar ofron një bazë për kërkime të ardhshme në hartimin dhe simulimin e sistemit robotik. Duke kombinuar modelimin CAD me simulimin dinamik, inxhinierët dhe studiuesit mund të optimizojnë në mënyrë efektive sistemet komplekse mekanike, duke nxitur përparimet në teknologjinë e robotikës.

## **Abstract**

This thesis presents the design and simulation of a simple yet versatile 4-degrees-of-freedom (DOF) writing robot controlled by PID controllers for each joint. A comprehensive simulation environment encompassing trajectory planning, inverse and forward kinematics, dynamics, and PID control was implemented in MATLAB/Simulink. The results confirm accurate calculations, demonstrating the potential for constructing the physical robot.

A key contribution lies in outlining the process for integrating CAD models with dynamic simulation. This approach establishes a valuable framework applicable to other mechanical systems, promoting efficient design and development of robotic systems.

The 4DOF design simplifies construction, control, and adaptability, making it an ideal solution for diverse writing tasks across industry, education, and assistive technology.

Successful simulation of core functionalities, including PID control for precise joint movement, paves the way for future exploration. This includes physical construction and testing of the robot, along with the development of advanced control algorithms for enhanced performance. Furthermore, the outlined methodology offers a foundation for future research in robotic system design and simulation. By combining CAD modeling with dynamic simulation, engineers and researchers can effectively optimize complex mechanical systems, propelling advancements in robotics technology.

## Deklaratë mbi originalitetin

Deklaroj se ky Punim Diplome është punimi im origjinal dhe nuk është prezantuar asnjëherë para ndonjë institucioni tjetër për vlerësim. E gjithë literatura dhe burime të tjera që i kam shfrytëzuar gjatë punimit janë të listuara në referenca dhe plotësisht të cituara.

Prishtinë, Fakulteti i Inxhinierisë Elektrike dhe Kompjuterike 2024

---

**Autori:** Rrezon Pllana

**Nënshkrimi**

**Data:** 24.04.2024

---

# Përmbajtja

<b>1</b>	<b>Hyrja</b>	<b>7</b>
1.1	Përshkrimi i problemit . . . . .	8
1.2	Teknologjitë e Përdorura . . . . .	8
1.2.1	Aplikacionet CAD për dizajnim të krahut robotik . . . . .	9
1.2.2	Aplikacionet e Simulimit . . . . .	9
<b>2</b>	<b>Trajtimi dhe zgjidhja e problemit</b>	<b>12</b>
2.1	Dizajnimi i krahut robotik . . . . .	12
2.2	Kinematika . . . . .	15
2.2.1	Kinematika Direkte . . . . .	15
2.2.2	Kinematika Inverse . . . . .	26
2.3	Dinamika . . . . .	31
2.3.1	Formulimi Lagrange-Euler . . . . .	31
2.3.2	Zgjidhja e problemit dinamik . . . . .	33
2.3.3	Sensori i Forcës . . . . .	37
2.4	Planifikimi i rrugës dhe trajektores . . . . .	39
2.4.1	Planifikimi i rrugës . . . . .	39
2.4.2	Planifikimi i trajektores . . . . .	40
2.5	PID rregullatori . . . . .	45
2.5.1	PID rregullatori i paraqitur në Matlab . . . . .	47
<b>3</b>	<b>Rezultatet eksperimentale</b>	<b>49</b>
3.1	Lidhja e blloqeve të shtjelluar më sipër . . . . .	49
3.1.1	Rezultatet Eksperimentale me planifikim të rrugës L . . . . .	50
3.1.2	Rezultatet Eksperimentale të rregullimit të sistemit me rregullator PID . . . . .	52
<b>4</b>	<b>Konkluzioni</b>	<b>58</b>
<b>5</b>	<b>Puna në të ardhmen</b>	<b>60</b>
<b>A</b>	<b>Shtesat</b>	<b>62</b>
A.1	Algoritmi Denavit-Hartenberg . . . . .	62
A.2	Koordinatat e përgjithësuara . . . . .	63
A.3	Sistemet holonomike . . . . .	63
A.4	Efekti i Coriolis . . . . .	63
A.5	Koordinatat homogjene . . . . .	64
A.6	Matricat ndihmëse $Q_i$ . . . . .	64
A.7	Matrica e inercionit . . . . .	65
A.8	Llogaritja e matricës D . . . . .	65

A.9 Llogaritja e elementeve të gravitetit . . . . .	68
A.10 Koeficienti filtrues tek PID rregullatori . . . . .	69
A.11 Diskretizimi i PID rregullatorit . . . . .	69
<b>Bibliografia</b>	<b>71</b>

# Lista e Figurave

2.1	Pjesët e krahut robotik . . . . .	12
2.2	Montimi i krahut robotik . . . . .	13
2.3	Skedarët e gjeneruar nga kodi për importim të krahut robotik nga Onshape . . . . .	14
2.4	Sistemi i gjeneruar nga importi i krahut robotik nga Onshape . . . . .	14
2.5	Nyja që bashkon dorën me veglën . . . . .	14
2.6	Simulimi i krahut robotik i shfaqur në Mechanics Explorer . . . . .	15
2.7	Struktura e thjeshtuar e paraqitur përmes Matlab . . . . .	16
2.8	Paraqitja e krahut robotik me linqe,nyje dhe frames . . . . .	17
2.9	Bllok funksioni i cili përmban kinematikën direkte . . . . .	24
2.10	Krahu robotik planar me tri nyje rrotulluese . . . . .	26
2.11	Krahu robotik planar me dy nyje rrotulluese . . . . .	27
2.12	Zbatimi i teoremës së kosinusit për $\theta_2$ . . . . .	27
2.13	Zgjidhja e $\theta_1$ përmes tangjenteve të këndit $\gamma$ dhe $\beta$ . . . . .	28
2.14	Zgjidhja e $\theta_3$ në mënyrë gjeometrike . . . . .	28
2.15	Bllok funksioni i kinematikës inverse . . . . .	29
2.16	Zgjidhjet e mundëshme për koordinata fundore të njëjta [9] . . . . .	30
2.17	Bllok funksioni i cili kryen analizën dinamike të krahut robotik . . . . .	35
2.18	Matësi i sforcimit piezorezistues me urë Wheatstone-it [12] . . . . .	38
2.19	Lakoret e pozicionit, shpejtësisë dhe nxitimit të një trajektoreje trapezoidale [14] . . . . .	41
2.20	Bllok funksioni i trajektores trapezoidale i librisë Robotics System Toolbox . . . . .	42
2.21	Konfigurimi i bllok funksioni të trajektores trapezoidale . . . . .	42
2.22	Grafiket për pozitën, shpejtësinë, dhe nxitimin e trajektores trapezoidale . . . . .	43
2.23	Bllok funksioni për vizualizimin e trajektores në Simulink . . . . .	43
2.24	Parametrat dhe gjenerimi i komponenteve të bllok funksionit të vizualizimit të trajektores . . . . .	44
2.25	Rruga e përshkruar nga shënuesi . . . . .	44
2.26	Definimi i fitimit proporcional [16] . . . . .	45
2.27	Definimi i fitimit integral [16] . . . . .	46
2.28	Definimi i fitimit derivativ [16] . . . . .	46
2.29	Konstruktimi bazik i PID rregullatorit [17] . . . . .	47
2.30	Lidhja e rregullatorit PID në Matlab . . . . .	48
2.31	Dritarja për modifikimin e parametrave të rregullatorit PID diskret . . . . .	48
3.1	Sistemi komplet i krahut robotik . . . . .	49
3.2	Simulimi i rrugës në formë të shkronjës L . . . . .	50
3.3	Trajektorja për rrugën në formë të shkronjës L . . . . .	50
3.4	Analiza e kinematikës direkte me rrugën në formë të shkronjës L . . . . .	51
3.5	Analiza e dinamikës për rrugën në formë të shkronjës L . . . . .	51

3.6	Analiza e dinamikës për rrugën në formë të shkronjës L e zmadhuar . . . . .	52
3.7	Sistemi i krahut robotik me rregullator PID diskret . . . . .	52
3.8	Veglat për akordim të PID rregullatorit . . . . .	52
3.9	Grafi i cili pritet te fitohet pas akordimit të PID rregullatorit . . . . .	53
3.10	Grafi me rezultatin më ideal i cili pritet te fitohet pas akordimit të PID rregullatorit	53
3.11	Rezultati i eksperimentit për trajektore vertikale brenda 1s . . . . .	54
3.12	Rezultati i eksperimentit për trajektore vertikale brenda 10ms . . . . .	54
3.13	Rezultati i eksperimentit për trajektore horizontale brenda 1s . . . . .	55
3.14	Rezultati i eksperimentit për trajektore horizontale brenda 10ms . . . . .	55
3.15	Rezultati i eksperimentit për trajektore të lëvizjes pen-up brenda 1s . . . . .	56
3.16	Rezultati i eksperimentit për trajektore të lëvizjes pen-up brenda 10ms . . . . .	56
3.17	Rezultati i eksperimentit për trajektoren në formë të shkronjës L brenda 5s . . .	57
3.18	Rezultati i eksperimentit për trajektoren në formë të shkronjës L brenda 10ms . .	57
A.1	PID rregullatori me koeficient filtrues [19] . . . . .	69



# Lista e Tabelave

2.1	Representimi i DH parametrave . . . . .	17
2.2	Parametrat me të cilët krijohet matrica B . . . . .	22
2.3	Parametrat me të cilët krijohet matrica e transformimeve të krahut robotik . . .	23
2.4	Parametrat me të cilët krijohet matrica H . . . . .	23

# Kapitulli 1

## Hyrja

Në një epokë të përcaktuar nga inovacioni teknologjik dhe përparimi shoqëror, integrimi i robotikës në aspekte të ndryshme të jetës njerëzore është bërë gjithnjë e më i përhapur. Ndër aplikimet e panumërta të sistemeve robotike, përdorimi i manipuluesve të robotëve për detyrat e shkrimit është premtues i rëndësishëm. Kjo tezë përpiqet të eksplorojë dizajnin, simulimin dhe ndikimin e mundshëm të një manipuluesi robotik me katër shkallë lirie (4DOF), i cili përbëhet nga një nyje prizmatike dhe tre nyje rrotulluese, të përshtatura posaçërisht për aplikime shkrimi.

Bashkimi i parimeve të inxhinierisë mekanike, algoritmeve të avancuara të kontrollit dhe teknikave të simulimit ka lehtësuar zhvillimin e këtij manipuluesi të specializuar robotik. Nëpërmjet përsëritjeve të përpikta të projektimit dhe simulimeve gjithëpërfshirëse, aftësitë e manipuluesit për të gjeneruar lëvizje të sakta dhe kontinuale shkrimi janë rafinuar, duke i hapur rrugën integritetit të tij në kontekste të ndryshme.

Nevoja për një sistem të tillë është e shumëanshme, duke përfshirë konsideratat teknike dhe shoqërore. Së pari, në industritë ku dokumentacioni i përpiktë, etiketimi ose vërtetimi i nënshkrimit është parësor, kufizimet e natyrshme të dorës së njeriut në konsistencë dhe qëndrueshmëri mund të pengojnë efikasitetin dhe saktësinë. Duke përdorur një robot manipulues të krijuar për detyrat e shkrimit, organizatat mund të zbusin gabimet njerëzore dhe të sigurojnë uniformitet në rezultatet e shkruara, duke rritur kështu efikasitetin dhe besueshmërinë operacionale.

Për më tepër, ndikimi i mundshëm shoqëror i kësaj teknologjie është i thellë, veçanërisht në fushën e aksesueshmërisë. Individët me aftësi të kufizuara, të cilët mund të hasin sfida për shkak të kufizimeve fizike, shpesh përballen me pengesa në aksesin dhe përdorimin e instrumenteve konvencionale të shkrimit. Zbatimi i një manipuluesi robot të krijuar për të akomoduar nevojat të ndryshme të përdoruesve ka potencialin të revolucionarizojë aksesin, duke fuqizuar individët me aftësi të kufizuara të shprehin veten dhe të përfshihen në aktivitete që kërkojnë komunikim me shkrim me pavarësi dhe lehtësi të sapogjetur.

Duke shfrytëzuar përparimet në robotikë dhe teknologjinë e simulimit, kjo tezë synon të sqarojë potencialin transformues të një manipuluesi robotik 4DOF në adresimin e sfidave të aksesueshmërisë dhe rritjen e saktësisë në detyrat e shkrimit. Nëpërmjet analizave gjithëpërfshirëse dhe vërtetimit empirik, do të eksplorohej efikasiteti dhe shkathtësia e këtij sistemi inovativ, duke hedhur bazat për adoptimin dhe integrimin e tij të gjerë në fusha të ndryshme. Në fund të fundit, ky hulumtim përpiqet të kontribuojë në avancimin e zgjidhjeve teknologjike gjithëpërfshirëse që fuqizojnë individët e të gjitha aftësive për të marrë pjesë plotësisht në shoqëri.

## 1.1 Përshkrimi i problemit

Pavarësisht përparimeve në automatizim dhe robotikë, detyrat e shkrimit konvencional vazhdojnë të paraqesin sfida të rëndësishme, veçanërisht në industritë që kërkojnë saktësi dhe përsëritshmëri të lartë. Kufizimet e shkrimit manual, të tilla si mospërputhja, gabimet e shkaktuara nga lodhja dhe barrierat e aksesit për individët me aftësi të kufizuara, nënvizojnë nevojën për zgjidhje inovatore për të rritur efikasitetin dhe përfshirjen.

- **Saktësia dhe konsistenca:** Metodat tradicionale të shkrimit shpesh rezultojnë në ndryshime në gjerësinë e fjalëve, formimin e shkronjave dhe cilësinë e përgjithshme, të cilat mund të rrezikojnë lexueshmërinë dhe profesionalizmin e dokumenteve të shkruara. Operatorët njerëzorë, që i nënshtrohen kufizimeve dhe ndryshueshmërisë fiziologjike, luftojnë për të mbajtur standarde të qëndrueshme të shkrimit, veçanërisht në detyrat me volum të lartë ose të përsëritur. Si pasojë, industritë që mbështeten në dokumentacion të saktë, si prodhimi, logjistika dhe kontrolli i cilësisë, përballen me sfida në sigurimin e uniformitetit dhe besueshmërisë në rezultatet e tyre të shkruara.
- **Efikasiteti dhe produktiviteti:** Detyrat e shkrimit manual janë në thelb kohë dhe punë intensive, veçanërisht kur kërkohen vëllime të mëdha dokumentacioni ose etiketimi. Natyra e përsëritur e këtyre detyrave jo vetëm që çon në joefikasitet, por gjithashtu rrit rrezikun e gabimeve njerëzore dhe tendosjes profesionale. Në industritë e karakterizuara nga orare të ngushta prodhimi dhe standarde të kërkuara të cilësisë, joefikasiteti që lidhet me shkrimin manual mund të pengojë produktivitetin dhe konkurrencën e përgjithshme.
- **Barrierat e aksesueshmërisë:** Individët me aftësi të kufizuara, duke përfshirë ata me dëmtime të lëvizshmërisë ose vështirësi në kontrollin motorik të imët, ndeshen me pengesa të rëndësishme në aksesin dhe përdorimin e instrumenteve konvencionale të shkrimit. Mungesa e dizajnit ergonomik dhe përshtatshmërisë në mjetet tradicionale të shkrimit i përkeqëson këto sfida, duke kufizuar pjesëmarrjen e individëve me aftësi të kufizuara në aktivitetet akademike, profesionale dhe sociale që kërkojnë komunikim me shkrim. Si rezultat, ekziston një nevojë urgjente për zgjidhje teknologjike që promovojnë gjithëpërfshirjen dhe fuqizojnë individët e të gjitha aftësive për t'u angazhuar plotësisht në detyrat me shkrim.

Adresimi i këtyre sfidave të shumëanshme kërkon qasje inovative që kombinojnë saktësinë mekanike, algoritmet e avancuara të kontrollit dhe parimet gjithëpërfshirëse të projektimit. Duke zhvilluar një robot manipulues të përshtatur posaçërisht për aplikacione shkrimi, të pajisur me funksione të kontrollit të saktë të lëvizjes, përshtatshmërisë dhe aksesueshmërisë, është e mundur të kapërcehen kufizimet e shkrimit manual dhe të revolucionarizohet mënyra se si kryhen detyrat e shkruara nëpër industri dhe kontekste të ndryshme.

## 1.2 Teknologjitë e Përdorura

Kjo tezë eksploron zhvillimin dhe simulimin e një sistemi të ri robotik të krijuar për detyrën specifike të shkrimit. Projekti është i ndarë në dy komponentë kryesorë: dizajnimi i krahut robotik dhe modelimi dhe simulimi i krahut robotik të cilat përdorin aplikacionet e më poshtme.

### 1.2.1 Aplikacionet CAD për dizajnim të krahut robotik

Për dizajnimin e krahut robotik është përdorur aplikacioni OnShape i cili është një aplikacion (CAD - Computer-aided design) që përdoret për dizajnimin e sistemeve mekanike 3D, duke përfshirë manipuluesit e robotëve [1]. Arsytet pse është përdorur OnShape si CAD aplikacion për dizajnim të krahut robotik janë:

1. **Aksesueshmëria e bazuar në renë kompjuterike (Cloud-Based Accessibility):** Qasja në projekte nga çdo pajisje me internet, duke promovuar bashkëpunimin dhe punën në distancë [1].
2. **Bashkëpunimi në kohë reale:** Përdorues të shumtë mund të punojnë njëkohësisht, duke eliminuar problemet e kontrollit të versionit dhe duke përshpejtuar procesin e projektimit [1].
3. **Modelimi parametrik:** Krijon modele inteligjente 3D me marrëdhënie të paracaktuara për eksplorim dhe optimizim të shpejtë të dizajnit [1].
4. **Modelimi i montimit dhe simulimi i lëvizjes:** Simuloni lëvizjen dhe analizon performancën për të identifikuar problemet në fillim, duke reduktuar kohën dhe koston e zhvillimit [1].
5. **Kontrolli i versionit dhe historia e rishikimeve:** Gjurmon automatikisht ndryshimet dhe lejon mundësin e kthimit në versionet e mëparshme nëse është e nevojshme, duke siguruar integritetin e dizajnit [1].
6. **Mjetet e integruara të simulimit dhe analizës:** Vlerëson integritetin dhe performancën strukturore, duke çuar në dizajne të optimizuara [1].
7. **Lehtësia e të mësuarit dhe e përdorimit:** Ndërfaqja intuitive dhe burimet gjithëpërfshirëse të të mësuarit kujdesen për përdoruesit e të gjitha niveleve [1].
8. **Shkallueshmëria dhe fleksibiliteti:** I përshtatshëm për projekte të të gjitha madhësive, duke akomoduar në mënyrë efikase nevojat e projektimit në zhvillim [1].

### 1.2.2 Aplikacionet e Simulimit

Sa i përket modelimit dhe simulimit të krahut robotik aplikacionet e përdorura janë Matlab dhe Simscape Multibody të cilat janë mjete të fuqishme softuerike që përdoren zakonisht për modelim dinamik, simulim dhe analizë të sistemeve mekanike, duke përfshirë robotët. Po ashtu e mira e këtyre aplikacioneve është që mund të përdoren në ndërlidhje me OnShape (apo cilindo aplikacion tjetër CAD) [2].

#### Matlab

Matlab është një gjuhë programimi e nivelit të lartë dhe një mjedis kompjuterik i përdorur gjerësisht në aplikimet inxhinierike dhe shkencore [2]. Ai ofron një grup të pasur funksionesh të integruara dhe kuti veglash për llogaritjen numerike, analizën e të dhënave, përpunimin e sinjalit dhe më shumë. Në kontekstin e sistemeve mekanike, Matlab ofron aftësi të gjera për modelim matematikor, simulim dhe dizajn kontrolli. Karakteristikat kryesore të Matlab përfshijnë:

- **Llogaritja numerike dhe vizualizimi i të dhënave:** Matlab ofron mjete të fuqishme për kryerjen e llogaritjeve numerike dhe vizualizimin e të dhënave, duke e bërë atë ideal për analizimin dhe interpretimin e rezultateve të simulimit [2].
- **Zhvillimi i algoritmit:** Matlab lejon përdoruesit të zhvillojnë dhe zbatojnë algoritme të personalizuar për zgjidhjen e problemeve komplekse inxhinierike, të tilla si planifikimi i trajektores, optimizimi dhe dizajni i kontrollit [2].
- **Simulimi dhe modelimi:** Matlab mbështet simulimin e sistemeve dinamike duke përdorur modele të vazhdueshme dhe diskrete, duke u mundësuar inxhinierëve të simulojnë sjelljen e sistemeve mekanike në kushte të ndryshme [2].
- **Dizajni i sistemit të kontrollit:** Matlab ofron mjete të avancuara për projektimin dhe analizimin e sistemeve të kontrollit, duke përfshirë teknika të tilla si kontrolli PID, kontrolli i hapësirës së gjendjes dhe kontrolli optimal [2].

### Simscape Multibody

Simscape Multibody është një kuti veglash e specializuar për Matlab që u mundëson inxhinierëve të modelojnë dhe simulojnë sistemet mekanike duke përdorur një qasje të modelimit fizik me shumë fusha [3]. Ai siguron një bibliotekë të komponentëve të parandërtuar që përfaqësojnë elemente mekanike, elektrike, hidraulike dhe të tjera fizike, duke i lejuar përdoruesit të mbledhin modele komplekse të sistemit shpejt dhe me lehtësi [3]. Karakteristikat kryesore të Simscape Multibody përfshijnë:

- **Simulimi i dinamikës së shumë trupave:** Simscape Multibody u mundëson përdoruesve të modelojnë kinematikën dhe dinamikën e sistemeve mekanike të ndërlydhura, duke përfshirë trupat e ngurtë, nyjet, kufizimet dhe forcat [3].
- **Qasja e modelimit fleksibël:** Simscape Multibody mbështet teknikat e modelimit simbolik dhe numerik, duke ofruar fleksibilitet në përfaqësimin e sjelljeve dhe ndërveprimeve komplekse të sistemit [3].
- **Integrimi CAD:** Simscape Multibody i lejon përdoruesit të importojnë modele CAD nga mjete të jashtme softuerike, si OnShape, dhe t'i konvertojnë ato në modele të gatshme për simulim për analizë dhe simulim [3].
- **Mjetet e vizualizimit dhe analizës:** Simscape Multibody ofron mjete për vizualizimin e rezultateve të simulimit, duke përfshirë animacione, komplete dhe grafikë, duke i lejuar inxhinierët të fitojnë njohuri mbi sjelljen dhe performancën e sistemit [3].

Sa i përket përdorimit të Matlab dhe Simscape Multibody në lidhje me OnShape në këtë tezë është bërë në këto hapa të listuar në vijim:

- **Integrimi CAD:** OnShape i lejon përdoruesit të krijojnë modele të detajuara CAD të sistemeve mekanike, duke përfshirë manipuluesit e robotëve, me aftësi modelimi parametrik. Pasi modeli CAD të krijohet në OnShape, ai mund të eksportohet në formate të ndryshme skedarësh, si STEP ose STL [4].

- **Importi i modelit:** Modeli i eksportuar CAD nga OnShape mund të importohet në Matlab/Simscape Multibody duke përdorur funksionet ose mjetet e duhura të importit. Simscape Multibody ofron mbështetje të drejtpërdrejtë për importimin e modeleve CAD nga softuer të ndryshëm CAD, duke përfshirë OnShape, duke e bërë të lehtë konvertimin e modeleve CAD në modele të gatshme për simulim [4].
- **Simulimi dhe analiza:** Pasi modeli CAD të importohet në Simscape Multibody, Matlab/Simscape Multibody është përdorur për të kryer simulimin dinamik dhe analizën e sistemit mekanik të krahut robotik. Kjo përfshin simulimin e lëvizjes, dinamikës dhe kontrollit të krahut robotik në kushte të ndryshme, të tilla si ngarkesa të ndryshme, trajektore ose hyrje kontrolli.
- **Dizajni i kontrollit:** Matlab ofron mjete të avancuara për dizajnimin dhe analizën e sistemit të kontrollit, të cilat mund të përdoren së bashku me Simscape Multibody për të dizajnuar dhe optimizuar algoritmet e kontrollit për manipuluesit e robotëve. Ku mund të zhvillohen dhe vërtetohen strategji kontrolli për të arritur specifikimet e dëshiruara të performancës, të tilla si gjurmimi i trajektores, stabiliteti dhe qëndrueshmëria.

Në përgjithësi, integrimi i Matlab/Simscape Multibody me OnShape ofron një rrjedhë pune gjithëpërfshirëse për projektimin, simulimin dhe analizimin e sistemeve mekanike, duke përfshirë manipuluesit e robotëve. Duke kombinuar aftësitë e modelimit CAD të OnShape me aftësitë dinamike të simulimit dhe të kontrollit të dizajnit të Matlab/Simscape Multibody, ka qenë e mundur të zhvillohet dhe optimizohet krah robotik në mënyrë efikase dhe efektive.

# Kapitulli 2

## Trajtimi dhe zgjidhja e problemit

### 2.1 Dizajnimi i krahut robotik

Siç është diskutuar edhe më sipër për projektimin (dizajnimin) e krahut robotik është bërë duke përdorur aplikacioni OnShape ku së pari janë dizajnuar secila komponente veçmas dhe pastaj është bërë montimi i pjesëve. Pjesët përbërëse të krahut robotik janë: baza prizmatike, baza e krahut robotik me nyje rrotulluese, linku i parë apo krahu (shoulder), linku i dytë apo parakrahu (forearm), linku i tretë apo dora (hand) dhe vegla (tool) në rastin tonë marker (shiko Figurën 2.1).

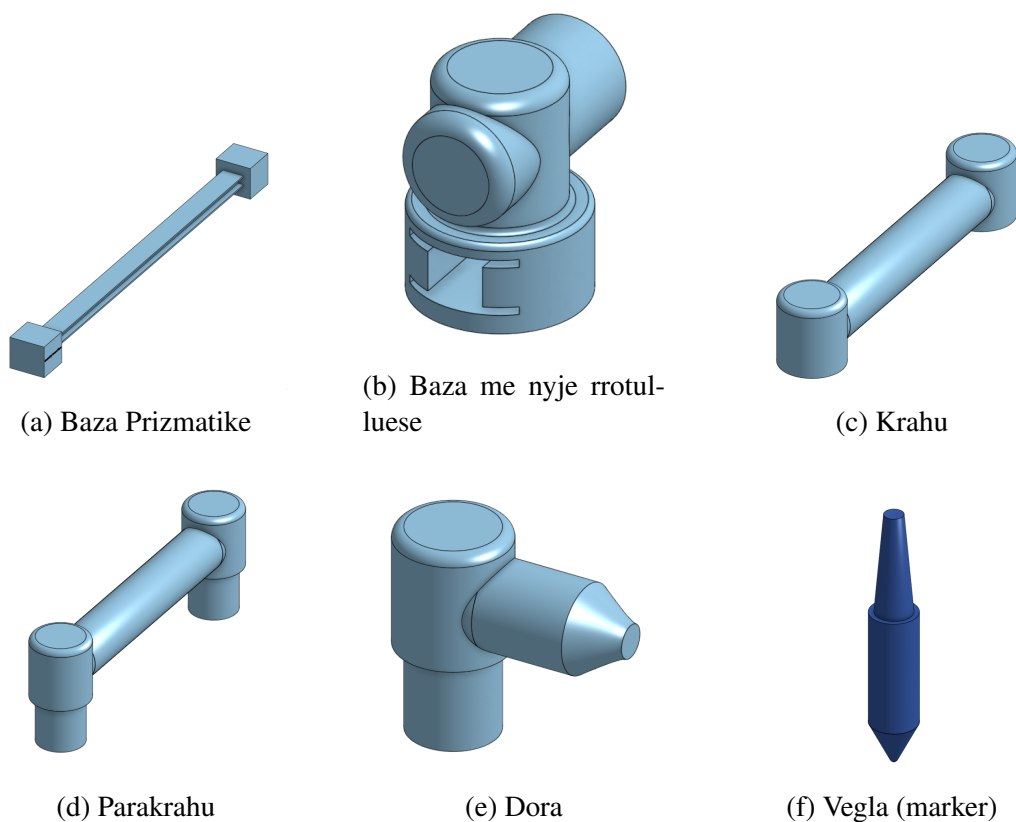


Figura 2.1: Pjesët e krahut robotik

Montimi i pjesëve të dizajnuara më sipër mund të bëhet po ashtu tek aplikacioni OnShape i

cili po ashtu mundëson edhe simulimin e lëvizjes për tu siguruar se pjesët janë montuar saktë. Pas montimit krahu robotik merr formën sikur në Figurën 2.2.

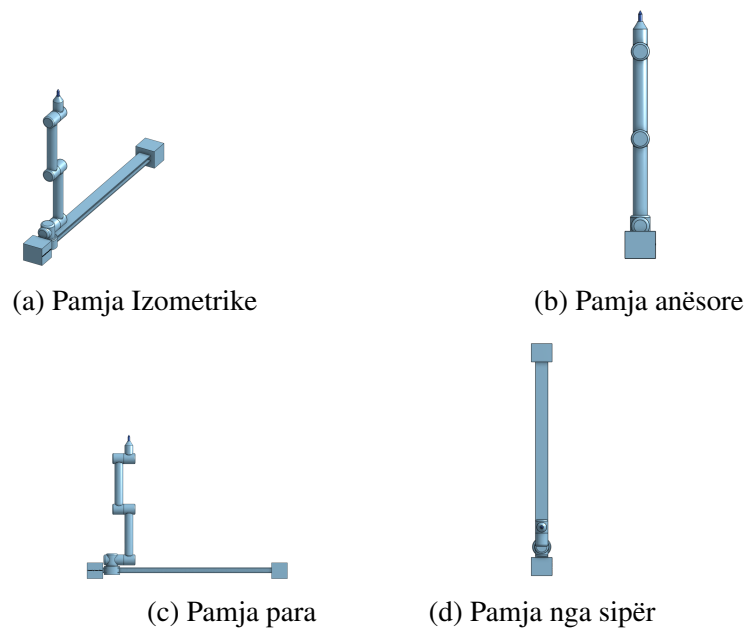


Figura 2.2: Montimi i krahut robotik

Pas përfundimit të fazës së projektimit të krahut robotik, hapi tjetër është importimi i modelit 3D CAD të krahut në MATLAB dhe Simscape Multibody, i cili mund të realizohet duke përdorur kodin e mëposhtëm [4]:

```

1 % linku i komponentes se montimit (Assembly object) ne
   Onshape
2 writing_robot_3D_model_url='https://cad.onshape.com/documents
   /bfc141b5563edffb66162f3/w/27626b9b2e22ca301d597735/e/9
   f8e18de84c3619a9ba321f2';
3
4 writing_robot_3D_xml_file=smexportonshape(
   writing_robot_3D_model_url); % Download-imi i CAD modelit
   te krahut robotik
5 smimport(writing_robot_3D_xml_file); % Importimi i krahut
   robotik ne Simulink/Simscape Multibody

```

Kodi i implementuar lehtëson gjenerimin e skedarëve të paraqitur në Figurën 2.3. Këta skedarë përfshijnë elementë thelbësorë të të dhënave, duke përfshirë skedarët e lidhjes së formës (modelit) gjeometrik në formatin .step, detajet e pozicionimit, vetitë e masës, koordinatat e qendrës së masës, specifikimet e orientimit të ruajtura në një skedar .xml dhe përfaqësimin e sistemit të krahut robotik brenda Simulink.



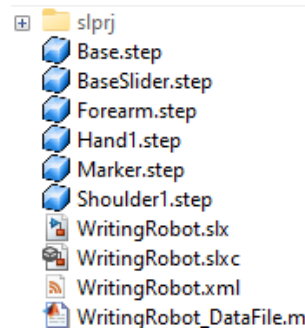


Figura 2.3: Skedarët e gjeneruar nga kodi për importim të krahut robotik nga Onshape

Skedarët .xml dhe .step përdoren nga skedari .slx i cili përdoret nga Simulink për të gjeneruar modelin e krahut robotik dhe mundësuar simulimin e tij përmes sistemit të krijuar (shiko Figurën 2.4).

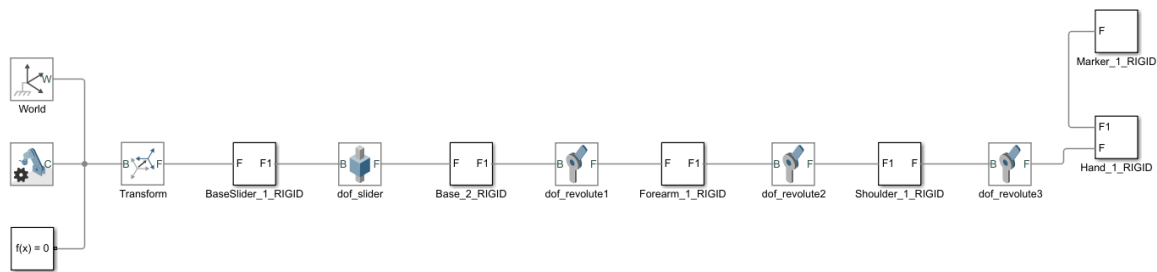


Figura 2.4: Sistemi i gjeneruar nga importi i krahut robotik nga Onshape

Meqenëse importimi dhe shndërrimi i nyjeve shfaq papërsosmëri, veçanërisht në pasqyrimin e saktë të nyjeve të fiksuara si ai midis dorës dhe veglës, ndërhyrja manuale bëhet e nevojshme. Kjo përfshin lidhjen e përpiktë të komponentëve të lidhur për të siguruar konfigurimin e dëshiruar të bashkimit fiks duke përdorur komponenten Simscape weld joint, siç ilustron në Figurën 2.5. Pas konfigurimit të saktë për të shikuar se a është importuar cdo gjë në rregull

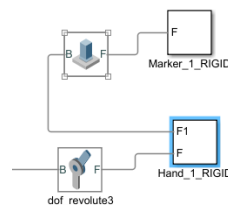


Figura 2.5: Nyja që bashkon dorën me veglën

mund ta startojmë aplikacionin ku Simulink shfaq simulimin përmes faqes Mechanics Explorer e cila gjendet në veglën Simscape Multibody siç shihet në Figurën 2.6.

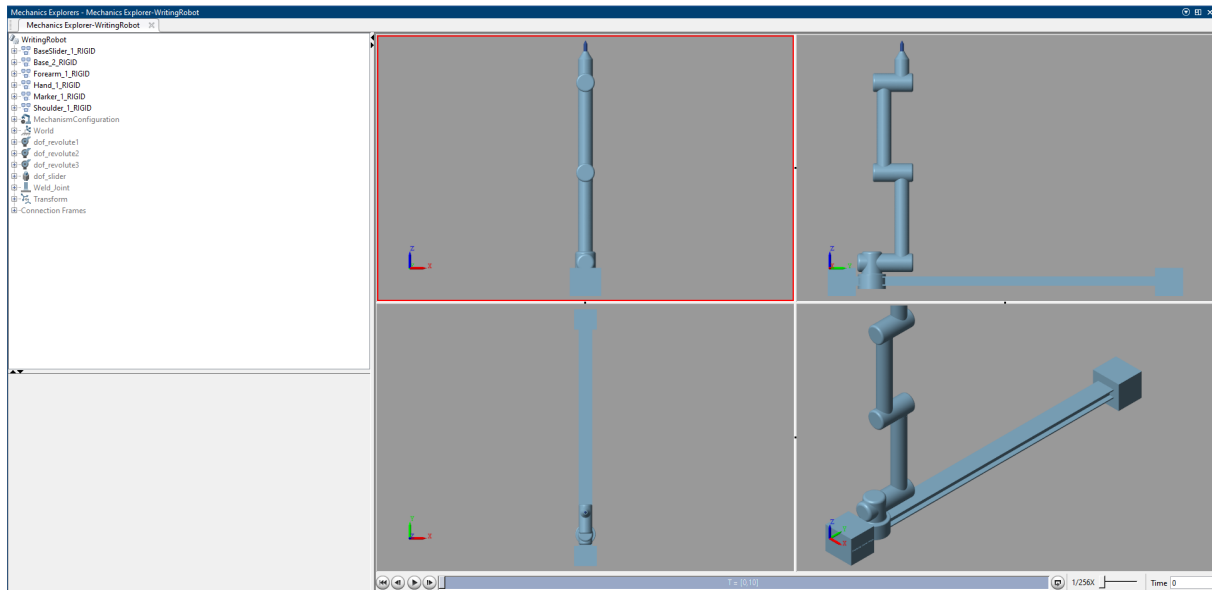


Figura 2.6: Simulimi i krahut robotik i shfaqur në Mechanics Explorer

## 2.2 Kinematika

Në fushën e manipuluesve të robotëve, shfaqen dy klasifikime kryesore: manipuluesit serik, të ndërtuar duke përdorur një zinxhir kinematik me unazë të hapur (open loop) dhe manipuluesit paralelë [5], të krijuar përmes zinxhirëve kinematikë me unazë të mbyllur (closed loop). Ky studim fokusohet ekskluzivisht tek manipuluesit serik.

Një manipulues serik përbëhet nga një seri  $n$ -lidhjesh të ndërlidhura me nyje [6]. Këto lidhje dhe nyje përbëjnë së bashku një zinxhir kinematik, një skaj i të cilit është ngjitur në një tokë të fik-suar ndërsa skaji tjetër mbetet i pakufizuar. Në skajin fundor të anës së palidhur ndodhet efektori fundor (end-effector), i shënuar si një shënues (marker) brenda modelit tonë të krahut robotik. Nomenklatura e një manipuluesi robot zakonisht korrespondon me shkallën e tij të lirisë (DOF), që përfaqëson numrin e nyjeve të tij. Për shembull, një robot manipulues i karakterizuar nga katër nyje ka 4 DOF, e kështu me radhë. Kinematika është gjeometria e lëvizjes së robotit ma-nipulues nga pozicioni referent në pozicionin e dëshiruar pa marrë parasysh forcat ose faktorët e tjerë që ndikojnë tek lëvizja e robotit [7]. E thënë thjesht, kinematika i përket lëvizjes së një ma-nipuluesi robot në lidhje me një kornizë të palëvizshme me kalimin e kohës. Në kontekstin e robotikës, kjo kornizë e palëvizshme nënkupton bazën, duke shërbyer si pikë e origjinës për të gjitha lëvizjet e mëvonshme. Kinematika qëndron si një disiplinë kryesore në robotikë, duke ofruar metodologji për përcaktimin e konfigurimit dhe dinamikës së sistemeve të manipuluesve të robotëve. Rëndësia e saj shtrihet në fusha praktike si planifikimi dhe kontrolli i trajektores, ku luan një rol kyç në lehtësimin e strategjive operacionale efektive.

Zgjidhja kinematike në këtë seksion do të fokusohet në dy probleme të rëndësishme që lindin në manipulatorët e robotëve, pra do të diskutokmë për zgjidhjen e kinematikës direkte (forward kinematics) dhe kinematikën inverse (inverse kinematics).

### 2.2.1 Kinematika Direkte

Ekuacionet kinematike direkte përcaktojnë korrelacionin midis variablave të përbashkëta dhe attributeve pozicionale si dhe orientuese të end efektorit. Duke supozuar se roboti përbëhet nga  $i$ -lidhje, me nyje dhe lidhje të numëruara në mënyrë sekuenciale nga 1 në  $i$  dhe 0 në  $i$ , re-

spektivisht, variablat e nyjes përfaqësohen si  $q_i$  [6]. Për nyjet prizmatike,  $q_i$  nënkupton zhvendosjen, ndërsa për nyjet rrotulluese,  $q_i$ , tregon këndin e rrotullimit [6].

Për të marrë ekuacionet e kinematikës direkte për krahun robotik (apo robotët në përgjithësi) duhet të kryhen hapat e mëposhtëm:

- **Marrim parametrat e DH:** Për të karakterizuar kinematikën e një roboti, çdo lidhje i caktohen katër parametra:  $\theta_i$ ,  $\alpha_i$ ,  $a_i$  dhe  $d_i$  [6]. Midis tyre, dy parametra përcaktojnë vetitë e vetë lidhjes, ndërsa dy të tjerët përshkruajnë lidhjen e saj me lidhjet ngjitur [6]. Në kontekstin e robotëve rrotullues dhe prizmatikë, variablat  $\theta_i$  dhe  $d_i$  [6]. identifikohen si variabla të përbashkëta. Parametrat Denavit-Hartenberg (DH) mund të llogariten manualisht ose me ndihmën e programeve kompjuterike si MATLAB. Duke u bazuar tek gjenerimi i struktuës së kordinatave të nyjeve (joint frames) është nxjerr tabela e Denavit-hartenberg për krahun robotik. Tek Figura 2.7 dhe Figura 2.8 është paraqitur struktura e thjeshtuar ku kornizat nuk kanë zhvendosje të pa përfillshme në boshtin Y:

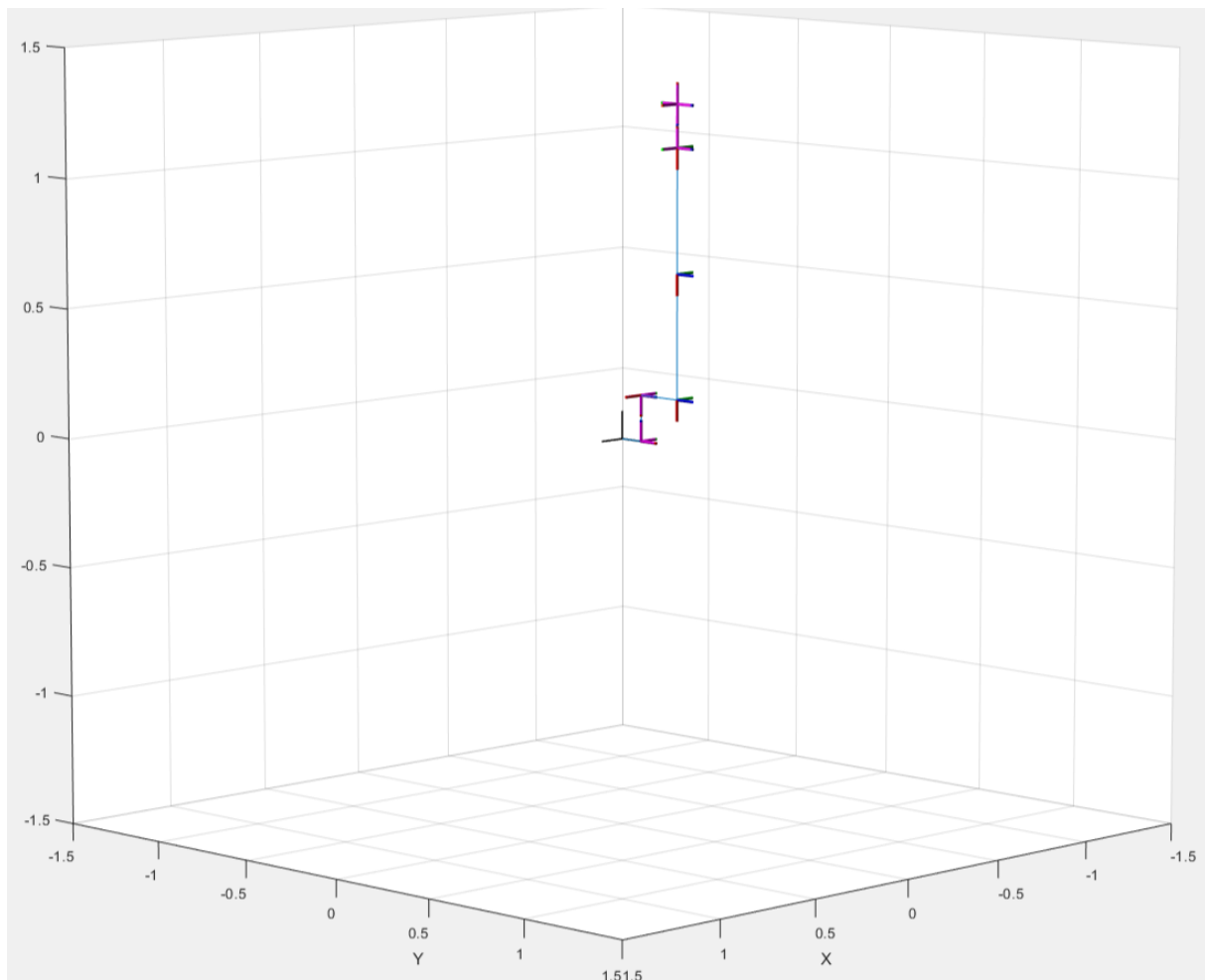


Figura 2.7: Struktura e thjeshtuar e paraqitur përmes Matlab

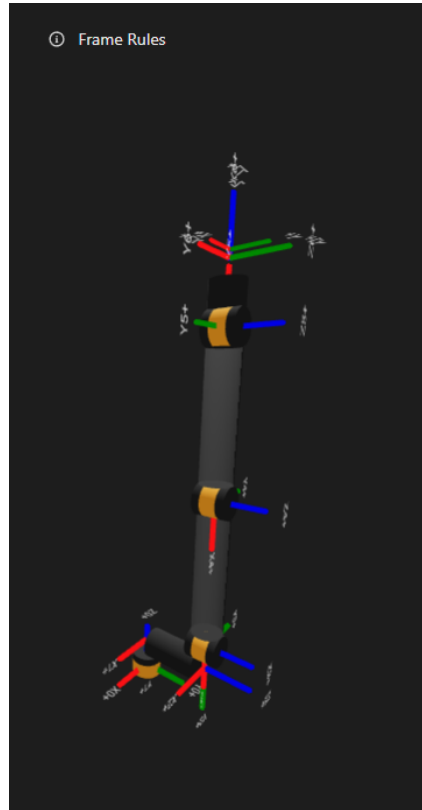


Figura 2.8: Paraqitja e krahut robotik me linqe,nyje dhe frames

Prandaj nga struktura do të fitojmë tabelën:

linku	$\theta_i$	$\alpha_i$	$a_i$	$d_i$
1	0	0	0	$0.1905+d_1$
2	$\theta_1 + \frac{\pi}{2}$	0	-0.48	0
3	$\theta_2$	0	-0.48	0
4	$\theta_3 + \pi$	0	0.206	0

Tabela 2.1: Representimi i DH parametrave

- **Gjejmë matricën e transformimeve  $T_{tool}^{ref}$ :** Pasi të jetë krijuar sistemi i koordinatave D-H për secilën lidhje, një matricë homogjene e transformimit mund të zhvillohet lehtësisht duke lidhur koordinatën  $i$ -të kornizë në kornizën e koordinatave  $(i - 1)$ [6]. Duke kryer transformimet vijuese të njëpasnjëshme është e qartë se pika  $r_i$  e shprehur në sistemin e  $i$  koordinatave mund të shprehet në  $(i - 1)$  të sistemit të koordinatave si  $r_{i-1}$  [6]:

1. Rrotulloni rreth boshtit  $z_{i-1}$  një kënd prej  $\theta_i$  për të rreshtuar boshtin  $x_{i-1}$  me boshtin  $x_i$  (boshti  $x_{i-1}$  është paralel me  $x_i$  dhe i drejtuar në të njëjtin drejtim) [6].
2. Lëvizni përgjatë boshtit  $z_{i-1}$  një distancë prej  $d_i$  për të sjellë boshtet  $x_{i-1}$  dhe  $x_i$  të puthiten (coincidence) [6].
3. Lëvizni përgjatë boshtit  $x_i$  një distancë të  $a_i$  për të sjellë edhe dy origjinat si dhe boshtin  $x$  të puthiten (coincidence) [6].
4. Rrotulloni rreth boshtit  $x_i$  një kënd prej  $\alpha_i$  për të vendosur dy sistemet koordinative të puthiten (coincidence) [6].

Pas këtyre transformimeve do të fitojmë të ashtuquajturën matrica e transformimeve D-H e cila shprehet si vijon:

$$T_i^{i-1} = Trans_{z_{i-1}}(d_i) \cdot Rot_{z_{i-1}}(\theta_i) \cdot Trans_{x_i}(r_i) \cdot Rot_{x_i}(\alpha_i)$$

$$T_i^{i-1} = \left[ \begin{array}{ccc|c} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & r_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & r_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ \hline 0 & 0 & 0 & 1 \end{array} \right] = \left[ \begin{array}{ccc|c} & & & \\ & R & & T \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$$

Duke pasur parasysh ekuacionin e mësipërme dhe duke përdorur një algoritëm në Matlab [8], matricën e transformimeve  $T_{tool}^{ref}$  e zgjidhim pasi modifikojm kodin si vijon:

```

1 function [DH_HTM] = DH_HTM(Matrix, angtype)
2 % Input Matrix: DH Table of (n,4) Dimension, else throw
  error
3 % Output matrix: Homogenous transformation: Dimension
  (4,4)
4
5 if size(Matrix,2) != 4
6     error("Matrix must have 4 columns");
7 end
8
9 output = eye(4);
10
11 len = size(Matrix,1); % Number of Rows
12 transformation_matrices=cell(1, len); % krijome matrica
  per secilin rresht ne tabelen e parametrave DH
13
14 for i = 1 : len
15     params = Matrix(i,:);
16     theta = params(1);
17     alpha = params(2);
18     rx = params(3);
19     dz = params(4);
20     next = dh_link(theta, alpha, rx, dz, angtype);
21     output = output * next;
22     transformation_matrices{i}=next;
23     simplify(transformation_matrices{i});
24     % Print the matrix with a formatted string
25     fprintf('%s =\n', sprintf('transformation_%d', i)); %
      Emerojme matricen transformuese me rreshtin
      perkates
26     disp(transformation_matrices{i}); % Shfaqim secilen
      matrice si te vecante
27 end
28
29 output = simplify(output);

```

```

30 DH_HTM = output;
31 end

```

Skedari `dh_link.m` mbetet i paprekur, pasi strehon ekskluzivisht matricën e përgjithshme të Denavit-Hartenberg.

```

1 function [matrix] = dh_link(theta,alpha,rx,dz,angle)
2
3 if angle == "d"
4     matrix = [
5         cosd(theta) -sind(theta)*cosd(alpha) sind(theta)*
6             sind(alpha) rx*cosd(theta);...
7         sind(theta) cosd(theta)*cosd(alpha) -cosd(theta)*
8             sind(alpha) rx*sind(theta);...
9         0 sind(alpha) cosd(alpha) dz;...
10        0 0 0 1 ...
11    ];
12 end
13 if angle == "r"
14     matrix = [
15         cos(theta) -sin(theta)*cos(alpha) sin(theta)*sin(
16             alpha) rx*cos(theta);...
17         sin(theta) cos(theta)*cos(alpha) -cos(theta)*sin(
18             alpha) rx*sin(theta);...
19         0 sin(alpha) cos(alpha) dz;...
20        0 0 0 1 ...
21    ];
22 end
23 matrix = vpa(matrix);
24 end

```

Duke përdorur funksionet e mësipërme dhe tabelën e paramterave DH mund të gjejmë transformimet e krahut robotik në mënyrë automatike. Kodi i cili duhet ekzekutuar është:

```

1 syms d1 theta1 theta2 theta3
2
3 % Matrica e transformimit te bazes me nyjen e pare
4   prizmatike
5 B_matrix = [pi/2 0 0.101 0]
6 B_DH_matrix = DH_HTM(B_matrix, 'r')
7
8 % Konfiguracioni i end efektorit nga korniza e fundit e
9   nyjes rrotulluese deri ne majen e shenuesit
10 H_matrix = [ pi/2 -pi/2 0 0]
11 H_DH_matrix = DH_HTM(H_matrix, 'r')
12
13 % Whole Matrix parameter table from Base to End effector
14 DH_final=[
15     pi/2 0 0.101 0;
16     -pi/2 -pi/2 0 0.1775;

```

```

15     0 0 0 0.1905+d1;
16     theta1+(pi/2) 0 -0.48 0;
17     theta2 0 -0.48 0;
18     theta3+pi 0 0.206 0;
19     pi/2 -pi/2 0 0;
20 ];
21
22 % The whole Calculation of forward kinematics from base
    to end-effector
23 solved_DH_Matrix_final = DH_HTM(DH_final, 'r')
24
25 px = solved_DH_Matrix_final(1,4)
26 py = solved_DH_Matrix_final(2,4)
27 pz = solved_DH_Matrix_final(3,4)

```

Pas ekzekutimit të kodit të mësipërm janë fituar rezultatet si më poshtë:

```

1 transformation_1 =
2 [ 0, -1.0, 0, 0]
3 [1.0, 0, 0, 0.101]
4 [ 0, 0, 1.0, 0]
5 [ 0, 0, 0, 1.0]
6
7 transformation_2 =
8 [ 0, 0, 1.0, 0]
9 [-1.0, 0, 0, 0]
10 [ 0, -1.0, 0, 0.1775]
11 [ 0, 0, 0, 1.0]
12
13 transformation_3 =
14 [1.0, 0, 0, 0]
15 [ 0, 1.0, 0, 0]
16 [ 0, 0, 1.0, d1 + 0.1905]
17 [ 0, 0, 0, 1.0]
18
19 transformation_4 =
20 [cos(theta1 + 1.5707963267948966192313216916398), -1.0*
    sin(theta1 + 1.5707963267948966192313216916398), 0,
    -0.48*cos(theta1 + 1.5707963267948966192313216916398)]
21 [sin(theta1 + 1.5707963267948966192313216916398),
    cos(theta1 + 1.5707963267948966192313216916398), 0,
    -0.48*sin(theta1 + 1.5707963267948966192313216916398)]
22 [ 0, 0, 1.0, 0]
23 [ 0, 0, 0, 0]

```

```

24                                                                 1.0]
25 transformation_5 =
26 [cos(theta2), -1.0*sin(theta2),    0, -0.48*cos(theta2)]
27 [sin(theta2),    cos(theta2),    0, -0.48*sin(theta2)]
28 [          0,          0, 1.0,          0]
29 [          0,          0,    0,          1.0]
30
31 transformation_6 =
32 [-1.0*cos(theta3),    sin(theta3),    0, -0.206*cos(
33   theta3)]
34 [-1.0*sin(theta3), -1.0*cos(theta3),    0, -0.206*sin(
35   theta3)]
36 [          0,          0, 1.0,
37   0]
38 [          0,          0,    0,
39   1.0]
40
41 transformation_7 =
42 [ 0,    0, -1.0,    0]
43 [1.0,    0,    0,    0]
44 [ 0, -1.0,    0,    0]
45 [ 0,    0,    0, 1.0]
46
47 solved_DH_Matrix_final =
48 [sin(theta1 + theta2 + theta3 +
49   1.5707963267948966192313216916398),    0,    cos(
50   theta1 + theta2 + theta3 +
51   1.5707963267948966192313216916398),    - 0.48*
52   cos(theta1 + theta2 +
53   1.5707963267948966192313216916398) - 0.206*cos(theta1
54   + theta2 + theta3 + 1.5707963267948966192313216916398)
55   - 0.48*cos(theta1 +
56   1.5707963267948966192313216916398)]
57 [
58
59   0, -1.0,
60
61   0,
62
63   d1 + 0.2915]
64 [cos(theta1 + theta2 + theta3 +
65   1.5707963267948966192313216916398),    0, -1.0*sin(
66   theta1 + theta2 + theta3 +
67   1.5707963267948966192313216916398), 0.48*sin(theta1 +
68   theta2 + 1.5707963267948966192313216916398) + 0.206*

```



```

49   sin(theta1 + theta2 + theta3 +
    1.5707963267948966192313216916398) + 0.48*cos(theta1 -
    1.4420985131352750410631430775095e-33) + 0.1775]
50   [
51     0,      0,
52     0,
53     1.0]
54   px =
55   - 0.48*cos(theta1 + theta2 +
    1.5707963267948966192313216916398) - 0.206*cos(theta1
    + theta2 + theta3 + 1.5707963267948966192313216916398)
    - 0.48*cos(theta1 +
    1.5707963267948966192313216916398)
56
57   py =
58   d1 + 0.2915
59
60
61
62   pz =
63
64   0.48*sin(theta1 + theta2 +
    1.5707963267948966192313216916398) + 0.206*sin(theta1
    + theta2 + theta3 + 1.5707963267948966192313216916398)
    + 0.48*cos(theta1 - 1.4420985131352750410631430775095
    e-33) + 0.1775

```

Ku numri 1.5707963267948966192313216916398 është numri  $\frac{\pi}{2}$  i shprehur si numër decimal.

Dy rreshtat e parë të matricës janë korniza të fiksuara të cilat shprehin rrotullimet e nevojshme për të shprehur ndërlidhjen e kornizës kryesore (world frame) me nyjen prizmatike, prandaj dy nyjet e para japin matricën B:

Link	$\theta_i$	$\alpha_i$	$a_i$	$d_i$
1	$\frac{\pi}{2}$	0	0.101	0
2	$-\frac{\pi}{2}$	$-\frac{\pi}{2}$	0	0.1775

Tabela 2.2: Parametrat me të cilët krijohet matrica B

$$B = T_{world}^0 = B_1 B_2 \quad (2.1)$$

$$B_1 = \begin{bmatrix} 0 & -1.0 & 0 & 0 \\ 1.0 & 0 & 0 & 0.101 \\ 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 1.0 \end{bmatrix} \quad (2.2)$$

$$B_2 = \begin{bmatrix} 0 & 0 & 1.0 & 0 \\ -1.0 & 0 & 0 & 0 \\ 0 & -1.0 & 0 & 0.1775 \\ 0 & 0 & 0 & 1.0 \end{bmatrix} \quad (2.3)$$

Kurse linqet e krahut robotik janë:

Link	$\theta_i$	$\alpha_i$	$a_i$	$d_i$
1	0	0	0	$0.1905+d_1$
2	$\theta_1 + \frac{\pi}{2}$	0	-0.48	0
3	$\theta_2$	0	-0.48	0
4	$\theta_3 + \pi$	0	0.206	0

Tabela 2.3: Parametrat me të cilët krijohet matrica e transformimeve të krahut robotik

$$T_1^0 = \begin{bmatrix} 1.0 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 \\ 0 & 0 & 1.0 & d_1 + 0.1905 \\ 0 & 0 & 0 & 1.0 \end{bmatrix} \quad (2.4)$$

$$T_2^1 = \begin{bmatrix} \cos(\theta_1 + \frac{\pi}{2}) & -1.0 \sin(\theta_1 + \frac{\pi}{2}) & 0 & -0.48 \cos(\theta_1 + \frac{\pi}{2}) \\ \sin(\theta_1 + \frac{\pi}{2}) & \cos(\theta_1 + \frac{\pi}{2}) & 0 & -0.48 \sin(\theta_1 + \frac{\pi}{2}) \\ 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 1.0 \end{bmatrix} \quad (2.5)$$

$$T_3^2 = \begin{bmatrix} \cos(\theta_2) & -1.0 \sin(\theta_2) & 0 & -0.48 \cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & -0.48 \sin(\theta_2) \\ 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 1.0 \end{bmatrix} \quad (2.6)$$

$$T_4^3 = \begin{bmatrix} -1.0 \cos(\theta_3) & \sin(\theta_3) & 0 & -0.206 \cos(\theta_3) \\ -1.0 \sin(\theta_3) & -1.0 \cos(\theta_3) & 0 & -0.206 \sin(\theta_3) \\ 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 1.0 \end{bmatrix} \quad (2.7)$$

Tani, matrica H përmbledh matricat e transformimit nga nyja përfundimtare rrotulluese deri në majën e mjetit (në këtë rast, shënuesin). Në thelb, ne kemi nevojë vetëm për rrotullimin e kornizës, duke pasur parasysh që distanca nga nyja e fundit rrotulluese deri në majën e mjetit është e integruar në matricën e transformimit të bashkimit përfundimtar rrotullues.

Link	$\theta_i$	$\alpha_i$	$a_i$	$d_i$
1	$\frac{\pi}{2}$	$-\frac{\pi}{2}$	0	0

Tabela 2.4: Parametrat me të cilët krijohet matrica H

$$H = T_4^{tool} = \begin{bmatrix} 0 & 0 & -1.0 & 0 \\ 1.0 & 0 & 0 & 0 \\ 0 & -1.0 & 0 & 0 \\ 0 & 0 & 0 & 1.0 \end{bmatrix} \quad (2.8)$$

Pra, matrica e plotë e transformimit pas produktit të:  $T_{world}^{tool} = BT_4^0 H$

$$T_{world}^{tool} = \begin{bmatrix} \sin(\theta_1 + \theta_2 + \theta_3 + \frac{\pi}{2}) & 0 & \cos(\theta_1 + \theta_2 + \theta_3 + \frac{\pi}{2}) & 0 \\ 0 & -1.0 & 0 & 0 \\ \cos(\theta_1 + \theta_2 + \theta_3 + \frac{\pi}{2}) & 0 & -1.0\sin(\theta_1 + \theta_2 + \theta_3 + \frac{\pi}{2}) & 0 \\ 0 & 0 & 0 & 0 \\ -0.48\cos(\theta_1 + \theta_2 + \frac{\pi}{2}) - 0.206\cos(\theta_1 + \theta_2 + \theta_3 + \frac{\pi}{2}) - 0.48\cos(\theta_1 + \frac{\pi}{2}) & d_1 + 0.2915 & 0 & 0 \\ 0.48\sin(\theta_1 + \theta_2 + \frac{\pi}{2}) + 0.206\sin(\theta_1 + \theta_2 + \theta_3 + \frac{\pi}{2}) + 0.48\cos(\theta_1 - 1.4421) + 0.1775 & 0 & 1.0 & 0 \end{bmatrix}$$

Pas zgjidhjes së ekuacioneve të kinematikës direkte pra thënë ndryshe zgjidhjen e matricave të transformimeve mund të krijojmë një bllok funksion i cili reprezenton kinematikën direkte (shiko Figurën 2.9)

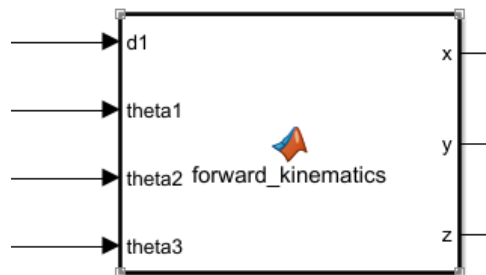


Figura 2.9: Bllok funksioni i cili përmban kinematikën direkte

Për arsye se me interes kemi vetëm pozicionin atëherë funksioni do të marr parasysh vetëm kolonën e fundit që përmban pozitën dhe shkallëzimin.

```

1 function [x, y, z] = forward_kinematics(d1, theta1, theta2,
2   theta3)
3 x = - 0.48*cos(theta1 + theta2 +
4   1.5707963267948966192313216916398) - 0.206*cos(theta1 +
5   theta2 + theta3 + 1.5707963267948966192313216916398) -
   0.48*cos(theta1 + 1.5707963267948966192313216916398); #
   0.206 sepse 0.206 + 0.04 (gjatesia e lapsit)
3 y = d1 + 0.2915;
4 z = 0.48*sin(theta1 + theta2 +
   1.5707963267948966192313216916398) + 0.206*sin(theta1 +
   theta2 + theta3 + 1.5707963267948966192313216916398) +
   0.48*cos(theta1 - 1.4420985131352750410631430775095e-33) +
   0.1775;
5 end

```

Po të nevojitej edhe orientimi i lapsit atëherë funksioni do të kishte marrë formën si më poshtë ku do të kthente matricë të transformimit në vend të koordinatave x, y, z.

```

1 function T_matrix = forward_kinematics(d1, theta1, theta2,
2   theta3)
3 T_matrix=[sin(theta1 + theta2 + theta3 +
4   1.5707963267948966192313216916398),    0,    cos(theta1
5   + theta2 + theta3 + 1.5707963267948966192313216916398), ...
6   - 0.48*cos(theta1 + theta2 +
7   1.5707963267948966192313216916398) - 0.206*cos(theta1
8   + theta2 + theta3 + 1.5707963267948966192313216916398)
9   - 0.48*cos(theta1 +
10  1.5707963267948966192313216916398);
11   0, -1.0, 0, d1 + 0.2915;
12 cos(theta1 + theta2 + theta3 +
13  1.5707963267948966192313216916398),    0, -1.0*sin(theta1
14  + theta2 + theta3 + 1.5707963267948966192313216916398), ...
15  0.48*sin(theta1 + theta2 + 1.5707963267948966192313216916398)
16  + 0.206*sin(theta1 + theta2 + theta3 +
17  1.5707963267948966192313216916398) + 0.48*cos(theta1 -
18  1.4420985131352750410631430775095e-33) + 0.1775;
19   0, 0, 0, 1.0];
20 end

```

### 2.2.2 Kinematika Inverse

Ky seksion është i përkushtuar për të trajtuar problemin e Kinematikës Inverse (IK), i cili ka të bëjë me përcaktimin e variablave të përbashkët të një manipuluesi robotik në përgjigje të një pozicioni dhe orientimi të përkohshëm të end efektorit (shënuesit). Edhe pse problemi i përgjithshëm i kinematikës inverse është mjaft i vështirë, rezulton se krah robotik i shtjelluar në këtë tezë mund të ndahet (decouple) në dy pjesë: pjesa prizmatike e cila rezulton në zhvendosje në boshtin y dhe krahun robotik planar me tri nyje rrotulluese (3R Planar manipulator). Pjesa prizmatike mund të gjendet me lehtësi nga matrica transformuese e fituar tek seksioni i kinematikës direkte pra elementi  $a_{24}$  pra:

$$y = d_1 + 0.2915$$

$$d_1 = y - 0.2915 \quad (2.9)$$

Tani zgjidhim pjesën e krahut robotik planar me tri nyje rrotulluese e cila gjendet në pozicion fillestar vertikal (shih Figurën 2.10):



Figura 2.10: Krah robotik planar me tri nyje rrotulluese

Së pari vlen të cekim identitetet trigonometrike të cilat do të përdoren për zgjidhjen gjeometrike të krahut robotik me tri nyje rrotulluese:

- Teorema e kosinusit:  $a^2 = b^2 + c^2 - 2bccos(\alpha)$
- Formulatat e adicionit:  
 $sin(\theta_1 \pm \theta_2) = s_{12} = cos(\theta_1)sin(\theta_2) \pm sin(\theta_1)cos(\theta_2)$   
 $cos(\theta_1 \pm \theta_2) = c_{12} = cos(\theta_1)cos(\theta_2) \mp sin(\theta_1)sin(\theta_2)$

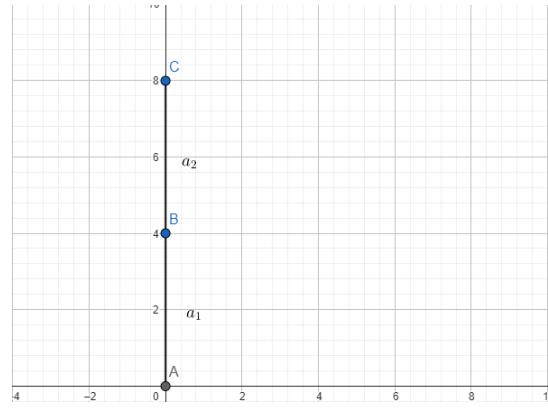
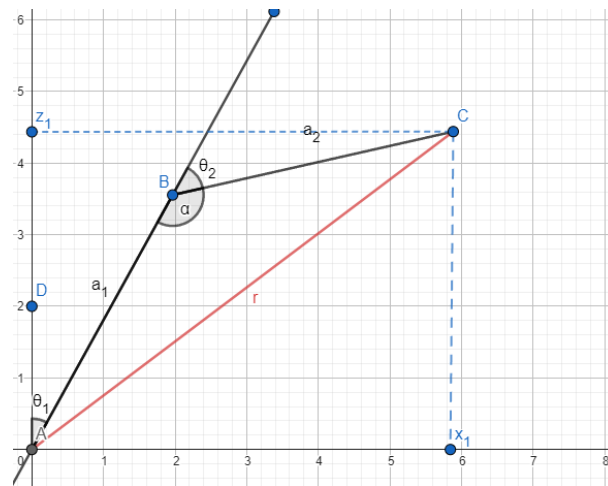


Figura 2.11: Krahu robotik planar me dy nyje rrotulluese

Për ta thjeshtuar edhe më shumë zgjidhjen e problemit ne do të zgjidhim kinematikën inverse vetëm për dy nyjet e para:

$$r^2 = a_1^2 + a_2^2 - 2a_1a_2\cos(\alpha) \quad (2.10)$$

Ku  $a_1 = a_2$  dhe  $r = x_1^2 + z_1^2$

Figura 2.12: Zbatimi i teoremës së kosinuset për  $\theta_2$ 

$$\cos(\alpha) = \frac{a_1^2 + a_2^2 - r^2}{2a_1a_2} = \frac{a_1^2 + a_2^2 - (x_1^2 + z_1^2)}{2a_1a_2} \quad (2.11)$$

$$\theta_2 = \pi - \alpha \Rightarrow \cos(\theta_2) = \cos(\pi - \alpha) = -\cos(\alpha) \quad (2.12)$$

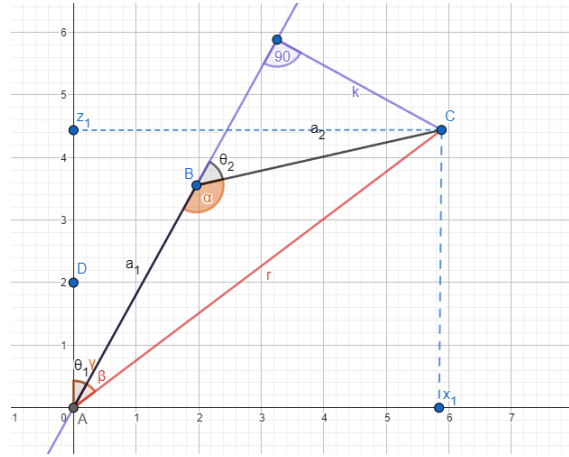
$$\cos(\theta_2) = \frac{(x_1^2 + z_1^2) - a_1^2 - a_2^2}{2a_1a_2} \quad (2.13)$$

$$\theta_2 = \arccos\left(\frac{(x_1^2 + z_1^2) - a_1^2 - a_2^2}{2a_1a_2}\right) \quad (2.14)$$

Pas zgjidhjes për  $\theta_2$  gjejmë  $\theta_1$  nga Figura 2.13:

$$\tan(\beta) = \frac{a_2\sin(\theta_2)}{a_1 + a_2\cos(\theta_2)} \quad (2.15)$$

$$\tan(\gamma) = \frac{x_1}{z_1} \quad (2.16)$$

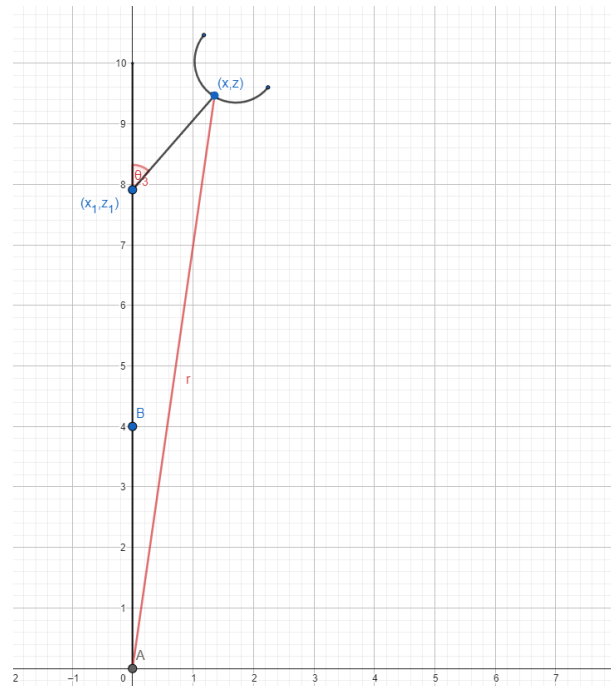
Figura 2.13: Zgjidhja e  $\theta_1$  përmes tangjenteve të këndit  $\gamma$  dhe  $\beta$ 

Është lehtësisht e dukshme që parametri  $\theta_1$  mund të nxirret nga diferenca ose nga shuma e ekuacioneve (2.15) dhe (2.16).

$$\theta_1 = \gamma - \beta = \arctan\left(\frac{x_1}{z_1}\right) - \arctan\left(\frac{a_2 \sin(\theta_2)}{a_1 + a_2 \cos(\theta_2)}\right) \quad (2.17)$$

$$\theta_1 = \gamma + \beta = \arctan\left(\frac{x_1}{z_1}\right) + \arctan\left(\frac{a_2 \sin(\theta_2)}{a_1 + a_2 \cos(\theta_2)}\right) \quad (2.18)$$

Se cilën do ta përdorim varet nga parashtrimi i modelit pasi që ekuacionet e më sipërme japin zgjidhje të njëjtë (shiko seksionin Zgjidhjet e mundshme (ekzistenca) dhe uniciteti) për pozitë të njëjtë, prandaj duke pasur parasysh se krahu robotik do të shkruaj në pozicion vertikal është vendosur të mirret ekuacioni (2.17) i cili merr formën e bërrylit nga sipër pra formimi i këndit të ngushtë mes dy linqeve formohet nga pjesa e poshtme e krahur robotik.

Figura 2.14: Zgjidhja e  $\theta_3$  në mënyrë gjeometrike

Tani duke ditur koordinatat  $x_1$  dhe  $z_1$  dhe pozitën e dëshiruar  $(x, z)$  mund të caktojmë vlerën e  $\theta_3$ .

$$\begin{cases} x_1 = x - a_3 \sin(\phi) \\ z_1 = z - a_3 \cos(\phi) \end{cases} \quad (2.19)$$

Ku  $\phi = \theta_1 + \theta_2 + \theta_3$ , pra rrjedh se  $\theta_3$  është:

$$\theta_3 = \phi - (\theta_1 + \theta_2) \quad (2.20)$$

Tani krejt çfarë na ka mbetur është të krijojmë bllok funksionin në Simulink që përmban kodin:

```

1 function [d1,theta1,theta2,theta3] = inverse_kinematics(x,y,z
2 )
3 d1=y-0.2915;
4 theta_total = pi/2;
5 x_1 = (x)-0.206*sin(theta_total);
6 z_1 = (z-0.1775)-0.206*cos(theta_total);
7 theta2 = acos(((x_1^2+z_1^2)-(2*0.48^2))/(2*0.48^2));
8 theta1 = atan(x_1/z_1) - atan((0.48*sin(theta2))/(0.48+(0.48*
9   cos(theta2))));
10 theta3 = theta_total - (theta1 + theta2);
11 end

```

Duke pasur parasysh se nyja e parë rrotulluese është e zhvendosur për 0.1775m nga origjina atëherë është e nevojshme koordinatës  $z$  ti ju zbritet kjo zhvendosje. Po ashtu pasi që shenuesi supozohet të jetë normal me rafshin ku shkuran (pra të ketë pozicion normal ndaj boshtit  $Z$  global) është e nevojshme që  $\phi$  të mirret si  $\frac{\pi}{2}$  apo nëse  $\phi = 0$  atëherë ekuacioni për zgjidhjen e  $\theta_3$  do të ndryshoj si vijon:

$$\theta_3 = \phi - (\theta_1 + \theta_2) + \frac{\pi}{2}$$

Kurse bllok funksioni i krijuar ka formën sikur në Figurën 2.15:

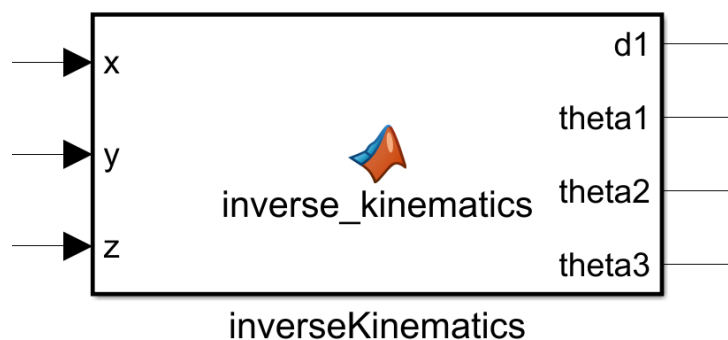


Figura 2.15: Bllok funksioni i kinematikës inverse

### Zgjidhjet e mundshme (ekzistenca) dhe uniciteti

Së pari shikojmë se a ekziston gjithmonë së paku një zgjidhje për  $\alpha = \arctan(\frac{\pm\sqrt{1-D^2}}{D})$ , atëherë do të nisemi nga premisa e kundërt se a plotësohet kushti kur asnjë zgjidhje s'ekziston



pra  $D^2 > 1$  [9]:

$$D^2 = \left( \frac{a_1^2 + a_2^2 - r^2}{2a_1a_2} \right)^2 \leq 1 \quad (2.21)$$

$$a_1^2 + a_2^2 - r^2 \leq \pm 2a_1a_2$$

$$a_1^2 \pm 2a_1a_2 + a_2^2 \leq r^2$$

$$(a_1 \pm a_2)^2 \leq r^2$$

$$|a_1 \pm a_2| \leq r \quad (2.22)$$

Nga inekuacioni (2.22) shihet qartë se një zgjidhje për  $\alpha$  gjithmonë ekziston dhe se krahut robotik gjithmonë do të ketë zgjidhje brenda hapësirës së punës së tij.

Kurse sa i përket unicitetit se a ekziston më shumë se një zgjidhje për pozitë të njëjtë të krahut robotik mund të vërehet dukshëm se zgjidhja për  $\alpha = \arctan(\frac{\pm\sqrt{1-D^2}}{D})$  nuk është unike duke parë se mund të zgjedhim dy mundësi për ta zgjedhur rrënjën katrore  $\sqrt{1-D^2}$  nga operatori  $\pm$  [9]. Në rastin e krahut robotik të shtjelluar në këtë tezë merret parasysh zgjidhja ku bërryli ka

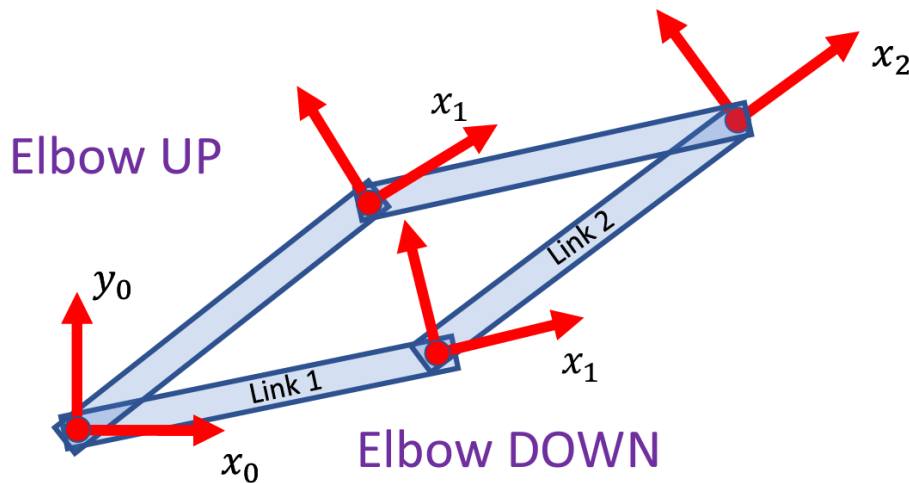


Figura 2.16: Zgjidhjet e mundëshme për koordinata fundore të njëjta [9]

pozitë të sipërme (elbow up), ku zgjidhja për këtë formacion të krahut robotik zgjidhet në ekuacionin (2.17), Pra, një mënyrë tjetër për konstatimin se a ekzitojnë më shumë se një zgjidhje për krahun robotik të dhënë mund të bëhet duke zgjidhur  $\theta_1$  për  $\pm\beta$  pra  $\theta_1 = \gamma \pm \beta$ .

**Redundanca**

Teprica (redundanca) kinematike ndodh kur një manipulues robotik ka më shumë shkallë lirie sesa ato që kërkohen rreptësisht për të ekzekutuar një detyrë të caktuar. Pra, për krahun me dy linqe [10], ne mund ta pozicionojmë origjinën e end-efektorit kudo në hapësirën e punës së krahut: dy hyrje ( $\theta_1, \theta_2$ ) dhe dy "dalje" ( $x, y$ ). Për krahun me tre linqe [10], ne mund ta pozicionojmë origjinën e end-efektorit kudo në hapësirën e punës të krahut dhe mund të zgjedhim orientimin e kornizës: tre hyrje ( $\theta_1, \theta_2, \theta_3$ ) dhe tre "dalje" ( $x, y, \phi$ ). Kurse sa i përket krahut robotik të shtjelluar në këtë tezë i cili i ka 4 shkallë lirie janë pafundësisht shumë mënyra për të arritur një konfigurim të dëshiruar të end-efektorit ( $x, y, \phi$ ), pra  $n > m$ .

Prandaj për  $n - m$  shkallë të lirisë të mbetura në një robot mund përshkrijnë dimensionet e shumëfishta vetëlëvizëse. Është e mundur të zgjedhësh nm nyjet e robotit, le të themi  $q_1, \dots, q_{n-m}$ , dhe rregulloni ato në vlera të caktuara dhe më pas zgjidhni një problem analitik IK për të përcaktuar zgjidhjen për  $m$  shkallë lirie të mbetura [10]. Vlerat  $q_1, \dots, q_{n-m}$  njihen si shkallë lirie të pavarura ndërsa pjesa tjetër njihen si shkallë lirie të varura [10].

Zgjedhja e shkallëve të lirisë të pavarura është disi arbitrare, por zakonisht zgjidhet më mirë për të çuar në zgjidhje të thjeshta analitike IK për shkallë lirie të varura [10]. Në çdo rast, nyjet e pavarura parametrizojnë shumëfishin e vetëlëvizjes [10]. Vështirësia kryesore në shprehjen e kësaj hapësire zgjidhjeje është përcaktimi i një domeni të përshtatshëm të shkallëve të lirisë të pavarura në mënyrë që të ekzistojë një zgjidhje analitike për shkallët e lirisë të varura [10].

**2.3 Dinamika**

Dinamika e një krahu robotik përshkruan se si lëvizja e krahut ndryshon në lidhje me kohën dhe forcat që veprojnë mbi të. Kuptimi i këtyre dinamikave është thelbësor për kontrollin e saktë të lëvizjes së krahut. Formulimi Lagrange-Euler është një metodë e zakonshme që përdoret për të nxjerrë ekuacionet e lëvizjes për manipuluesit robotikë [6].

**2.3.1 Formulimi Lagrange-Euler**

Për kalkulimin e dinamikës së krahut robotik të shtjelluar në këtë tezë kemi përdorur formulimin Lagrange-Euler i cili është një teknikë sistematike dhe e fuqishme për nxjerrjen e ekuacioneve të lëvizjes së një sistemi mekanik, siç është krahu i robotit [6]. Kjo metodë kombinon parimet e mekanikës së Lagranzhit dhe ekuacionet e lëvizjes së Euler-it [6].

**Mekanika Lagranzhiane**

Lagranzhiani  $L$  e një sistemi mekanik përcaktohet si diferenca midis energjisë kinetike  $T$  dhe energjia potenciale  $V$  i sistemit:

$$L = T - V$$

Ku:

- $T$  është energjia kinetike
- $V$  është energjia potenciale

Lagranzhiani  $L$  është një funksion i koordinatave të përgjithësuara  $q_i$  dhe deviatit të tyre i kohës  $\dot{q}_i$

$$L = L(q_1, q_2, \dots, q_n, \dot{q}_1, \dot{q}_2, \dots, \dot{q}_n, t)$$

Ku:

- $q_i$  janë koordinatat e përgjithësuara
- $\dot{q}_i$  janë derivatet kohore të koordinatave të përgjithësuara

Ekuacionet e lëvizjes mund të nxirren duke përdorur ekuacionin Euler-Lagrange:

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = \tau_i \quad i = 1, 2, \dots, n \quad (2.23)$$

### Ekuacionet e lëvizjes për një manipulator

Ekuacionet e energjisë kinetike dhe potenciale mund ta shkruajmë si:

$$T = \sum_{i=1}^n T_i = \frac{1}{2} \sum_{i=1}^n \sum_{p=1}^i \sum_{r=1}^i Tr(U_{ip} J_i U_{ir}^T) \dot{q}_p \dot{q}_r \quad (2.24)$$

$$V = \sum_{i=1}^n V_i = \sum_{i=1}^n V_i = \sum_{i=1}^n -m_i g ({}^0A_i^i r_i) \quad (2.25)$$

Kurse Lagranzhiani  $L = T - V$  nga ekuacionet e mësipërme do të jetë:

$$L = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^i \sum_{k=1}^i Tr(U_{ij} J_i U_{ik}^T) \dot{q}_j \dot{q}_k + \sum_{i=1}^n m_i g ({}^0A_i^i r_i) \quad (2.26)$$

Duke zëvendësuar (2.26) tek (2.23) do marrim formën matricore të ekuacioneve të lëvizjes:

$$\tau(t) = D(q(t)) \ddot{q}(t) + h(q(t), \dot{q}(t)) + c(q(t)) \quad (2.27)$$

Ku:

- $\tau(t) = n \times 1$  është vektori i momenteve të forcës për nyjet  $i = 1, 2, \dots, n$ :

$$\tau(t) = (\tau_1(t), \tau_2(t), \dots, \tau_n(t))^T$$

- $q(t) = n \times 1$  është vektori i ndryshoreve të nyjeve të krahut robotik:

$$q(t) = (q_1(t), q_2(t), \dots, q_n(t))^T$$

- $\dot{q}(t) = n \times 1$  është vektori i ndryshoreve të nyjeve të krahut robotik:

$$\dot{q}(t) = (\dot{q}_1(t), \dot{q}_2(t), \dots, \dot{q}_n(t))^T$$

- $\ddot{q}(t) = n \times 1$  është vektori i shpejtësive të nyjeve të krahut robotik:

$$\ddot{q}(t) = (\ddot{q}_1(t), \ddot{q}_2(t), \dots, \ddot{q}_n(t))^T$$

- $\ddot{q}(t) = n \times 1$  është vektori i nxitimit për ndryshoret  $q(t)$  të nyjeve të krahut robotik:

$$\ddot{q}(t) = (\ddot{q}_1(t), \ddot{q}_2(t), \dots, \ddot{q}_n(t))^T$$

- $D(q) = n \times n$  është matrica e inercioneve të lidhura me nxitimin elementet e të cilës janë:

$$D_{ik} = \sum_{j=\max(i,k)}^n Tr(U_{jk}J_jU_{ji}^T) \quad i, k = 1, 2, \dots, n$$

- $h(q, \dot{q}) = n \times 1$  është vektori i forcave jolineare të Coriolis-it dhe focave centrifugale elementet e të cilës janë:

$$h(q, \dot{q}) = (h_1, h_2, \dots, h_n)^T \quad (2.28)$$

Ku:

$$h_i = \sum_{k=1}^n \sum_{m=1}^n h_{ikm} \dot{q}_k \dot{q}_m \quad i = 1, 2, \dots, n \quad (2.29)$$

dhe

$$h_{ikm} = \sum_{j=\max(i,k,m)}^n Tr(U_{jkm}J_jU_{ji}^T) \quad i, k, m = 1, 2, \dots, n \quad (2.30)$$

- $c(q) = n \times 1$  është vektori i forcave të gravitetit mbi ngarkesën i cili shprehet me elementet:

$$c_i = \sum_{j=i}^n (-m_j g U_{ji}^j r_j) \quad i = 1, 2, \dots, n \quad (2.31)$$

### 2.3.2 Zgjidhja e problemit dinamik

Për krahun robotik në fjalë kemi ekuacionet e lëvizjes të shprehur si më poshtë, kur kemi parasysh se nyja e fundit ka masë aq të vogël në krahasim me dy nyjet tjera sa mund të mos përfillet:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{12} & D_{22} & D_{23} \\ D_{13} & D_{23} & D_{33} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} + \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} + \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} \quad (2.32)$$

Ku:

- Elementet e matricës së inercioneve të lidhura me nxitimin kanë vlerën (llogaritjet mund ti gjeni tek shtesa A.8):

$$D_{11} = m_1 + m_2 + m_3$$

$$D_{12} = -\frac{1}{2}l m_2 c_2 + c_1 l m_3 + \frac{3}{2}l m_3 c_{12}$$

$$D_{13} = \frac{3}{2}l m_3 c_{12}$$

$$D_{22} = \frac{1}{3}l^2 m_2 s_1^2 + \frac{1}{3}l^2 m_2 c_1^2 + \frac{1}{3}l^2 m_3 c_{12}^2 + l^2 m_3 s_1^2 + l^2 m_3 s_1 c_{12} + l^2 m_3 s_{12}^2 + l^2 m_3 s_{12} c_{12} \\ + 2l^2 m_3 s_1 s_{12} + l^2 m_3 c_1^2 + \frac{1}{3}l^2 m_3 c_{12}^2 + l^2 m_3 c_1 c_{12}$$

$$D_{23} = \frac{1}{3}l^2m_3c_{12}^2 + \frac{1}{2}l^2m_3s_1c_{12} + l^2m_3s_{12}^2 + l^2m_3s_{12}c_{12} + l^2m_3s_1s_{12} + \frac{1}{3}l^2m_3c_{12}^2 + \frac{1}{2}l^2m_3c_1c_{12}$$

$$D_{33} = \frac{2}{3}l^2m_3c_{12}^2 + l^2m_3s_{12} + l^2m_3c_{12}s_{12}$$

Ku  $c_1 = \cos(\theta_1^*)$ ,  $s_1 = \sin(\theta_1^*)$ ,  $c_{12} = \cos(\theta_1^* + \theta_2)$ ,  $s_{12} = \sin(\theta_1^* + \theta_2)$  dhe  $\theta_1^* = \theta_1 + \frac{\pi}{2}$

- Elementet e Coriolis-it dhe Focave Centrifugale kanë vlerën:

$$h_1 = \sum_{k=1}^3 \sum_{m=1}^3 h_{1km} \dot{\theta}_k \dot{\theta}_m = 0$$

$$h_2 = \sum_{k=1}^3 \sum_{m=1}^3 h_{2km} \dot{\theta}_k \dot{\theta}_m = -\frac{1}{2}l^2m_2s_2\dot{\theta}_2^2 - l^2m_2s_2\dot{\theta}_1\dot{\theta}_2$$

$$h_3 = \sum_{k=1}^3 \sum_{m=1}^3 h_{3km} \dot{\theta}_k \dot{\theta}_m = \frac{1}{2}l^2m_2s_2\dot{\theta}_1^2$$

- Elementet e gravitetit janë (llogaritjet mund ti gjeni tek shtesa A.9):

$$c_1 = 0$$

$$c_2 = -\frac{1}{2}m_1glc_1 - \frac{1}{2}m_2glc_{12} - m_2glc_1$$

$$c_3 = -glm_3c_{12}$$

**Kodi dhe dizajnimi i hyrje/daljeve të bllok funksionit**

Sa i përket implementimit të analizës dinamike për krahun robotik atë e kemi implementuar përmes krijimit të një bllok funksioni në simulink (Figura 2.17).

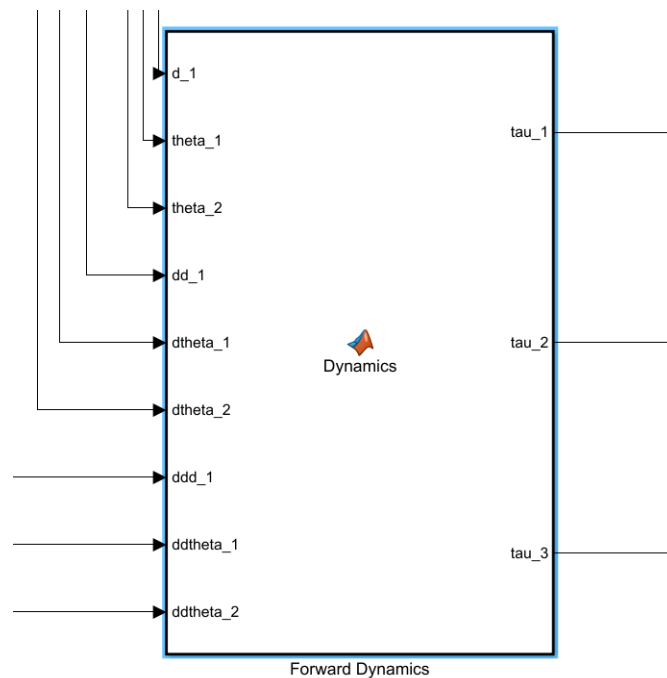


Figura 2.17: Bllok funksioni i cili kryen analizën dinamike të krahut robotik

Kurse kodi i cili ndodhet brenda këtij funksioni është paraqitur më poshtë:

```

1 function [tau_1, tau_2, tau_3] = Dynamics(d_1,theta_1,theta_2
2     ,dd_1,dtheta_1,dtheta_2,ddd_1, ddtheta_1,ddtheta_2)
3
4 m_1=7.49980007;
5 m_2=8.651076019;
6 m_3=8.651076019;
7
8 l_1=0.1;
9 l=0.48; % l=l_2=l_3=0.48
10
11 g=9.80665;
12
13 % Declare trigonometric functions
14 theta_1_90=theta_1+pi/2;
15
16 c_1=cos(theta_1_90);
17 c_2=cos(theta_2);
18 c_12=cos(theta_1_90+theta_2);
19
20 s_1=sin(theta_1_90);
21 s_2=sin(theta_2);
22 s_12=sin(theta_1_90+theta_2);

```

```

22
23 % Acceleration-Related Symmetric Matrix
24 D_11 = m_1+m_2+m_3;
25
26 D_12 = -(1/2)*l*m_2*c_2+c_1*l*m_3+(3/2)*l*m_3*c_12;
27
28 D_13 = (3/2)*l*m_3*c_12;
29
30 D_22=(1/3)*l^2*m_2*s_1^2+(1/3)*l^2*m_2*c_1^2+(1/3)*l^2*m_3*
    c_12^2+l^2*m_3*s_1^2+...
31     l^2*m_3*s_1*c_12+l^2*m_3*s_12^2+l^2*m_3*s_12*c_12+2*l^2*
    m_3*s_1*s_12+...
32     +l^2*m_3*c_1^2+(1/3)*l^2*m_3*c_12^2+l^2*m_3*c_1*c_12;
33
34 D_23=(1/3)*l^2*m_3*c_12^2+(1/2)*l^2*m_3*s_1*c_12+l^2*m_3*s_12
    ^2+l^2*m_3*s_12*c_12+...
35     l^2*m_3*s_1*s_12+(1/3)*l^2*m_3*c_12^2+(1/2)*l^2*m_3*c_1*
    c_12;
36
37 D_33=(2/3)*l^2*m_3*c_12^2+l^2*m_3*s_12^2+l^2*m_3*c_12*s_12;
38
39 % Coriolis and centrifugal terms
40 h_1 = 0; % no coriolis or centrifugal effect to the prismatic
    joint
41 h_2 = -(1/2)*m_2*s_2*l^2*dtheta_2^2-m_2*s_2*l^2*dtheta_1*
    dtheta_2;
42 h_3 = (1/2)*m_2*s_2*l^2*dtheta_1^2;
43
44 % Gravity terms
45 c_1 = 0;
46 c_2 = -(1/2)*m_1*g*l*c_1-(1/2)*m_2*g*l*c_12-m_2*g*l*c_1;
47 c_3 = -l*g*m_3*c_12;
48
49 %Multiplication of matrices
50 D_matrix=[D_11, D_12, D_13;
51           D_12, D_22, D_23;
52           D_13, D_23, D_33];
53
54 q_dd = [ddd_1; ddtheta_1; ddtheta_2];
55
56 h_matrix = [h_1; h_2; h_3];
57
58 q_d = [dd_1; dtheta_1; dtheta_2];
59
60 c_matrix = [c_1; c_2; c_3];
61
62 M_matrix = D_matrix * q_dd;
63

```

```

64 %torques
65 torques = M_matrix + h_matrix + c_matrix;
66 tau_1 = torques(1);
67 tau_2 = torques(2);
68 tau_3 = torques(3);
69 end

```

### 2.3.3 Sensori i Forcës

Në mes linkut të fundit (dorës) dhe shënuesit (marker-it) është i vendosur edhe një sensor i forcës i cili parashihet të llogarisë forcën me të cilën shënuesi aplikon në rrafshin ku shkruan [11]. Ky sensorë ndihmon që rrafshi mbi të cilin shkruajmë të mos dëmtohet dhe po ashtu mundëson që të kemi konsistencë të trashësisë së vijave nga shënuesi [11].

#### Principi i punës

Ekzistojnë dy lloje të sensorëve piezorezistues: sensorë piezorezistues të përshtatshëm për mikroinjektimin e qeliave, domethënë, matës të sfrocimit piezorezistues dhe membrana silikonit e gdhendur [12].

Tipi i cili është përdorur tek ky krah robotik është sensori piezorezistues me matës të sforcimit i cili përdorë matës të sforcimit i bërë nga një material përçues që ndryshon rezistencën e tij elektrike kur tërhiqet (ose shtypet). Matësi i sforcimit mund të ngjitet në një diafragmë që njeh një ndryshim në rezistencë kur elementi i sensorit deformohet. Ndryshimi i rezistencës shndërrohet në një sinjal dalës [12]. Ekzistojnë tre efekte të veçanta që kontribuojnë në ndryshimin e rezistencës së një përcjellësi. Këto janë:

- Rezistenca e një përcjellësi është proporcionale me gjatësinë e tij, kështu që shtrirja rrit rezistencën [12]
- Ndërsa përçuesi shtrihet, zona e tij e prerjes tërthore zvogëlohet, gjë që gjithashtu rrit rezistencën [12]
- Rezistenca e qenësishme e disa materialeve rritet kur ajo shtrihet [12]

Po ashtu, efekti piezorezistues, ndryshon shumë ndërmjet materialeve [12]. Ndjeshmëria përcaktohet nga faktori i matësit, i cili përcaktohet si ndryshimi i rezistencës relative i ndarë me sforcimin:

$$GF = \frac{\frac{\Delta R}{R}}{\epsilon} \quad (2.33)$$

Ku sforcimi përcaktohet si ndryshimi relativ në gjatësi  $l$ :

$$\epsilon = \frac{\Delta l}{l} \quad (2.34)$$

Ndryshimi i rezistencës në sensor zakonisht matet duke përdorur një qark urë Wheatstone-it (siç tregohet më poshtë) [12]. Kjo lejon që ndryshime të vogla në rezistencën e sensorit të shndërrohen në një tension daljeje. Matjet e matësit të tendosjes piezorezistuese bëhen duke përdorur një qark urë Wheatstone-it [12]. Një tension ngacmues duhet të sigurohet në urë [12]. Kur nuk ka tendosje dhe të gjitha rezistorët në urë janë të balancuara, atëherë dalja do të jetë zero volt [12]. Një ndryshim në presion (ngacmim me forcë) do të shkaktojë një ndryshim në rezistencat



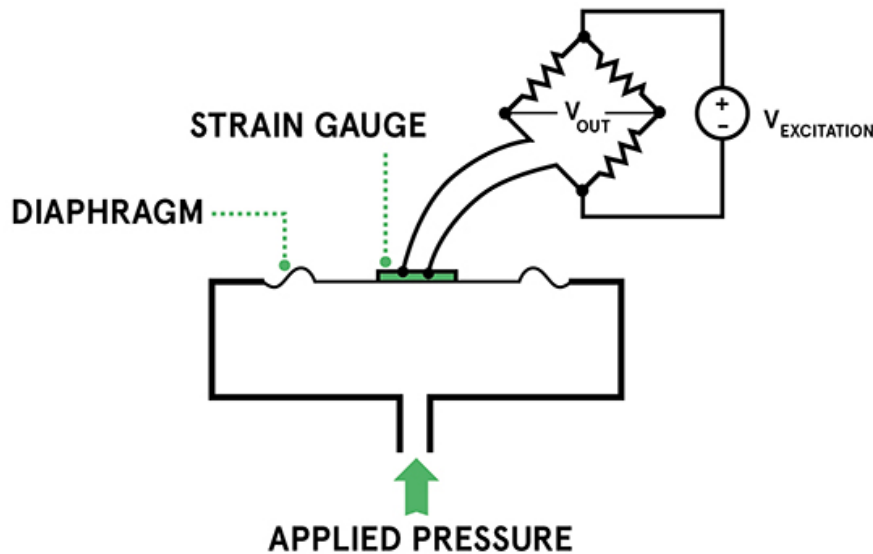


Figura 2.18: Matësi i sforcimit piezorezistuese me urë Wheatstone-it [12]

në urë duke rezultuar në një tension ose rrymë daljeje përkatëse. Mënyra se si llogaritet kjo tregohet në formulën e mëposhtme [12].

$$V_o = \left( \frac{R_3}{R_3 + R_4} - \frac{R_2}{R_1 + R_2} \right) V_{ex} \quad (2.35)$$

Performanca mund të përmirësohet duke përdorur dy ose katër elementë lexues (ndijues) në urë, me elementët në çdo çift që i nënshtrohen sforcimit (tendosjes) së barabartë dhe të kundërt [12]. Kjo rrit sinjalin e daljes dhe mund të minimizojë efektet e temperaturës në elementët e sensorit [12].

Prandaj në shënuesin tonë ne planifikojmë të shkruajm brenda rrafshit shënues për të gjeneruar forcën e nevojshme që shënuesi të bëjë vija të plota dhe të vazhdueshme. Në rastin konkret është supozuar se rrafshi gjendet i larguar nga krahu roboti për 19.5cm dhe vlerat për komandat pen-up (shënuesi nuk prek rrashfin) dhe pen-down (shënuesi ka kontakt me rrafshin) janë  $pen - up = 18cm$ ,  $pen - down = 20cm$  pra që shënuesi të krijoj kontakt të mirë me rrafshin kemi supozuar se rrafshi gjendet i larguar për 20cm duke krijuar kështu një diferencë në distancë prej 0.5cm e cila kontrollohet nga sensori i forcës ku tregon forcë të caktuar për këtë distancë. Kurse distanca e pen-up është 1.5cm nga shënuesi thjeshtë sa për të qenë të sigurt se nuk do të ketë ndonjë pengesë në levizje të dorës (nëse rrafshi nuk është saktësisht i njëjtë në të gjitha drejtimet por ka një diferencë prej  $\pm 0.5cm$ ) por kjo distancë mund të zvogëlohet më shumë nëse kërkohet shpejtësi më e madhe për shënim.

## 2.4 Planifikimi i rrugës dhe trajektore

### 2.4.1 Planifikimi i rrugës

#### Hyrje

Planifikimi i rrugës është një aspekt kryesor i robotikës, duke u mundësuar agjentëve autonome të lëvizin nëpër mjedise komplekse në mënyrë efikase duke shmangur pengesat dhe duke iu përmbajtur kufizimeve specifike të detyrave [12]. Kjo aftësi themelore gjen aplikime në një spektër fushash, nga automatizimi industrial deri te automjetet autonome dhe më gjerë. Në ndjekje të sistemeve robotike të fuqishme dhe adaptive, zhvillimi i algoritmeve efektive të planifikimit të rrugës mbetet një pikë qendrore e kërkimit dhe inovacionit [13].

#### Planifikimi i rrugës nëprmet pikave në hapësirë

Në fushën e planifikimit të rrugës, një konsideratë thelbësore është hartimi i algoritmeve që drejojnë robotët nga një pozicion fillestar në një qëllim të caktuar, duke optimizuar kritere të ndryshme si koha e udhëtimit, konsumi i energjisë ose siguria [13]. Një qasje për arritjen e këtij objekti është përdorimi i algoritmeve të planifikimit të bazuara në pika, të cilat zbërthejnë shtegun në një seri pikash të ndërmjetme, ose pika të rrugës, dhe planifikojnë lëvizjen e robotit midis këtyre pikave [13].

Procesi zakonisht fillon me përcaktimin e pozicioneve fillestare dhe të qëllimit në mjedis. Më pas, një sekuençë e pikave të rrugës përcaktohet bazuar në trajektoren e rrugës së dëshiruar, kërkesat e detyrës dhe kufizimet mjedisore. Këto pika të rrugës shërbejnë si qëllime të ndërmjetme që roboti të lëviz drejt, duke lehtësuar zbërthimin e shtegut të përgjithshëm në segmente të menaxhueshme [13].

Pasi të vendoset sekuenca e pikave të rrugës, hapi tjetër përfshin planifikimin e lëvizjes së robotit midis pikave të njëpasnjëshme. Për këtë qëllim mund të përdoren disa algoritme, në varësi të karakteristikave specifike të mjedisit dhe dinamikës së robotit por të cilat nuk trajtohen në këtë tezë [13].

Një qasje e zakonshme është përdorimi i algoritmeve gjeometrike, të tilla si interpolimi i rrugës së drejtë ose algoritmi i planifikimit të rrugës Dubins për automjetet me kufizime diferenciale [13]. Këto algoritme gjenerojnë trajektore të thjeshta, por efektive midis pikave të rrugës, shpesh duke minimizuar distancën ose kohën e udhëtimit [13].

Në këtë tezë është përdorur metoda e zakonshme i cili lidh pikat në hapësirë (waypoints) në vijë të drejtë (interpolimi i rrugës së drejtë), duke formuar kshu shkronjat e caktuara. Më poshtë mund të shikohen pikat e hapësirës të cilat janë përdorur për gjenerimi e shkronjës L.

```
1 waypoints_L = [  
2     0.18 0.32 0.7; ...  
3     0.2 0.32 0.7; ...  
4     0.2 0.32 0.7; ...  
5     0.2 0.32 0.3; ...  
6     0.2 0.7 0.3; ...  
7     0.18 0.7 0.3; ]  
8  
9 waypoint_L_T=transpose waypoints_L  
10  
11 waypoints_Visuals_L = [
```

```

12         0.18 0.32 0.7; ...% pen down
13         0.2 0.32 0.7; ...
14         0.2 0.32 0.6; ...
15         0.2 0.32 0.5; ...
16         0.2 0.32 0.4; ...
17         0.2 0.32 0.3; ...
18         0.2 0.4 0.3; ...
19         0.2 0.5 0.3; ...
20         0.2 0.6 0.3; ...
21         0.2 0.7 0.3; ...
22         0.18 0.7 0.3]; % pen up

```

### 2.4.2 Planifikimi i trajektores

Planifikimi i trajektores është një ligj lëvizjeje që përcakton kohën sipas një rruge të caktuar gjeometrike [13]. Prandaj, qëllimi i planifikimit të trajektores është të plotësojë nevojat e detyrave dhe të gjenerojë inputet referencë të kërkuara nga sistemi i kontrollit, në mënyrë që robotët të mund të kryejnë lëvizjet [14]. Hyrat e çdo algoritmi të planifikimit të trajektores janë një rrugë e pritshme e lëvizjes, parametrat e sistemit kinematikë dhe dinamikë të robotëve, dhe rezultatet janë pozicionet, shpejtësitë dhe nxitimet e pikave të interpolimit të çdo nyjeje ose efekti fundor [13].

Për të siguruar që robotët të mund të funksionojnë në mënyrë efikase dhe të qëndrueshme, përgjithësisht kërkohet që profili i lëvizjes të përmbushë të paktën derivatin e parë të vazhdueshëm dhe duhet të ketë karakteristikat e nxitimit dhe ngadalësimit të shpejtë dhe shpejtësisë konstante [13]. Aktualisht, algoritmet e kontrollit të shpejtësisë që përdoren zakonisht në sistemet robotike përfshijnë trajektoren trapezoidale dhe trajektoren me profilin e shpejtësisë S [13].

Në rastin tonë ne kemi përdorur trajektoren trapezoidale pasi përmbush mjaftueshëm kërkesat e krahut robotik në fjalë. Trajektorja trapezoidale do të thotë që profili i shpejtësisë është trapezoid, dhe kthesat e saj të plota janë paraqitur në Figurën 2.19. Trajektorja e plotë përbëhet nga tre faza. Në fazën e parë, nxitimi ndryshon nga zero në një nxitim fiks, dhe shpejtësia rritet në mënyrë lineare me një nxitim fiks nga zero. Në fazën e dytë, kur shpejtësia arrin vlerën e dëshiruar, ajo hyn në fazën e shpejtësisë konstante [13]. Në këtë fazë, shpejtësia është konstante dhe nxitimi befas ndryshon në zero. Në fazën e tretë, shpejtësia fillon të ulet me një nxitim fiks derisa të arrijë zero. Ekuacionet pjesë-pjesë (piecewise) të nxitimit shprehen si:

$$a(t) = \begin{cases} a_{max} & (0 \leq t \leq t_1) \\ 0 & (t_1 \leq t \leq t_2) \\ -a_{max} & (t_2 \leq t \leq t_3) \end{cases} \quad (2.36)$$

Ekuacionet pjesë-pjesë (piecewise) të shpejtësisë shprehen si:

$$v(t) = \begin{cases} a_{max}t & (0 \leq t \leq t_1) \\ v_2 & (t_1 \leq t \leq t_2) \\ v_2 - a_{max}(t - t_2) & (t_2 \leq t \leq t_3) \end{cases} \quad (2.37)$$

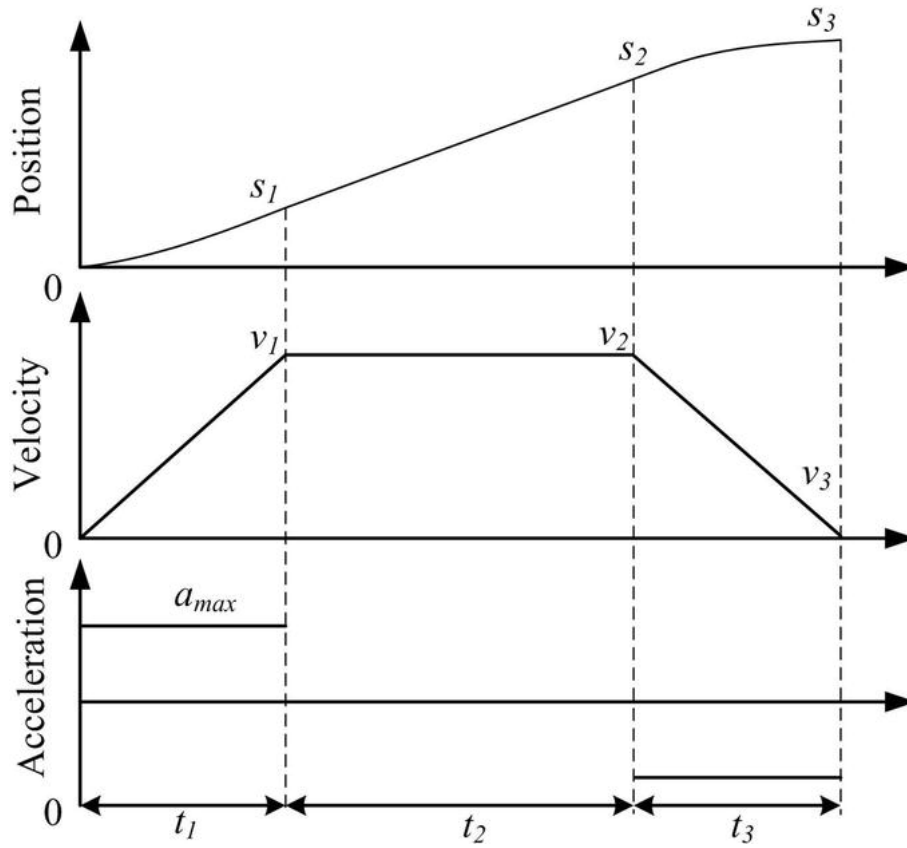


Figura 2.19: Lakoret e pozicionit, shpejtësisë dhe nxitimit të një trajektoreje trapezoidale [14]

Ekuacionet pjesë-pjesë (piecewise) të pozitës shprehen si:

$$s(t) = \begin{cases} \frac{1}{2}a_{max}t^2 & (0 \leq t \leq t_1) \\ s_1 + v_2(t - t_1) & (t_1 \leq t \leq t_2) \\ s_2 - \frac{1}{2}v_2 - a_{max}(t - t_2)^2 & (t_2 \leq t \leq t_3) \end{cases} \quad (2.38)$$

ku intervalët kohore  $t_1$ ,  $t_2$ , dhe  $t_3$  janë të specifikuara.  $s_1$ ,  $s_2$  dhe  $s_3$  janë vlerat përkatëse të pozicionit, dhe  $v_1$ ,  $v_2$  dhe  $v_3$  janë vlerat përkatëse të shpejtësisë.  $a_{max}$  është një vlerë maksimale e nxitimit.

Implementimi i profilit të trajektores është bërë duke përdorur bllokun e librarisë së Simulink Robotics System Toolbox (shiko Figurën 2.20):

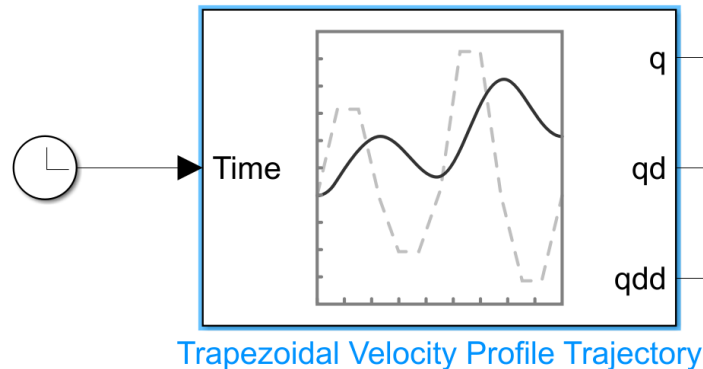


Figura 2.20: Bllok funksioni i trajektores trapezoidale i librarisë Robotics System Toolbox

Kurse konfigurimi i këtij blloku është shprehur përmes Figurës 2.21, ku caktohet pikat e rrugës të cilat roboti duhet të kalojë:

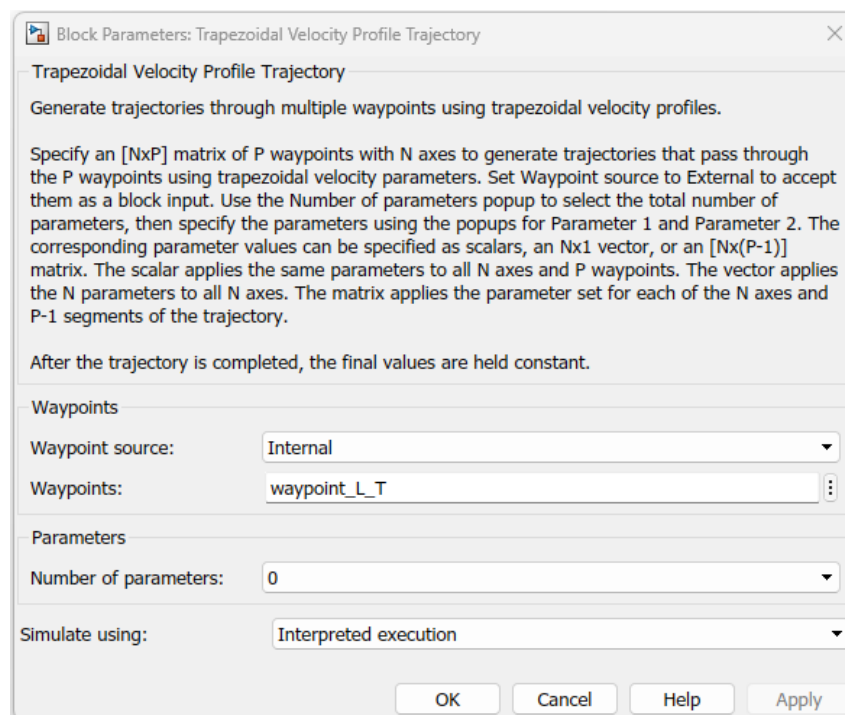


Figura 2.21: Konfigurimi i bllok funksioni të trajektores trapezoidale

Nëse ekzekutojmë bllokun e tillë do të marrim grafën si në Figurën 2.22:

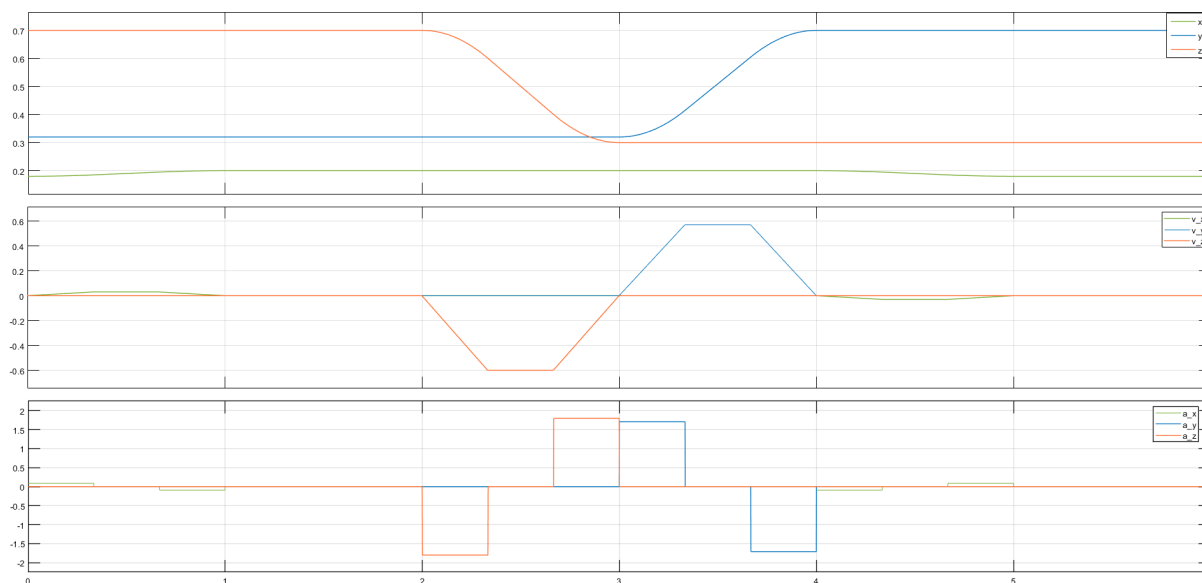


Figura 2.22: Grafiqet për pozitën, shpejtësinë, dhe nxitimin e trajektores trapezoidale

### Vizualizimi i trajektores në Simulink

Për vizualizimin e trajektores është përdorur bllok funksioni i paraqitur tek Figura 2.23.

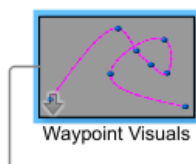
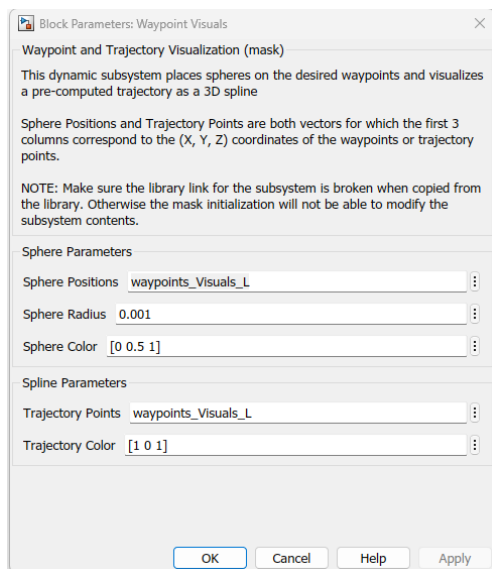
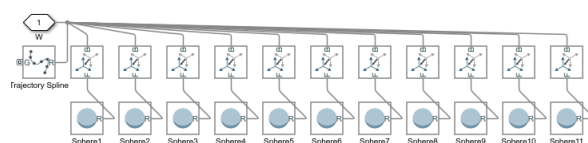


Figura 2.23: Bllok funksioni për vizualizimin e trajektores në Simulink

Ku ky bllok konfigurohet duke i shtuar më shumë pika të rrugës për arsye se trajektorja e cila vizatohet bëhet përmes algoritmit B-Spline. Prandaj përdorim një tjetër bashkësi të pikave të rrugës për të shfaqur rrugën të ngjashme me atë të cilën përshkruhet nga shënuesi. Konfigurimin e bllokut mund ta gjeni tek Figura 2.24. Kurse vizualizimi i rrugës është treguar tek Figura 2.25.



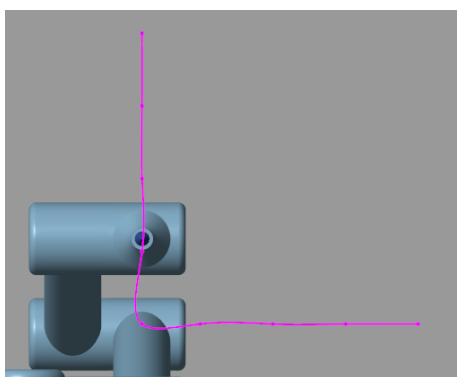
(a) Parametrat e bllok funksionit të vizualizimit të trajektores



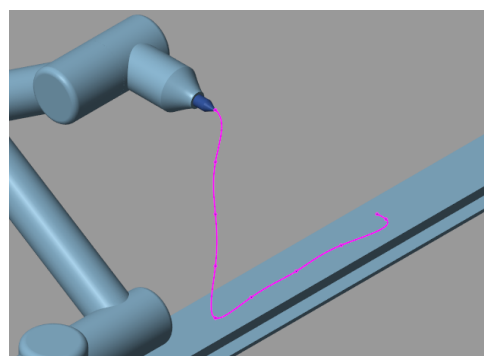
(b) Komponentet e bllok funksionit të vizualizimit të trajektores

Figura 2.24: Parametrat dhe gjenerimi i komponenteve të bllok funksionit të vizualizimit të trajektores

Rruga e përshkruar nga shënuesi do të duket:



(a) Pamja para



(b) Pamja izometrike

Figura 2.25: Rruga e përshkruar nga shënuesi

## 2.5 PID rregullatori

Një PID rregullator (Proportional-Integral-Derivative) është një komponent kyç në teorinë dhe inxhinierinë e kontrollit, i përdorur gjerësisht në aplikacione të ndryshme industriale dhe inxhinierike për të rregulluar dhe mbajtur rezultatet e dëshiruara të sistemeve dinamike. Rregullatori PID llogarit vazhdimisht një sinjal gabimi si diferencë midis pikës së caktuar të dëshiruar dhe vlerës të procesit të matur [15]. Ky sinjal gabimi përdoret më pas për të rregulluar hyrjen e kontrollit në proces.

Le të shikojmë pjesët përbërëse të një rregullatori PID:

- **Termi proporcional (P):** Termi proporcional prodhon një dalje që është proporcionale me vlerën aktuale të gabimit [15]. Ai llogaritet duke shumëzuar gabimin me një faktor konstant ( $K_p$ ) [15], i njohur si fitimi proporcional. Termi proporcional ndihmon për të reduktuar gabimin aktual (shih Figurën 2.26), por nuk mund ta eliminojë atë plotësisht. Nëse fitimi proporcional është shumë i lartë, sistemi mund të bëhet i paqëndrueshëm dhe të lëkundet rreth pikës së caktuar [15]. Matematikisht, termi proporcional mund të përfaqësohet si:  $P(t) = K_p e(t)$

- $P(t)$  është termi proporcional në kohë  $t$
- $K_p$  është fitimi proporcional
- $e(t)$  është gabimi në kohë  $t$

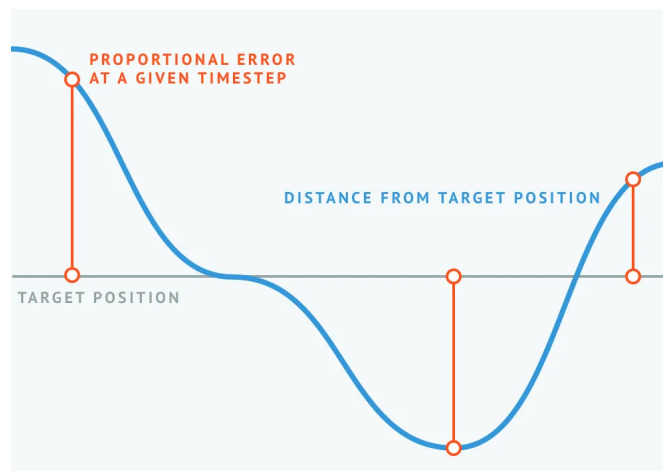


Figura 2.26: Definimi i fitimit proporcional [16]

- **Termi integral (I):** Termi integral integron gabimin me kalimin e kohës (shih Figurën.2.27) [15]. Ky term përshpejton lëvizjen e procesit drejt pikës së caktuar dhe eliminon gabimin e mbetur të gjendjes së qëndrueshme që ndodh me një kontrollues të pastër proporcional. Termi integral zvogëlon kompensimin midis ndryshores së procesit dhe pikës së caktuar. Megjithatë, mund të shkaktojë që sistemi të tejkalojë pikën e caktuar dhe madje ta bëjë sistemin të paqëndrueshëm nëse nuk akordohet siç duhet. Matematikisht, termi integral mund të përfaqësohet si:  $I(t) = K_i \int_0^t e(\tau) d\tau$

- $I(t)$  është termi integral në kohë  $t$
- $K_i$  është fitimi integral
- $e(\tau)$  është gabimi në kohë  $\tau$



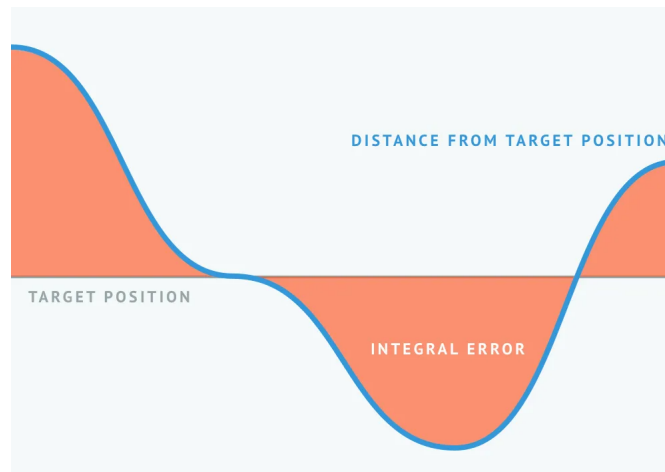


Figura 2.27: Definimi i fitimit integral [16]

- **Termi i derivatit (D):** Termi derivativ parashikon vlerën e gabimit në të ardhmen bazuar në shkallën aktuale të ndryshimit (shih Figurën 2.28) [15]. Ai zbut përgjigjen e sistemit, duke ngadalësuar termin proporcional ndërsa ndryshorja e procesit i afrohet pikës së caktuar. Termi derivat ndihmon në reduktimin e tejkalimit dhe përmirësimin e qëndrueshmërisë së sistemit [15]. Megjithatë, mund të shkaktojë që sistemi të bëhet i zhurmshëm dhe të rrisë potencialin për paqëndrueshmëri nëse vendoset shumë lart. Matematikisht, termi derivat mund të përfaqësohet si:  $D(t) = K_d \frac{de(t)}{dt}$

- $D(t)$  është termi derivativ në kohë  $t$
- $K_d$  është fitimi derivativ
- $\frac{de(t)}{dt}$  është gabimi në kohë  $\tau$



Figura 2.28: Definimi i fitimit derivativ [16]

Sinjali dalës i rregullatorit PID është shuma e këtyre tre termave (paraqitja në blloqe është shfaqur në Figurën 2.29):

$$u(t) = P(t) + I(t) + D(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (2.39)$$

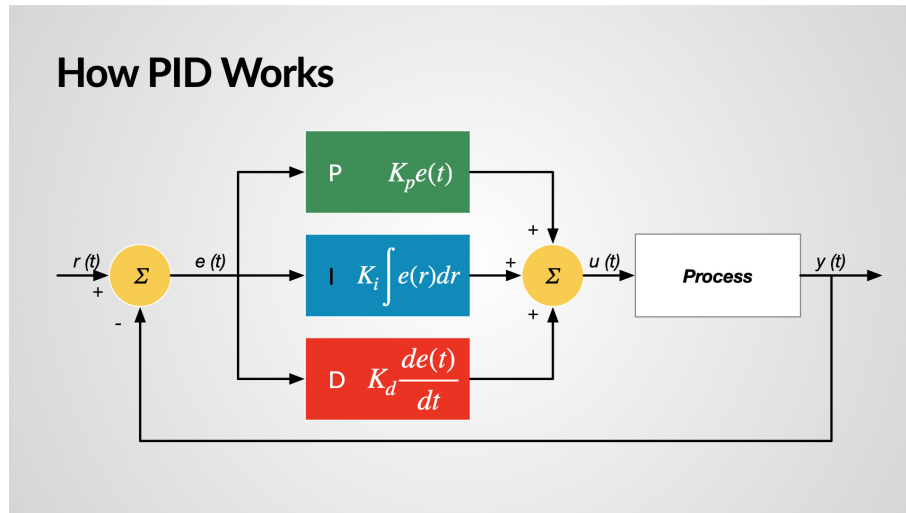


Figura 2.29: Konstruktimi bazik i PID rregullatorit [17]

Kurse PID rregullatori i cili funksionon përmes kohës diskrete dhe jo asaj kontinue si mësipër, do të marrë formën si vijon ku së pari bëhet diskretizimi i pjesës integrale (integratorit) përmes metodës direkte të Eulerit dhe pastaj bëjmë diskretizimin e pjesës derivative të PID rregullatorit përmes metodën direkte të Eulerit (shih Shtesën A.11):

$$U(z) = P + IT_s \frac{1}{z-1} + D \frac{N}{1 + NT_s \frac{1}{z-1}} \quad (2.40)$$

Ku:

- $P$  është fitimi propocional për kohë diskrete
- $I$  është fitimi integral për kohë diskrete
- $D$  është fitimi derivativ për kohë diskrete
- $N$  është koeficienti i filtrimit

### 2.5.1 PID rregulatori i paraqitur në Matlab

Nga informatat të dhëna më sipër implementimi i rregullatorit PID në Matlab është mjaft i thjeshtë. Në Figurën 2.30 është demonstruar në PID ku për secilën variabël të nyjeve krijohet një lidhje e veçantë:

Kurse modifikimi i parametrave të rregullatorit PID bëhet përmes dritares të prezentuar në Figurën 2.31 ku caktohen vlerat për komponentet e rregullatorit PID pra caktohen vlerat për P-komponenten propocionale, I-komponenten integrale, dhe D-komponenten derivative.

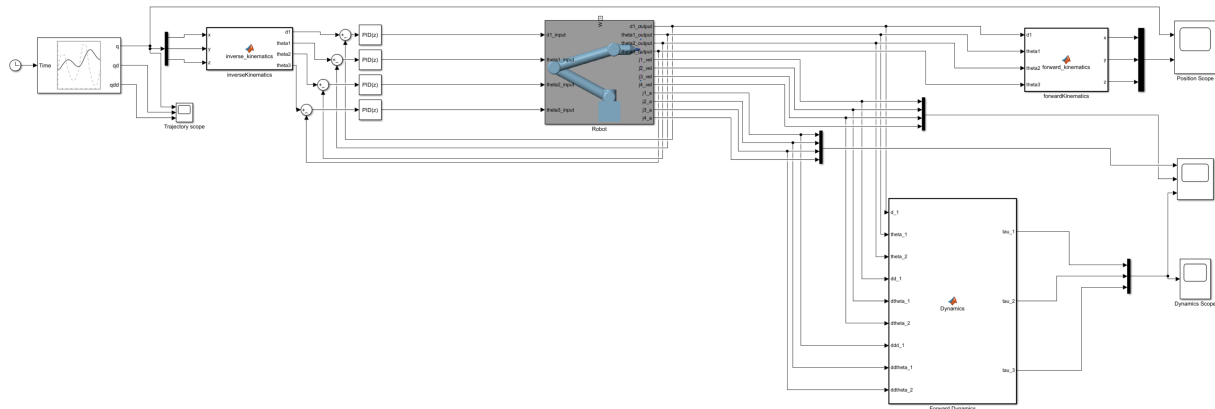


Figura 2.30: Lidhja e rregullatorit PID në Matlab

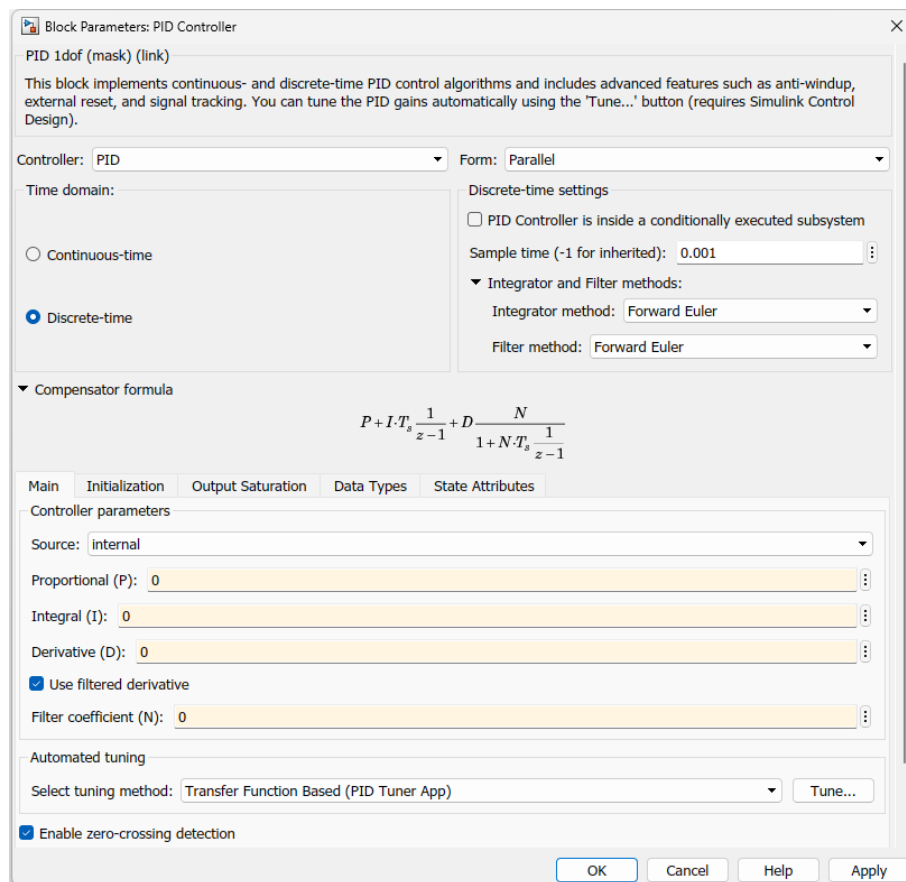


Figura 2.31: Dritarja për modifikimin e parametrave të rregullatorit PID diskret

# Kapitulli 3

## Rezultatet eksperimentale

Ky kapitull ka për qëllim të largoj hapësirën midis modeleve teorike dhe performancës së robotëve në botën reale përmes vërtetimit eksperimental. Objektivat specifike të këtij vërtetimi janë:

- **Verifikimi i saktësisë së modelit:** Krahasimi i sjelljes aktuale të robotit në kushte të kontrolluara me sjelljen e parashikuar nga modeli Simulink.
- **Kuantifikimi i gabimeve të modelit:** Për të identifikuar dhe për të përcaktuar çdo mospërputhje midis parashikimeve të modelit dhe të dhënave eksperimentale.
- **Përmirësimi i besnikërisë së modelit:** Për të përmirësuar modelin Simulink duke përfshirë njohuritë e marra nga testimi eksperimental.

### 3.1 Lidhja e blloqeve të shtjelluar më sipër

Në Figurën 3.1 kemi paraqitur komplet sistemin e krahut robotik i cili është krijuar prej bllok funksioneve të spjeguar më sipër.

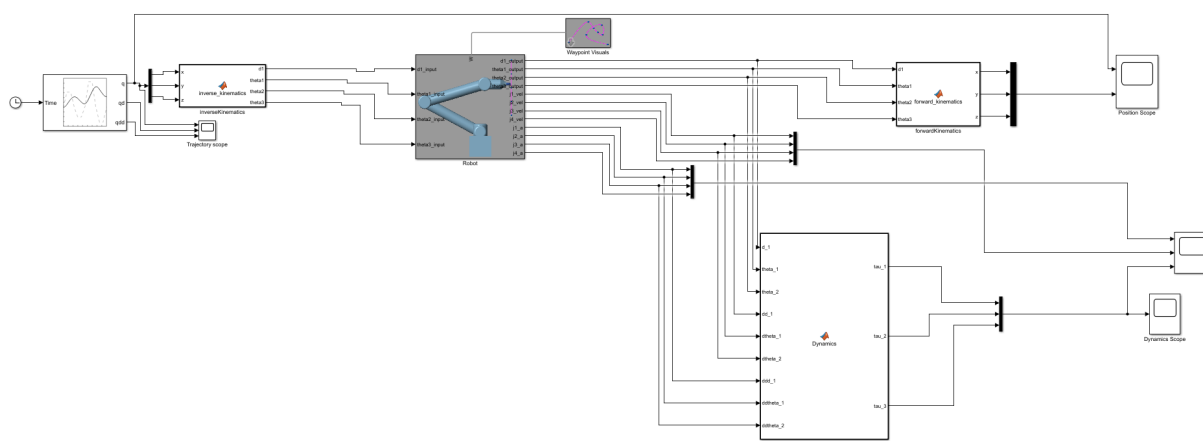


Figura 3.1: Sistemi komplet i krahut robotik

### 3.1.1 Rezultatet Eksperimentale me planifikim të rrugës L

Eksperimenti i cili është ekzekutuar ka qenë një përmes rrugës së planifikuar në formë të shkronjës L i prezentuar në Figurën 3.2:

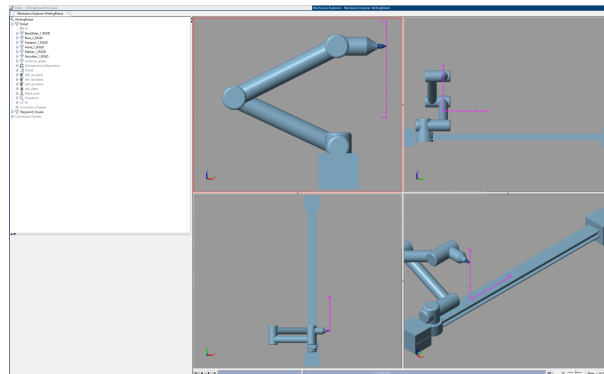


Figura 3.2: Simulimi i rrugës në formë të shkronjës L

Trajektorja e cila do të krijohet kur ekzekutohet simulimi i krahut robotik me rrugën në formë të shkronjës L do të marrim rezultatet e prezantuara në Figurën 3.3:

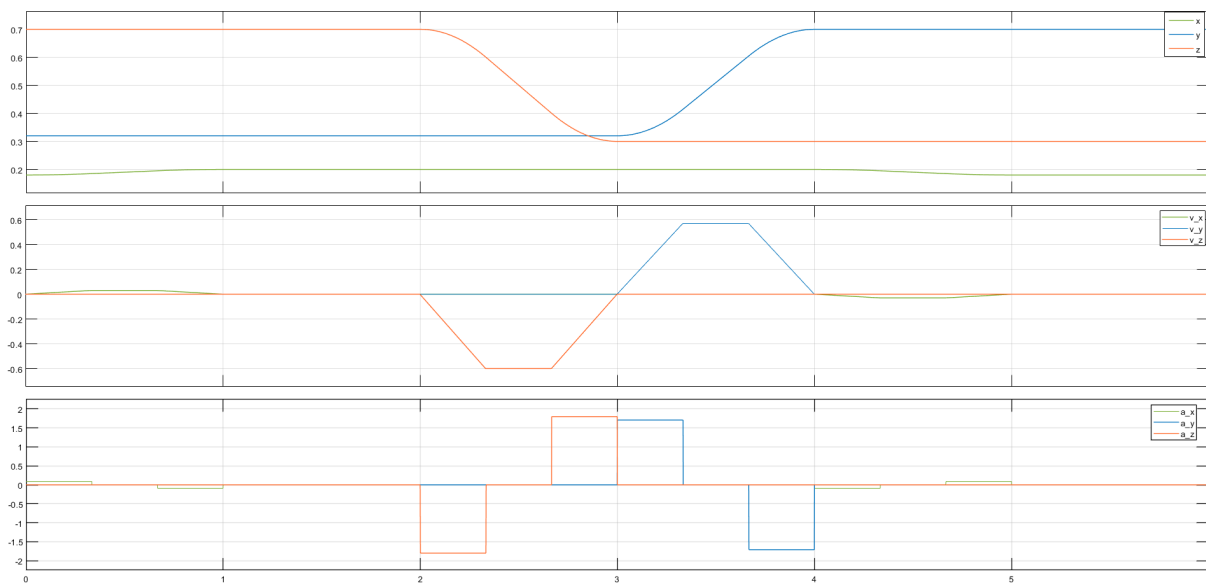


Figura 3.3: Trajektorja për rrugën në formë të shkronjës L

Kurse në Figurën 3.4 analizohet funksioni i kinematikës direkte e cila parashikohet të jetë e njëjtë me rrugën në formë të shkronjës L:

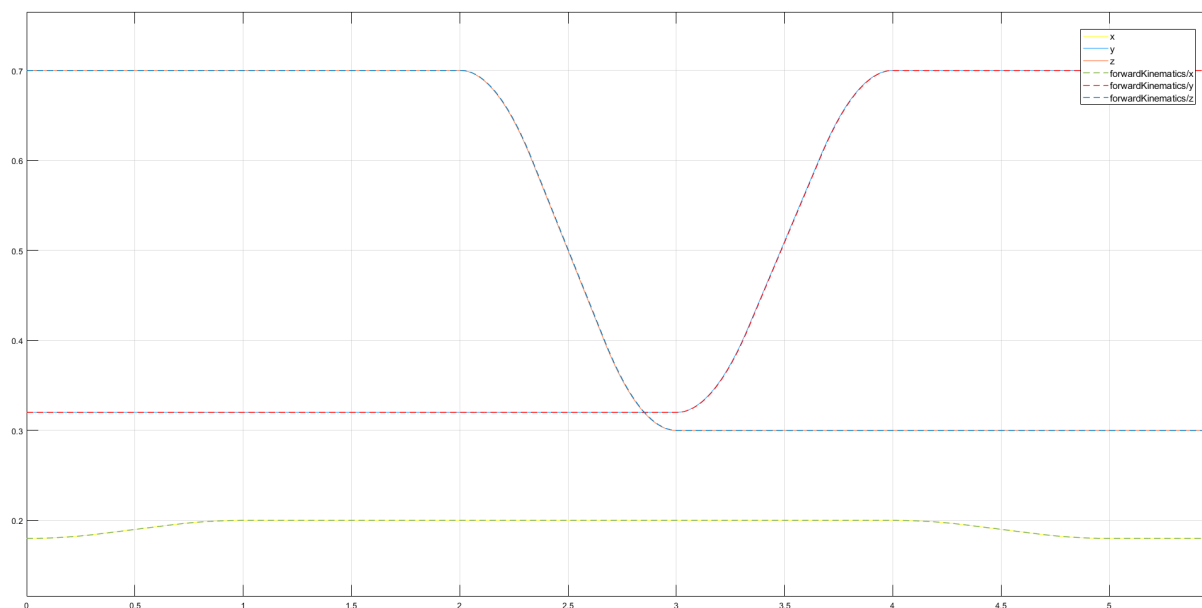


Figura 3.4: Analiza e kinematikës direkte me rrugën në formë të shkronjës L

Rezultatet e dinamikës direkte të krijuara nga lëvizjet e krahut robotik japin këto momente të focave (shiko në Figurën 3.5 dhe Figurën 3.6):

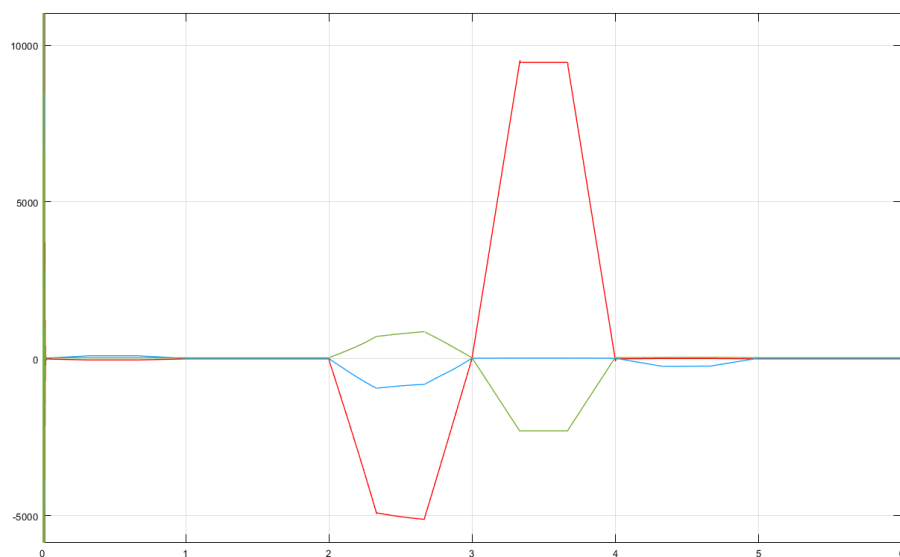


Figura 3.5: Analiza e dinamikës për rrugën në formë të shkronjës L

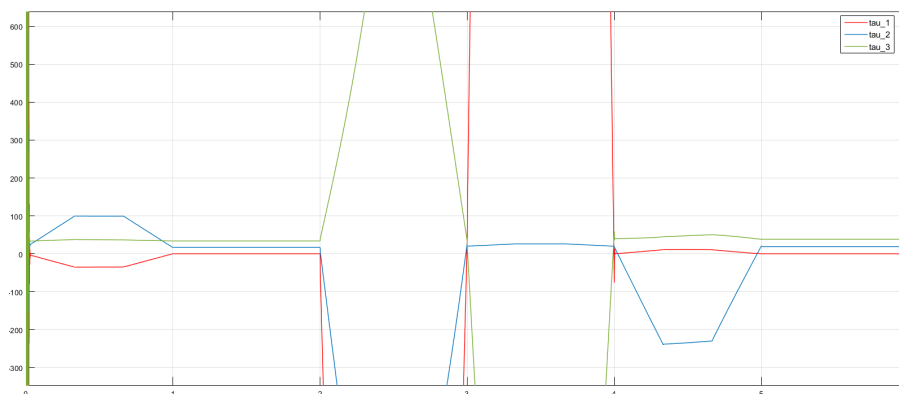


Figura 3.6: Analiza e dinamikës për rrugën në formë të shkronjës L e zmadhuar

### 3.1.2 Rezultatet Eksperimentale të rregullimit të sistemit me rregullator PID

Në këtë seksion janë treguar rezultatet e fituara nga rregullimi i sistemit me PID rregullator të paraqitur në Figurën 3.7.

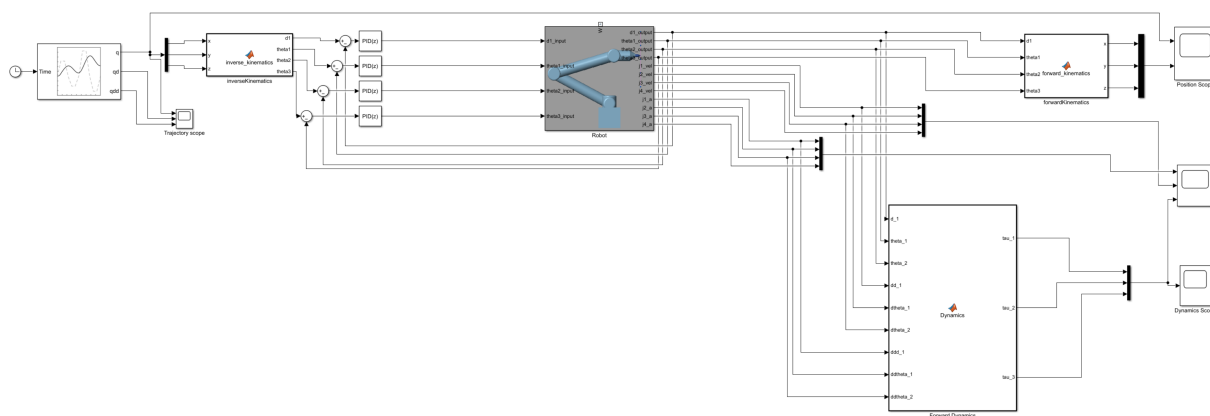


Figura 3.7: Sistemi i krahut robotik me rregullator PID diskret

#### Analiza e rregullatorit PID me vlera të ndryshme të $K_p$ , $K_i$ , $K_d$ për trajektoren vertikale

Në këtë seksion është analizuar rregullimi (përshkruarja) i trajektores vertikale duke përdorur aplikacionin e siguruar nga Matlab PID Tuner App, ku konfigurimi i PID rregullatorit lehtësohet shumë, krejt çka duhet të caktojme është se sa shpejt dëshirojmë që sistemi jonë të veproj dhe me sa saktësi dëshirojmë që ai të veproj.

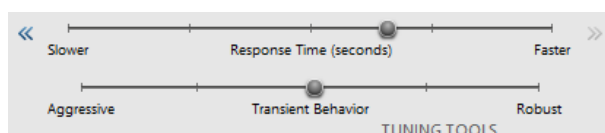


Figura 3.8: Veglat për akordim të PID rregullatorit

Pra, duke lëvizur sliderat e paraqitur në Figurën 3.8 mund të akordojmë PID rregullatorët derisa marrim sinjalin e dëshiruar në kohë sa më të shpejtë (shih Figurën 3.9):

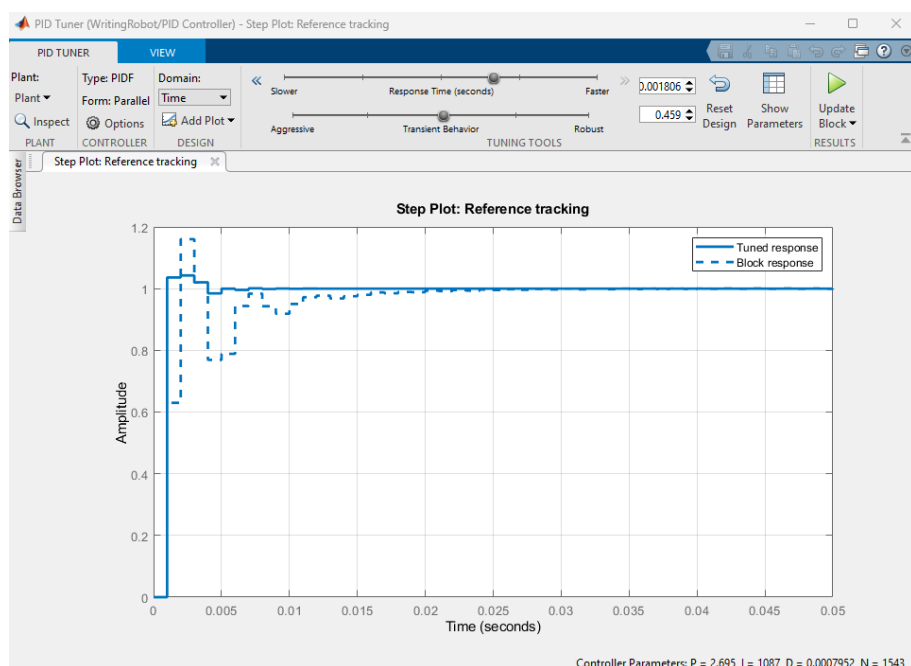


Figura 3.9: Grafi i cili pritet te fitohet pas akordimit të PID rregullatorit

Pas disa provave kemi ardh deri te rezultati më ideal i mundshëm ashtu si shfaqet në Figurën 3.10:

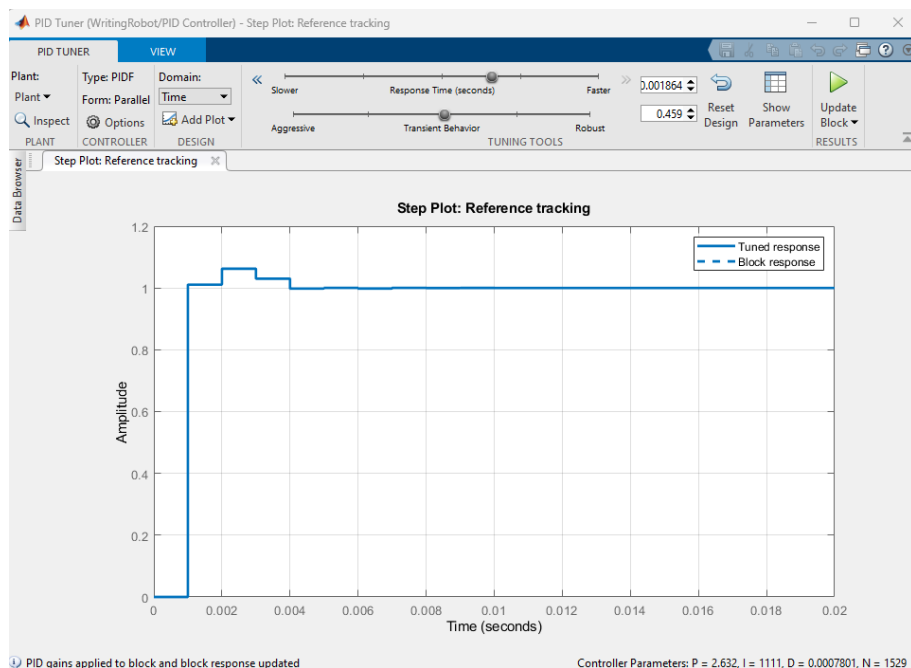


Figura 3.10: Grafi me rezultatin më ideal i cili pritet te fitohet pas akordimit të PID rregullatorit

Siç mund ta shihni vlerat të cilat do të përdoren për PID rregullatorë janë  $P = 2.632$ ,  $I = 1111$ ,  $D = 0.0007801$ ,  $N = 1529$ . Këto vlera pas ekzekutimit japin rezultatet më të mira të mundshme ku koha e qetësimit ndodh brenda 10 milisekondave kohë e cila është e pranueshme për lëvizjen 1 sekond (shih Figurën 3.11 dhe Figurën 3.12).

Arsyeja e oscilimeve të shumta të pozitës Z është se nyja e fundit duhet të kalojë nga pozita paralele me boshtin Z në pozitë normale me boshtin Z duke krijuar kështu oscilime për pozi-



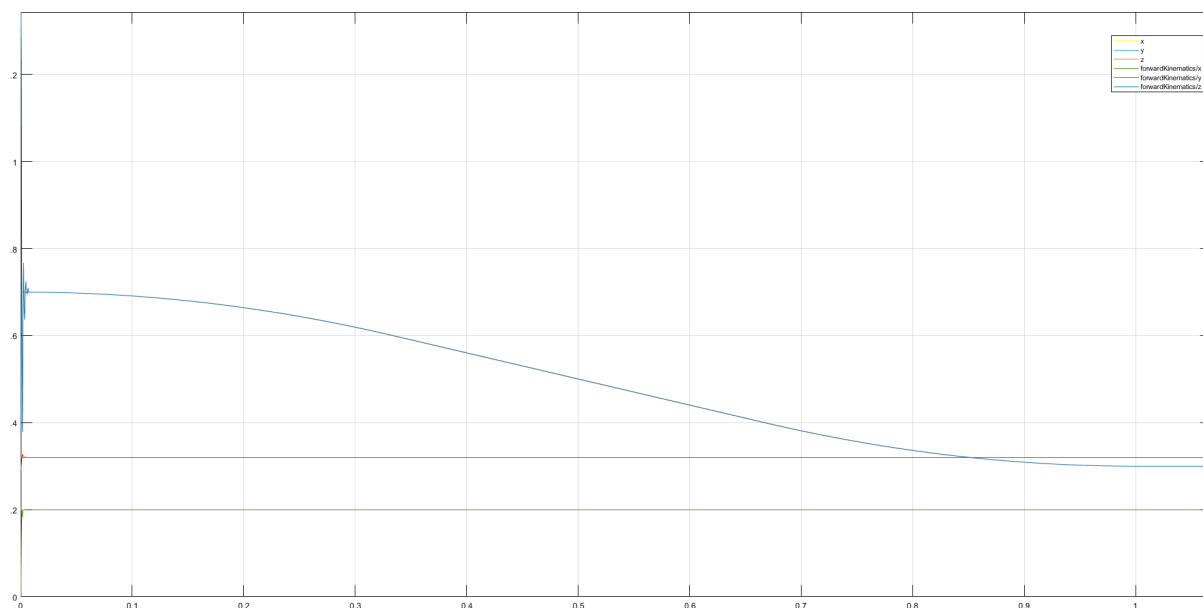


Figura 3.11: Rezultati i eksperimentit për trajektore vertikale brenda 1s

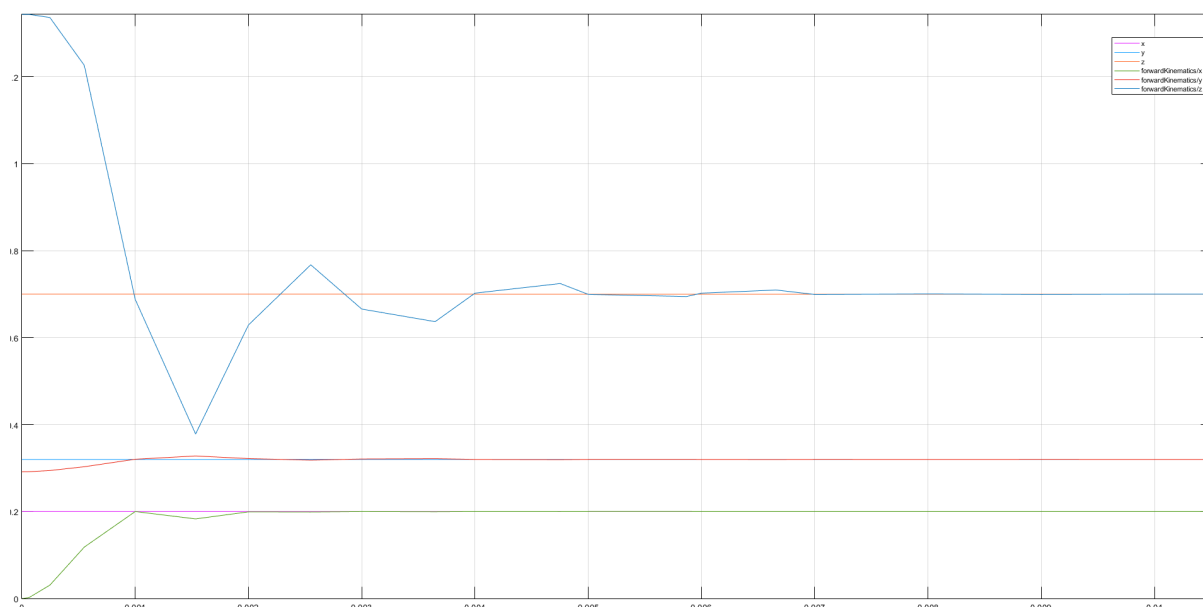


Figura 3.12: Rezultati i eksperimentit për trajektore vertikale brenda 10ms

cionim të menjëhershëm të lapsit brenda 10ms.

Kurse për drejtimet tjera të lëvizjes p.sh. në drejtim të boshtit X apo Y shënuesi mban pozitë të njëjtë prandaj këtu shihet qartë rregullimi i PID rregullatorit nëse për një ndryshim të pozitës (në fakt paraqiten sinjalet shkallë) dhe mund të shikohet stabilizimi dhe ndjekja e pozitës së dëshiruar ndodh në kohë më të shkurtër se 0.005s.

### Analiza e rregullatorit PID me vlera të ndryshme të $K_p$ , $K_i$ , $K_d$ për trajektoren horizontale

Duke përdor vlerat e njëjta për trajektoren horizontale marrim rezultatet më të mira të mundshme ku koha e qetësimit ndodh brenda 10 milisekondave kohë e cila është e pranueshme për lëvizjen 1 sekond (shih Figurën 3.13 dhe Figurën 3.14).

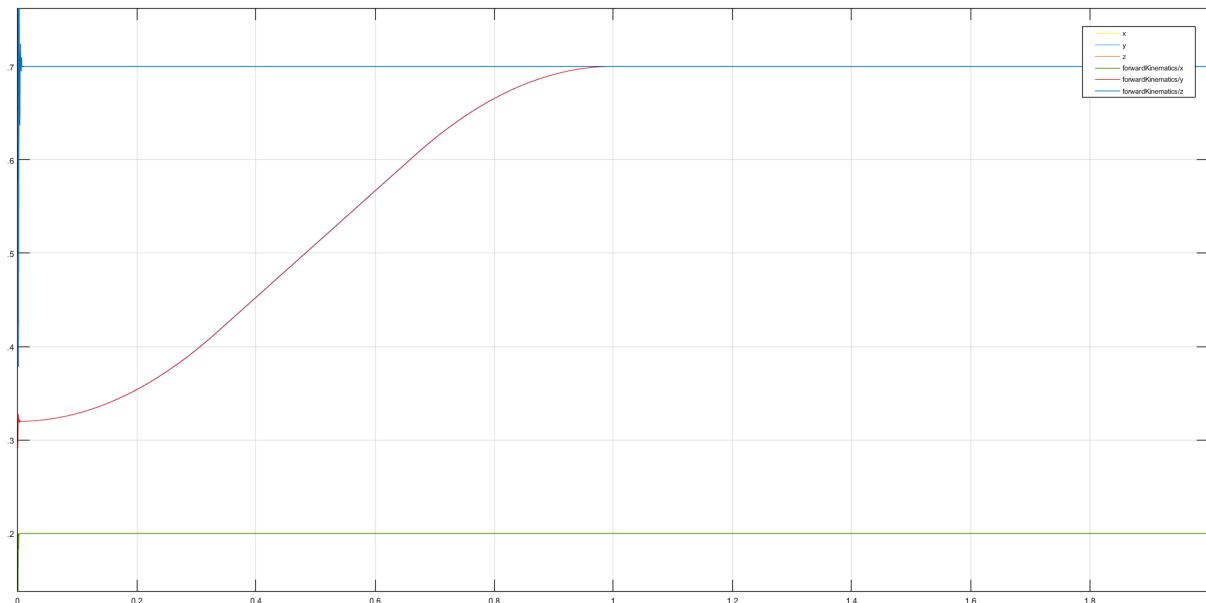


Figura 3.13: Rezultati i eksperimentit për trajektore horizontale brenda 1s

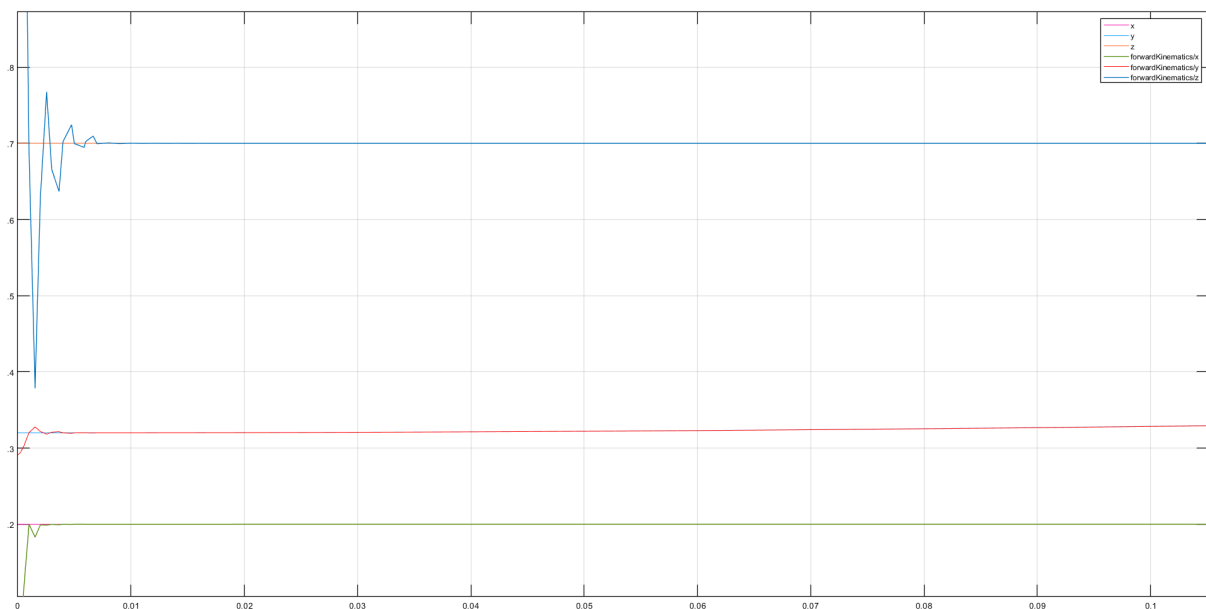


Figura 3.14: Rezultati i eksperimentit për trajektore horizontale brenda 10ms

Edhe këtu tek trajektorja horizontale mund të nxjerrim konkluzione të ngjashme me atë sikur tek eksperimenti me trajektore vertikale por tani lëvizja shihet se ndodh në lakoren  $y$  dhe jo atë  $z$  kurse lakorja  $x$  është komplet e njëjtë me atë të eksperimentit paraprak

### Analiza e rregullatorit PID me vlera të ndryshme të $K_p$ , $K_i$ , $K_d$ për trajektoren në drejtim të boshtit X (pen-up dhe pen-down)

Tani testojmë drejtimin e fundit me të cilin përfundohet analiza e lëvizjes në drejtim të gjitha boshteve. Për testim në drejtim të boshtit X është marrë trajektorja e lëvizjes pen-up (rezultatet do të jenë të ngjashme edhe për pen-down vetëm se me funksion invers ndaj lëvizjes pen-up).

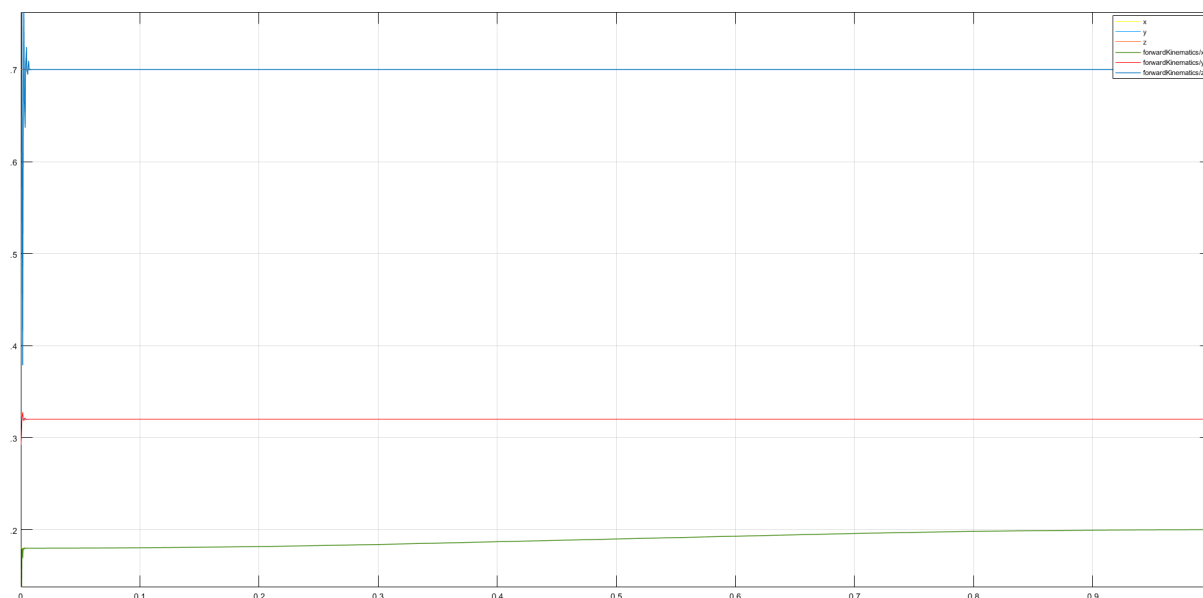


Figura 3.15: Rezultati i eksperimentit për trajektore të lëvizjes pen-up brenda 1s

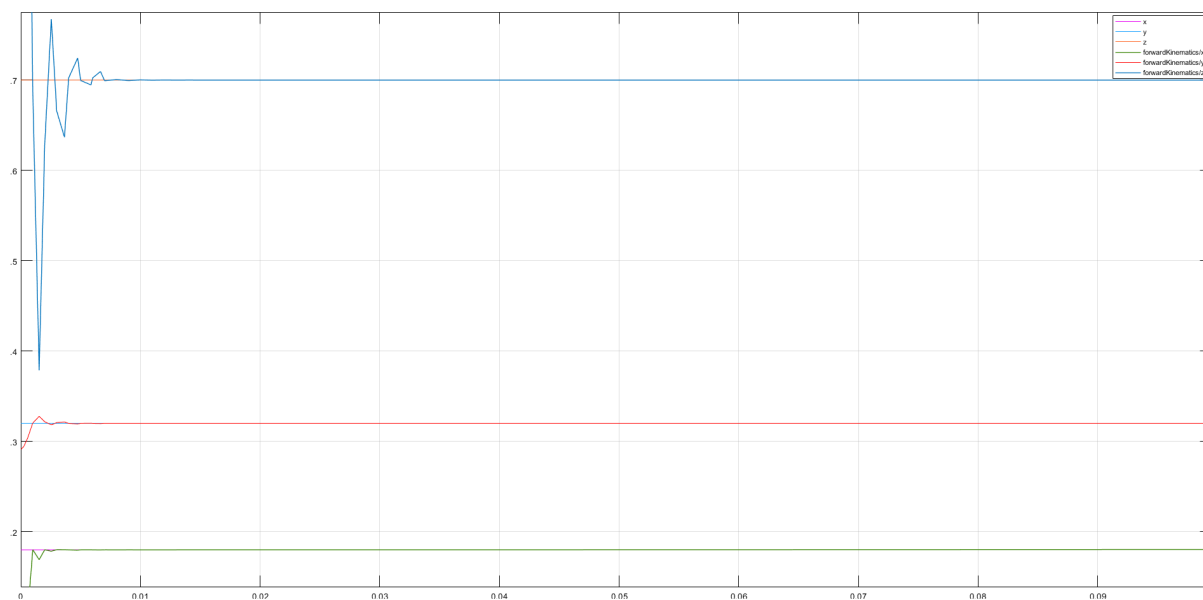


Figura 3.16: Rezultati i eksperimentit për trajektore të lëvizjes pen-up brenda 10ms

Në Figurën 3.15 shihet qartë se lëvizja e lapsit bëhet vetëm në drejtim të boshtit X pasi që lakorja e vetme që ndrron pozitë nga 0.18m në 0.20m pra shënuesi zhvendoset përpara për 2 cm që të mundësohet shkrimi në rrafshin e caktuar. Siç shihet edhe te Figura 3.16 koha e qetësimit për pozitën në fjalë është më e vogël se 0.005s, ku gjatë simulimit nuk është e dukshme kjo lëvizje nëse simulimi kryhet për kohë normale (jo të ngadalsuar).

### Analiza e rregullatorit PID me vlera të ndryshme të $K_p$ , $K_i$ , $K_d$ për trajektoren në formë të shkronjës L

Tani për trajektoren në formë të shkronjës L bashkojmë trajektoret paraprake në një rradhë të caktuar duke krijuar kshtu shkronjën L e cila përbëhet prej lëvizjeve pen-down, vertikale, horizontale, dhe në fund pen-up. Prandaj lakoret për koordinatat x,y,z do të duken sikur në Figurën 3.17 dhe 3.18.

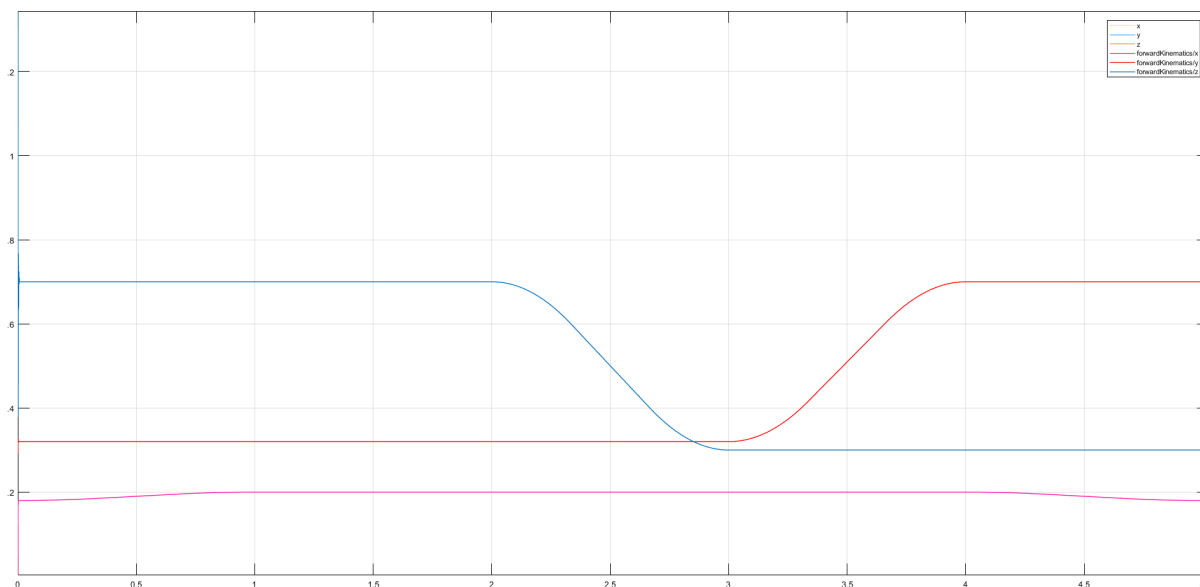


Figura 3.17: Rezultati i eksperimentit për trajektoren në formë të shkronjës L brenda 5s

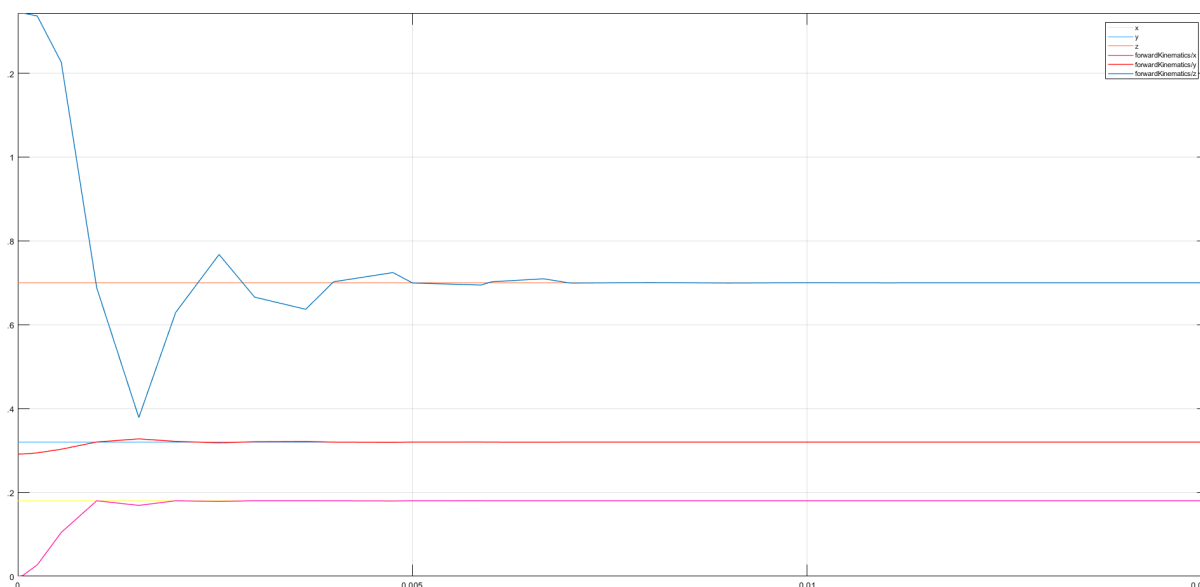


Figura 3.18: Rezultati i eksperimentit për trajektoren në formë të shkronjës L brenda 10ms

Po ashtu edhe këtu shihet se koha e qetësimit fillestar është më e vogël se 0.008s dhe pjesa tjetër e lakoreve do të përputhet gjithmonë me lakoret e dëshiruara.

# Kapitulli 4

## Konkluzioni

Në këtë tezë, ne kemi ndërmarrë një analizë gjithëpërfshirëse të dizajnit të një roboti shkrimi me katër shkallë lirie (4DOF). Pavarësisht ekzistencës së shumë robotëve të tjerë të shkrimit, shkathtësia dhe thjeshtësia e një roboti 4DOF theksohen si pikat e tij kryesore të forta. Gjatë gjithë këtij hulumtimi, ne zbatuam të gjithë procesin e simulimit, duke filluar me planifikimin e trajektores duke përdorur profilin e shpejtësisë trapezoidale, e ndjekur nga zhvillimi i kinematikës inverse, kinematikës direkte, dinamikës së robotit si dhe rregullimin e lëvizjes përmes rregullatorit PID. Vëzhgimet tona nga simulimi tregojnë se kinematika inverse, kinematika direkte dhe dinamika e robotit janë llogaritur me saktësi, duke demonstruar potencialin e tyre për përdorim në krijimin e modelit fizik të robotit.

Një nga kontributet e rëndësishme të kësaj teze është sqarimi i të gjithë procedurës për dizajnimin e një modeli roboti në një aplikacion 3D CAD dhe importimin e tij në MATLAB/Simulink për simulim të mëtejshëm. Kjo procedurë ofron një kuadër të vlefshëm që mund të aplikohet në sisteme të tjera mekanike, duke lehtësuar integrimin e modelimit CAD me simulimin dinamik, një proces ky që është thelbësor në projektimin dhe zhvillimin e sistemeve të ndryshme robotike.

Shkathtësia dhe thjeshtësia e robotit të shkrimit 4DOF të paraqitur në këtë tezë hap rrugë të reja për aplikimin e tij në fusha të ndryshme, duke përfshirë industrinë, arsimin dhe teknologjinë ndihmëse. Dizajni 4DOF jo vetëm që thjeshton strukturën mekanike, por gjithashtu rrit lehtësinë e kontrollit dhe përshtatshmërisë, duke e bërë atë një zgjidhje ideale për një gamë të gjerë detyrash shkrimi.

Simulimi u krye në një mjedis të kontrolluar dhe rregullatori PID u zbatua për të kontrolluar pozicionin e secilës nyje. Objektivi ishte të vëzhgohej se sa mirë rregullatori PID rregullon nyjet e robotit për të gjurmuar me saktësi trajektorët e dëshiruara.

Zbatimi i suksesshëm i planifikimit të trajektores, kinematikës inverse, kinematikës direkte, dhe dinamikës së robotit hap rrugën për kërkime dhe zhvillim të mëtejshëm në këtë fushë. Puna e ardhshme mund të përfshijë realizimin fizik dhe testimin e manipuluesit të robotit, si dhe zhvillimin e algoritmeve të avancuara të kontrollit për të përmirësuar më tej performancën e tij.

Për më tepër, metodologjia e përshkruar në këtë tezë ofron një bazë për kërkime të ardhshme në projektimin dhe simulimin e sistemeve robotike. Duke kombinuar modelimin CAD me

simulimin dinamik, inxhinierët dhe studiuesit mund të zhvillojnë dhe optimizojnë sisteme komplekse mekanike në mënyrë efikase dhe efektive, duke kontribuar në avancimin e teknologjisë robotike.

Si përfundim, kjo tezë paraqet një hap të rëndësishëm përpara në projektimin dhe simulimin e një roboti shkrimi me 4DOF. Vështrimet e marra nga ky hulumtim kanë potencialin të ndikojnë në zhvillimin e sistemeve robotike të ardhshme, duke theksuar rëndësinë e integritetit të modelimit CAD me simulimin dinamik për dizajnimin dhe optimizimin efikas të sistemeve mekanike.

# Kapitulli 5

## Puna në të ardhmen

Derisa kjo tezë ka dhënë një kontribut të rëndësishëm në dizajnimin dhe simulimin e një roboti shkrimi me 4DOF, ka disa fusha që ofrojnë mundësi për kërkime dhe përmirësime në të ardhmen. Më poshtë janë drejtimet e mundshme kërkimore që mund të ndiqen në tezat ose projektet e mëvonshme:

- **Gjenerimi automatik i pikave të rrugës**

Një domen për kërkime të ardhshme mund të përfshijë zhvillimin e një algoritmi që gjeneron automatikisht pikat e rrugës për secilën shkronjë, si dhe fut hapësirat mes shkronjave duke përdorur teknikën e stilolapsit stilolapsi lart-stilolapsi poshtë (pen up - pen down). Ky automatizim do të thjeshtonte procesin e shkrimit, duke eliminuar nevojën për gjenerim manual të pikave dhe hapësirave. Zbatimi i një algoritmi të tillë mund të rrisë ndjeshëm efikasitetin dhe autonominë e robotit të shkrimit, duke e bërë atë më praktik dhe të adaptueshëm për aplikacione të ndryshme.

- **Shkallëzimi dhe zbatimi i fontit**

Një projekt tjetër i mundshëm mund të përfshijë shkallëzimin e shkronjave në një madhësi të caktuar dhe zbatimin e shkronjave të ndryshme. Duke shtuar këtë veçori, roboti i shkrimit mund të përshtatet për të shkruar në stile dhe madhësi të ndryshme, duke e bërë atë më të gjithanshëm dhe të përshtatshëm për një gamë më të gjerë aplikimesh. Ky projekt do të përfshinte zhvillimin e algoritmeve për shkallëzimin dhe zbatimin e fonteve, duke rritur fleksibilitetin dhe opsionet e personalizimit të robotit të shkrimit.

- **Algoritmi i shkrimit kursiv (me shkronja të shkrimit)**

Një rrugë emocionuese për kërkimet e ardhshme do të ishte zhvillimi i një algoritmi që i mundëson robotit të shkruajë në kursive. Shkrimi kursiv paraqet sfida unike për shkak të rrjedhës së vazhdueshme të shkronjave dhe nevojës për kalime pa probleme midis shkronjave. Duke përfshirë një algoritm kursiv të shkrimit, roboti i shkrimit mund të zgjerojë aftësitë e tij dhe të aplikohet në një gamë më të gjerë detyrash shkrimi, duke përfshirë shkrimin e dorës dhe replikimin e nënshkrimit.

- **Mbajtës universal për end efektorin**

Për sa i përket dizajnit, një përmirësim mund të bëhet duke shtuar një mbajtës universale të shënuesit që mund të mbajë shënues, stilolapsa, lapsa dhe madje edhe shkurtesa të ndryshme. Kjo veçori do të rriste shkathtësinë dhe përshtatshmërinë e robotit të shkrimit, duke e lejuar atë të përdorë mjete të ndryshme shkrimi sipas kërkesave

specifike të detyrës. Për më tepër, mbajtësi i vendit për shkumësin do të duhet të përfshijë një algoritëm që përshtatet me presionin e aplikuar në sipërfaqen e shkrimit, pasi shkumësay janë të brishtë dhe kërkojnë një nivel të caktuar presioni për të shkruar në mënyrë efektive.

Hulumtimet e ardhshme në këto fusha do të avancojnë ndjeshëm aftësitë dhe aplikimet e robotit të shkrimit. Këto projekte mund të ndiqen si teza apo projekte plotësuese, duke u ndërtuar mbi themelet e vendosura nga kjo tezë. Duke adresuar këto drejtime kërkimore, funksionaliteti, përshtatshmëria dhe praktika e robotit të shkrimit mund të përmirësohen shumë, duke e bërë atë një mjet më të vlefshëm në fusha të ndryshme, duke përfshirë industrinë, arsimin dhe teknologjinë ndihmëse.



# Shtesa A

## Shtesat

### A.1 Algoritmi Denavit-Hartenberg

Duke pasur parasysh një krah robotik me  $n$  shkallë lirie, ky algoritëm i cakton një sistem koordinativ ortonormal secilit link të krahut të robotit duke u bazuar tek konfigurimet të gjeometrisë të krahut të njeriut [6]. Caktimi i sistemeve të koordinatave fillon nga baza deri te end efektori i krahut të robotit [6]. Marrëdhëniet ndërmjet linqeve mund përfaqësohen nga një matricë homogjene e transformimit  $4 \times 4$  [6]. Rëndësia e kësaj detyre është të ndihmojë zhvillimin e një procedure të qëndrueshme për nxjerrjen e zgjidhjes së përbashkët [6].

- **D1. Krijoni sistemin e koordinatave bazë.** Krijoni një sistem i koordinativ ortonormal të dorës së djathtë (right-handed)  $(x_0, y_0, z_0)$  në bazë me boshtin  $z_0$  të shtrirë përgjatë boshtit të lëvizjes së nyjës 1 dhe duke u drejtuar drejt shpatullës së krahut robotik [6]. Boshtet  $x_0$  dhe  $y_0$  mund të vendosen lehtësisht dhe janë normale me boshtin  $z_0$  [6].
- **D2. Inicializoni dhe unaza.** Për çdo  $i, i = 1, \dots, n - 1$ , kryeni hapat D3 deri në D6 [6].
- **D3. Vendosja e boshtit të përbashkët.** Rreshtoni  $z_i$  me boshtin e lëvizjes (rrotullues ose rrëshqitës) i nyjes  $i + 1$ . Për robotët që kanë konfigurime krahu majtas-djathtas, Boshtet  $z_1$  dhe  $z_2$  janë të drejtuara larg nga shpatulla dhe "trungu" i krahut të robotit [6].
- **D4. Përcaktoni origjinën e sistemit të koordinatave  $i$ -të.** Gjeni origjinën e  $i$ -të të sistemit koordinativ në prerjen e boshteve  $z_i$  dhe  $z_{i-1}$  ose në prerjen e normales së përbashkët ndërmjet boshteve  $z_i$  dhe  $z_{i-1}$  dhe boshtit  $z_i$ .
- **D5. Vendosni boshtin  $x_i$ .** Vendosni  $x_i = \pm(z_{i-1} \times z_i) / \|z_{i-1} \times z_i\|$  ose përgjatë normales së përbashkët ndërmjet boshteve  $z_{i-1}$  dhe  $z_i$  kur janë paralele [6].
- **D6. Vendosni boshtin  $y_i$ .** Cakto  $y_i = +(z_i \times x_i) / \|z_i \times x_i\|$  për të plotësuar sistemin e koordinatave të dorës së djathtë. (Zgjatni boshtet  $z_i$  dhe  $x_i$  nëse është e nevojshme për hapat D9 deri në D12) [6].
- **D7. Krijoni sistemin e koordinatave të dorës.** Zakonisht nyja e  $n$ -të është një nyje rrotulluese [6]. Vendosni  $z_n$ , përgjatë drejtimit të boshtit  $z_{n-1}$  dhe me drejtim duke u larguar nga robot [6]. Vendosni  $x_n$  të tillë që të jetë normale për të dy boshtet  $z_{n-1}$  dhe  $z_n$ . Cakto  $y_n$  për të përfunduar sistemin e koordinatave të djathta [6].
- **D8. Gjeni parametrat e nyjeve dhe linqeve.** Për çdo  $i, i = 1, \dots, n$ , kryeni hapat D9 deri në D12 [6].

- **D9. Gjeni  $d_i$ .**  $d_i$  është distanca nga origjina e sistemit të koordinatave  $(i - 1)$  të tek prerja e boshtit  $z_{i-1}$  dhe boshti  $x_i$  përgjatë boshtit  $z_{i-1}$ . Ajo është variabla e nyjes nëse nyja  $i$  është prizmatike [6].
- **D10. Gjeni  $a_i$ .**  $a_i$  është distanca nga prerja e boshtit  $z_{i-1}$  dhe  $x_i$  boshti në origjinën e sistemit të koordinatave  $i$ -të përgjatë boshtit  $x_i$  [6].
- **D11. Gjeni  $\theta_i$ .**  $\theta_i$  është këndi i rrotullimit nga boshti  $x_{i-1}$  në boshtin  $x_i$  rreth boshti  $z_{i-1}$ . Është variabla e nyjes nëse nyja  $i$  është rrotulluese [6].
- **D12. Gjeni  $\alpha_i$ .**  $\alpha_i$  është këndi i rrotullimit nga boshti  $z_{i-1}$  në boshtin  $z_i$  rreth boshti  $x_i$  [6].

## A.2 Koordinatat e përgjithësuara

Koordinatat e përgjithësuara të një sistemi mekanik janë grupi minimal i parametrave të cilët mund të përcaktojnë plotësisht dhe pa mëdyshje konfigurimin e atij sistemi [18]. Ato quhen të përgjithësuara sepse nuk janë të kufizuara të jenë koordinata karteziene dhe as që kërkojnë të maten nga një inercial kornizë referimi [18]. Megjithatë, ato përdoren për të shprehur energjitë kinetike dhe potenciale të trupat e ngurtë që përbëjnë sistemin dinamik. Energjia kinetike dhe potenciale duhet të llogaritet në lidhje me një kornizë referimi inerciale [18]. Prandaj i përgjithësuar koordinatat duhet të jenë në gjendje të shprehin shpejtësinë dhe zhvendosjen e trupave të ngurtë me në lidhje me një kornizë inerciale të referencës [18].

## A.3 Sistemet holonomike

Një sistem holonomik është ai në të cilin numri i koordinatave e përgjithësuara të pavarura të nevojshme për të përshkruar lëvizjen e sistemit janë të barabartë me numrin e shkallëve të lirisë [18].

## A.4 Efekti i Coriolis

Efekti i Coriolisit është një fenomen fizik që ndodh kur një objekt lëviz brenda një kornize të qarkullimit. Emërtuar sipas shkencëtarit francez Gaspard-Gustave de Coriolis, i cili e përshkroi për herë të parë në fillim të shekullit të 19-të, efekti i Coriolisit shfaqet si një devijim i objekteve që lëvizin për shkak të rrotacionit të Tokës. Është një forcë e dukshme që vepron në objekte që lëvizin brenda një sistemi të rrotullimit, siç është Toka.

- **Korniza e Rrotullimit të Referencës**

Imagjinoni një objekt që lëviz në një drejtim të drejtë në sipërfaqen e një trupi që rrotullohet, siç është Toka. Ndërsa objekti lëviz në një drejtim të drejtë në raport me Tokën, një vëzhgues në Tokën e rrotulluar e sheh objektin si duke marrë një rrugë të krivuar. Kjo krivinë është efekti i Coriolisit.

- **Forca**

Forca e Coriolisit  $F_C$  është ortogonale me shpejtësinë e objektit që lëviz dhe me aksin e rrotullimit. Drejtimi i saj përcaktohet nga rregulli i dorës së djathtë. Kjo forcë shprehet me formulën:

$$F_C = -2m(\omega \times v)$$

Ku:

- $F_C$  është forca e Coriolisit
- $m$  është masa e objektit
- $\omega$  është shpejtësia këndore e kornizës që rrotullohet

• **Ndikimi në Objektet që Lëvizin**

Në jetën e përditshme, efekti i Coriolisit është më i dukshëm kur vëzhgohen objekte që lëvizin në distanca të gjata, si rruga e një raketë ose trajektoria e rrjedhave oqeanike. Për shembull, në hemisferën veriore, objektet që lëvizin devijojnë në të djathtë, ndërsa në hemisferën jugore, ato devijojnë në të majtë.

## A.5 Koordinatat homogjene

Koordinatat homogjene  ${}^i r_i$  shprehen përmes vektorit të më poshtëm:

$${}^i r_i = \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} = [x_i \ y_i \ z_i \ 1]^T \quad (\text{A.1})$$

## A.6 Matricat ndihmëse $Q_i$

Matricat ndihmëse  $Q_i$  lehtësojnë kalkulimin e derivatit parcial të matricave  ${}^0 A_i$ , varësisht prej llojit të nyjeve kemi këto matrica ndihmëse:

1. Për nyje rrotulluese:

$$Q_i = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{A.2})$$

2. Për nyje prismatike (rrëshqitëse):

$$Q_i = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{A.3})$$

Pra, funksionin parcial të matricave  ${}^0 A_i$  mund ta shkruajmë si:

$$\frac{\partial {}^{i-1} A_i}{\partial q_i} = Q_i {}^{i-1} A_i \quad (\text{A.4})$$

Kurse  $\frac{\partial {}^{i-1} A_i}{\partial q_i}$  mund të shkruhet si  $U_{ji}$ :

$$U_{ji} = \begin{cases} {}^0 A_j Q_j {}^{j-1} A_i & j \leq i \\ 0 & j > i \end{cases} \quad (\text{A.5})$$

## A.7 Matrica e inercionit

Momentet skalar të inercisë shfaqen si elementë në një matricë kur një sistem grimcash mbledhet në një trup të ngurtë që lëviz në hapësirën tredimensionale [6]. Kjo matricë e inercisë shfaqet në llogaritjen e momentit këndor, energjisë kinetike dhe çift rrotullues rezultues të sistemit të ngurtë të grimcave [6]. Matrica e tillë është paraqitur tek ekuacioni (A.6)

$$J_i = \int r_i r_i^T dm = \begin{bmatrix} \int x_i^2 dm & \int x_i y_i dm & \int x_i z_i dm & \int x_i dm \\ \int x_i y_i dm & \int y_i^2 dm & \int y_i z_i dm & \int y_i dm \\ \int x_i z_i dm & \int y_i z_i dm & \int z_i^2 dm & \int z_i dm \\ \int x_i dm & \int y_i dm & \int z_i dm & \int dm \end{bmatrix} \quad (\text{A.6})$$

## A.8 Llogaritja e matricës D

Këtu është kalkuluar matrica e nxitimeve D, ku së pari janë zgjidhur matricat e transformimeve  ${}^{i-1}A_i$  pastaj efektet e lëvizjes për një  $U_{ij}$ :

$${}^0A_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & d_1 + 0.2915 \\ 0 & -1 & 0 & 0.1775 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.7})$$

$${}^1A_2 = \begin{bmatrix} c_1 & -s_1 & 0 & lc_1 \\ s_1 & c_1 & 0 & ls_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.8})$$

$${}^2A_3 = \begin{bmatrix} c_2 & -s_2 & 0 & lc_2 \\ s_2 & c_2 & 0 & ls_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.9})$$

$${}^0A_2 = \begin{bmatrix} c_1 & -s_1 & 0 & -lc_1 \\ 0 & 0 & 1 & d_1 + 0.2915 \\ -s_1 & -c_1 & 0 & ls_1 + 0.1775 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.10})$$

$${}^1A_3 = \begin{bmatrix} c_{12} & -s_{12} & 0 & l(c_{12} + c_1) \\ s_{12} & c_{12} & 0 & l(s_{12} + s_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.11})$$

$${}^0A_3 = \begin{bmatrix} c_{12} & -s_{12} & 0 & -l(c_{12} + c_1) \\ 0 & 0 & 1 & d_1 + 0.2915 \\ -s_{12} & -c_{12} & 0 & l(s_{12} + s_1) + 0.1775 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.12})$$

Matricat  $Q_i$  janë:

$$Q_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{A.13})$$

$$Q_2 = Q_3 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{A.14})$$

Kurse koeficientët e lëvizjes për nyje  $U_{ij}$ :

$$U_{11} = Q_1^0 A_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{A.15})$$

$$U_{21} = Q_1^0 A_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{A.16})$$

$$U_{31} = Q_1^0 A_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{A.17})$$

$$U_{22} = {}^0A_1 Q_2^1 A_2 = \begin{bmatrix} -s_1 & -c_1 & 0 & -ls_1 \\ 0 & 0 & 0 & 0 \\ -c_1 & s_1 & 0 & -lc_1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{A.18})$$

$$U_{32} = {}^0A_1 Q_3^1 A_3 = \begin{bmatrix} -s_{12} & -c_{12} & 0 & -l(s_1 + s_{12}) \\ 0 & 0 & 0 & 0 \\ -c_{12} & s_{12} & 0 & -l(c_1 + c_{12}) \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{A.19})$$

$$U_{33} = {}^0A_2 Q_3^2 A_3 = \begin{bmatrix} -s_{12} & -c_{12} & 0 & -ls_{12} \\ 0 & 0 & 0 & 0 \\ -c_{12} & s_{12} & 0 & -lc_{12} \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{A.20})$$

Matricat e inercionit janë:

$$J_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{3}m_1 l_1^2 & -(\frac{1}{2})m_1 l_1 \\ 0 & 0 & -\frac{1}{2}m_1 l_1 & m_1 \end{bmatrix} \quad (\text{A.21})$$

$$J_2 = \begin{bmatrix} \frac{1}{3}m_2 l_2^2 & 0 & 0 & -(\frac{1}{2})m_2 l_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{2}m_2 l_2 & 0 & 0 & m_2 \end{bmatrix} \quad (\text{A.22})$$

$$J_3 = \begin{bmatrix} \frac{1}{3}m_3 l_3^2 & 0 & 0 & -(\frac{1}{2})m_3 l_3 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{2}m_3 l_3 & 0 & 0 & m_3 \end{bmatrix} \quad (\text{A.23})$$

Për arsye të thjeshtimit të kalkulimeve në vazhdim për  $l_2$  dhe  $l_3$  marrim  $l = l_2 = l_3$ . Tani zgjidhim

$$\text{matricën } D = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{12} & D_{22} & D_{23} \\ D_{13} & D_{23} & D_{33} \end{bmatrix} :$$

$$\begin{aligned} D_{11} &= Tr(U_{11}J_1U_{11}^T) + Tr(U_{21}J_2U_{21}^T) + Tr(U_{31}J_3U_{31}^T) \\ &= Tr \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & m_1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + Tr \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & m_2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + Tr \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & m_3 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ &= m_1 + m_2 + m_3 \end{aligned} \quad (A.24)$$

$$\begin{aligned} D_{12} &= Tr(U_{22}J_2U_{21}^T) + Tr(U_{32}J_3U_{32}^T) \\ &= Tr \begin{bmatrix} 0 & 0 & \frac{-lm_2s_1}{2} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-c_1lm_2}{2} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + Tr \begin{bmatrix} 0 & 0 & \frac{-c_{12}lm_3-2lm_3s_1-2lm_3s_{12}}{2} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{2c_1lm_3+3c_{12}lm_3}{2} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ &= -\frac{1}{2}lm_2c_2 + c_1lm_3 + \frac{3}{2}lm_3c_{12} \end{aligned} \quad (A.25)$$

$$D_{13} = Tr(U_{33}J_3U_{31}^T) = Tr \begin{bmatrix} 0 & 0 & \frac{-c_{12}lm_3-2lm_3s_{12}}{2} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{3c_{12}lm_3}{2} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \frac{3}{2}lm_3c_{12} \quad (A.26)$$

$$\begin{aligned} D_{22} &= Tr(U_{22}J_2U_{21}^T) + Tr(U_{32}J_3U_{32}^T) = Tr \begin{bmatrix} \frac{l^2m_2s_1^2}{3} & 0 & \frac{c_1l^2m_2s_1}{3} & 0 \\ 0 & 0 & 0 & 0 \\ \frac{c_1l^2m_2s_1}{3} & 0 & \frac{c_1^2l^2m_2}{3} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ &+ Tr \begin{bmatrix} \frac{c_{12}^2l^2m_3+3l^2m_3s_1^2+3c_{12}l^2m_3s_1+3l^2m_3s_{12}^2+3c_{12}l^2m_3s_{12}+6l^2m_3s_1s_{12}}{3} & 0 & X_1 & 0 \\ 0 & 0 & 0 & 0 \\ X_2 & 0 & \frac{3c_1^2l^2m_3+c_{12}^2l^2m_3+3c_1c_{12}l^2m_3}{3} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ &= \frac{1}{3}l_2m_2s_1^2 + \frac{1}{3}l^2m_2c_1^2 + \frac{1}{3}l^2m_3c_{12}^2 + l^2m_3s_1^2 + l^2m_3s_1c_{12} + l^2m_3s_{12}^2 + l^2m_3s_{12}c_{12} \\ &\quad + 2l^2m_3s_1s_{12} + l^2m_3c_1^2 + \frac{1}{3}l^2m_3c_{12}^2 + l^2m_3c_1c_{12} \end{aligned} \quad (A.27)$$

Ku:

$$\begin{aligned} X_1 &= \frac{c_{12}^2l^2m_3 + 3c_1c_{12}l^2m_3 + 6c_1l^2m_3s_1 + 3c_{12}l^2m_3s_1 + 6c_1l^2m_3s_{12} + 3c_{12}l^2m_3s_{12}}{6} \\ X_2 &= \frac{c_{12}^2l^2m_3 + 3c_1c_{12}l^2m_3 + 6c_1l^2m_3s_1 + 3c_{12}l^2m_3s_1 + 6c_1l^2m_3s_{12} + 3c_{12}l^2m_3s_{12}}{6} \end{aligned}$$

$$\begin{aligned}
 D_{23} &= Tr(U_{33}J_3U_{32}^T) \\
 &= Tr \begin{bmatrix} \frac{2c_{12}^2l^2m_3+3c_{12}l^2m_3s_1+6l^2m_3s_{12}^2+6c_{12}l^2m_3s_{12}+6l^2m_3s_1s_{12}}{6} & 0 & \frac{c_{12}^2l^2m_3+3c_1c_{12}l^2m_3+6c_1l^2m_3s_{12}+3c_{12}l^2m_3s_{12}}{6} & 0 \\ 0 & 0 & 0 & 0 \\ \frac{c_{12}^2l^2m_3+3c_{12}l^2m_3s_1+3c_{12}l^2m_3s_{12}}{6} & 0 & \frac{2c_{12}^2l^2m_3+3c_1c_{12}l^2m_3}{6} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
 &= \frac{1}{3}l^2m_3c_{12}^2 + \frac{1}{2}l^2m_3s_1c_{12} + l^2m_3s_{12}^2 + l^2m_3s_{12}c_{12} + l^2m_3s_1s_{12} + \frac{1}{3}l^2m_3c_{12}^2 + \frac{1}{2}l^2m_3c_1c_{12} \\
 &\quad (A.28)
 \end{aligned}$$

$$\begin{aligned}
 D_{33} &= Tr(U_{33}J_3U_{33}^T) = Tr \begin{bmatrix} \frac{c_{12}^2l^2m_3+3l^2m_3s_{12}^2+3c_{12}l^2m_3s_{12}}{3} & 0 & \frac{c_{12}^2l^2m_3+3c_{12}l^2m_3s_{12}}{6} & 0 \\ 0 & 0 & 0 & 0 \\ \frac{c_{12}^2l^2m_3+3c_{12}l^2m_3s_{12}}{6} & 0 & \frac{c_{12}^2l^2m_3}{3} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
 &= \frac{2}{3}l^2m_3c_{12}^2 + l^2m_3s_{12} + l^2m_3c_{12}s_{12} \\
 &\quad (A.29)
 \end{aligned}$$

Ku  $c_1 = \cos(\theta_1^*)$ ,  $s_1 = \sin(\theta_1^*)$ ,  $c_{12} = \cos(\theta_1^* + \theta_2)$ ,  $s_{12} = \sin(\theta_1^* + \theta_2)$  dhe  $\theta_1^* = \theta_1 + \frac{\pi}{2}$

## A.9 Llogaritja e elementeve të gravitetit

$$\begin{aligned}
 c_1 &= -m_1 \begin{bmatrix} 0 & 0 & -g & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -\frac{l_1}{2} \\ 1 \end{bmatrix} \\
 -m_2 &\begin{bmatrix} 0 & 0 & -g & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -\frac{l}{2} \\ 1 \end{bmatrix} \\
 -m_3 &\begin{bmatrix} 0 & 0 & -g & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -\frac{l}{2} \\ 1 \end{bmatrix} \\
 &= 0
 \end{aligned} \quad (A.30)$$

$$\begin{aligned}
 c_2 &= -m_2 \begin{bmatrix} 0 & 0 & -g & 0 \end{bmatrix} \begin{bmatrix} -s_1 & -c_1 & 0 & -ls_1 \\ 0 & 0 & 0 & 0 \\ -c_1 & s_1 & 0 & -lc_1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -\frac{l}{2} \\ 1 \end{bmatrix} \\
 -m_3 &\begin{bmatrix} 0 & 0 & -g & 0 \end{bmatrix} \begin{bmatrix} -s_{12} & -c_{12} & 0 & -l(s_{12}+s_1) \\ 0 & 0 & 0 & 0 \\ -c_{12} & s_{12} & 0 & -l(c_{12}+c_1) \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -\frac{l}{2} \\ 1 \end{bmatrix} \\
 &= -\frac{1}{2}m_1glc_1 - \frac{1}{2}m_2glc_{12} - m_2glc_1 \\
 &\quad (A.31)
 \end{aligned}$$

$$c_3 = -m_3 \begin{bmatrix} 0 & 0 & -g & 0 \end{bmatrix} \begin{bmatrix} -s_{12} & -c_{12} & 0 & -ls_{12} \\ 0 & 0 & 0 & 0 \\ -c_{12} & s_{12} & 0 & -lc_{12} \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -\frac{l}{2} \\ 1 \end{bmatrix} \quad (A.32)$$

$$= -glm_3c_{12}$$

## A.10 Koeficienti filtrues tek PID rregullatori

E meta e veprimit të derivatit është se derivati ideal ka fitim shumë të lartë për sinjalet me frekuencë të lartë [19]. Kjo do të thotë se zhurma e matjes me frekuencë të lartë do të gjenet erojë oscilime të mëdha të sinjalit të kontrollit [19].

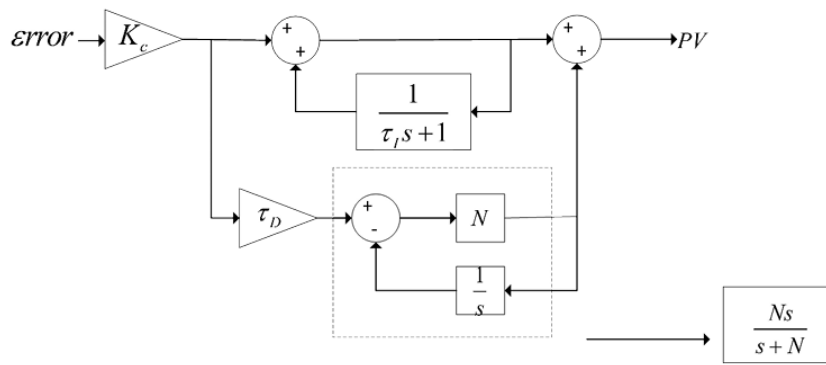


Figura A.1: PID rregullatori me koeficient filtrues [19]

## A.11 Diskretizimi i PID rregullatorit

Meqenëse kompjuterët dhe mikrokontrolluesit nuk mund të merren me kohën e vazhdueshme, ligji i kontrollit duhet të diskretizohet [20]. Për diskretizim na duhet një kohë e mostrimit  $T_s$  e cila do të ketë vlerën  $1ms$ .

### Diskretizimi i pjesës integrale me metodën direkte të Euler-it

Metoda e parë e diskretizimit që do t'i hedhim një vështrim është metoda direkte e Euler-it, është një nga metodat më të thjeshta të disponueshme për të përafëruar një ekuacion diferencial të zakonshëm në kohë të vazhdueshme me një ekuacion të diferencës në kohë diskrete ose relacion përsëritjeje [20].

Kur koha e mostrimit  $T_s$  është mjaft e vogël atëherë pjesa integrale e PID rregullatorit për kohë  $t = kT_s$  mund të llogaritet përmes shumave të Reimann-it.

$$e_i(t) = \int_0^t e(\tau) d\tau \approx \sum_{n=0}^{k-1} e[n]T_s = e_i[k] \quad (A.33)$$

Vini re se kjo është një përafërim,  $e_i(kT_s) \approx e_i[k]$ , nuk janë saktësisht të barabarta [20].



Sinjali  $e_i[k]$  mund të definohet nga relacioni i shprehur më poshtë:

$$\begin{cases} e_i[k] = e_i[k-1] + e_i[k-1]T_s \\ e_i[0] = 0 \end{cases} \quad (\text{A.34})$$

Në domenin  $z$ , diskretizimi direkt i Euler-it që kemi kryer në paragrafin e mëparshëm mund të shprehet si:

$$E_i(z) = z^{-1}E_i(z) + T_s z^{-1}E(z) \quad (\text{A.35})$$

$$E_i(z) = \frac{T_s}{z-1}E(z) \quad (\text{A.36})$$

Ne mund të përkufizojmë diskretizimin me metodë direkte të Euler-it si pasqyrim nga domena  $s$  në domenin  $z$  ku:  $s \rightarrow \frac{z-1}{T_s}$

### Diskretizimi i pjesës derivative

Edhe këtu mund të përafërsojmë termin derivativ të PID rregullatorit duke përdorur diferencën e fundme.

$$e_d(t) = \frac{d}{dt}e_f(t) \approx \frac{e_f(t) - e_f(t - T_s)}{T_s} = e_d[k] \quad (\text{A.37})$$

Në domenin  $z$  do të kemi ekuivalentin:

$$E_d(z) = \frac{1 - z^{-1}}{T_s}E_f(z) \quad (\text{A.38})$$

$$E_d(z) = \frac{z-1}{zT_s}E_f(z) \quad (\text{A.39})$$

Prandaj ne mund të caktojmë pasqyrimin e domenës  $s$  në domenin  $z$  përmes ekuacionit (A.13):  
 $s \rightarrow \frac{z-1}{zT_s}$

# Bibliografia

- [1] Onshape. *Onshape Documentation*. [https://cad.onshape.com/help/Content/EnterpriseHelp/Content/home.htm#\\_ga=2.15660790.1372310290.1713807301-1973486785.1713807301](https://cad.onshape.com/help/Content/EnterpriseHelp/Content/home.htm#_ga=2.15660790.1372310290.1713807301-1973486785.1713807301), Accessed on 18th April 2024. 2024.
- [2] Matlab. *What is Matlab?* <https://www.mathworks.com/discovery/what-is-matlab.html>, Accessed on 18th April 2024. 2024.
- [3] Matlab. *Simscape MultiBody*. <https://www.mathworks.com/products/simscape-multibody.html>, Accessed on 18th April 2024. 2024.
- [4] MathWors. *Convert CAD Assembly into Simscape Multibody Model*. <https://www.mathworks.com/help/sm/ug/cad-translation.html>, Accessed on 18th April 2024. 2024.
- [5] Kevin M. Lynch **and** Frank C. Park. *MODERN ROBOTICS: MECHANICS, PLANNING, AND CONTROL*. Cambridge University Press, 2017.
- [6] R. C. Gonzalez K. S. Fu **and** C. S. G. Lee. *ROBOTICS: Control, Sensing, Vision, and Intelligence*. McGraw-Hill Book Company, 1987.
- [7] J.J. Crago. *Introduction to Robotics Mechanics and Contro*. Prentice Hall, 2005.
- [8] Gautam Chettiar. *DH-Table-Solver*. <https://github.com/chettiargautam/DH-Table-Solver>, Accessed on 18th April 2024. 2022.
- [9] Frank Dellaert. *Inverse Kinematics*. [https://dellaert.github.io/20S-3630/Slides/L12\\_Inverse\\_Kinematics.pdf](https://dellaert.github.io/20S-3630/Slides/L12_Inverse_Kinematics.pdf), Accessed on 18th April 2024. 2020.
- [10] Kris Hauser. *Robotic Systems*. <https://motion.cs.illinois.edu/RoboticSystems/InverseKinematics.html>, Accessed on 18th April 2024. 2023.
- [11] Guanqiao Shan Changsheng Dai **and** Yu Sun. *Robotics for Cell Manipulation and Characterization*. ScienceDirect, 2023.
- [12] Kris Hauser. *Electricity - Magnetism*. <https://www.electricity-magnetism.org/piezoresistive-force-sensor/>, Accessed on 18th April 2024. 2024.
- [13] Serdar Küçük. *Multi-Robot Systems - New Advances*. Artificial Intelligence. IntechOpen, 2023.
- [14] Zixing C **and** Bin X. *Robotics. 3rd ed*. Tsinghua Press, 2015.
- [15] Farid Golnaraghi Benjamin C. Kuo. *Automatic Control Systems 10th ed*. McGraw-Hill Education, 2017.
- [16] James Teow. *Understanding Robot Motion: PID Control*. <https://medium.com/@jaems33/understanding-robot-motion-pid-control-8931899c31df>, Accessed on 19th April 2024. 2018.

- [17] Kevin McAleer. *PID Controllers*. [https://www.kevsrobots.com/resources/how\\_it\\_works/pid-controllers.html](https://www.kevsrobots.com/resources/how_it_works/pid-controllers.html), Accessed on 19th April 2024. 2023.
- [18] J. Kim Vandiver. *AN INTRODUCTION TO LAGRANGE EQUATIONS*. [https://ocw.mit.edu/courses/2-003sc-engineering-dynamics-fall-2011/resources/mit2\\_003scf11\\_lagrange/](https://ocw.mit.edu/courses/2-003sc-engineering-dynamics-fall-2011/resources/mit2_003scf11_lagrange/), Accessed on 18th April 2024. 2011.
- [19] vshankar. *PID Controller implementation in Simulink*. [https://home.iitk.ac.in/~vshankar/che381/files/PID\\_Description.pdf](https://home.iitk.ac.in/~vshankar/che381/files/PID_Description.pdf), Accessed on 20th April 2024. 2014.
- [20] Pieter P. *PID Controllers*. <https://tttapa.github.io/Pages/Arduino/Control-Theory/Motor-Fader/PID-Controllers.html>, Accessed on 20th April 2024. 2023.