Sučelje čovjeka i robota zasnovano na preslikavanju pokreta

Sorić, Petar

Master's thesis / Diplomski rad

2025

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:168:628847

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2025-03-26**

Repository / Repozitorij:

FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repozitory

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 704

SUČELJE ČOVJEKA I ROBOTA ZASNOVANO NA PRESLIKAVANJU POKRETA

Petar Sorić

Zagreb, veljača 2025.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 704

SUČELJE ČOVJEKA I ROBOTA ZASNOVANO NA PRESLIKAVANJU POKRETA

Petar Sorić

Zagreb, veljača 2025.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 30. rujna 2024.

DIPLOMSKI ZADATAK br. 704

Pristupnik:	**Petar Sorić (0036525427)**
Studij:	Računarstvo
Profil:	Znanost o podacima
Mentor:	izv. prof. dr. sc. Matko Orsag

Zadatak: **Sučelje čovjeka i robota zasnovano na preslikavanju pokreta**

Opis zadatka:

U sklopu diplomskog zadatka potrebno je razviti programsko rješenje i pripadajuće elektromehaničke komponente kojim će se omogućiti demonstracijom podržano programiranje robota s pokretnom bazom. Potrebno je pregledati i analizirati postojeću literaturu u području demonstracijom podržanog programiranja robotskih sustava. Odabranu metodu potrebno je prilagodit i implementirati na robotskom sustavu s pokretnom bazom. Potrebno je prikupiti podatke o demonstraciji, obraditi ih i prilagoditi kinematičkim i dinamičkim ograničenjima odabranog robotskog sustava. Sustav je potrebno ispitati u simulacijskom i laboratorijskom okruženju.

Rok za predaju rada: 14. veljače 2025.

Prije svega htio bi se zahvaliti mentoru, prof. dr.sc. Matku Orsagu, na velikom znanju koje je kroz posljednjih nekoliko godina prenio na mene i na pomoći koju je uvijek pružao. Također se zahvaljujem asistentu i budućem doktoru znanosti univ. mag. ing. el. techn. inf. Filipu Zoriću koji je pomogao u testiranju programskog rješenja. Zahvaljujem se mojoj obitelji i prijateljima, koji su me podržavali tijekom cijelog studija.

Sadržaj

1. Uvod 3

1.1. Struktura rada 3

2. Pregled područja 4

3. Teorijska osnova 5

3.1. Rotacija koordinatnih sustava 5

3.2. Elementarne matrice rotacije 6

3.3. Translacija koordinatnih sustava 7

3.4. Skaliranje koordinatnih sustava 7

3.5. Homogena matrica transformacije 8

3.6. Kvaternioni 9

3.6.1. Osnovne operacije s kvaternionima 9

3.7. Kinematika mobilnog robota s diferencijalnim pogonom 11

3.7.1. Unaprijedna kinematika mobilnog robota s omnidirekcionalnim pogonom 12

3.7.2. Inverzna kinematika mobilnog robota s omnidirekcionalnim pogonom 12

3.8. Kvadkopteri 13

3.9. Upravljačka arhitektura kvadkoptera 15

4. Pregled sustava i alata 16

4.1. Robotski operacijski sustav (ROS) 16

4.2. Sustav za virtualizaciju (Docker) 16

4.3. Godot Engine 17

4.4. Programsko okruženje za uređaj virtualne stvarnosti 17

4.5. SCOUT mobilni robot 17
4.6. Kvadkopter 18
4.7. Arhitektura sustava 19
4.8. Korisničko sučelje 20
4.9. Načini upravljanja 20
4.10. Komunikacijski sustavl 21
4.11. Prikaz oblaka točaka u virtualnoj stvarnosti 22
4.12. Preslikavanje pokreta 22
4.12.1. Generiranje naredbi, ažuriranje pozicija 23

5. Eksperimentalni rezultati 25
5.1. Upravljanje letjelicom u simulaciji 25
5.2. Upravljanje letjelicom kontrolerom u simulaciji 26
5.3. Upravljanje letjelicom pokretom ruku u simulaciji 28
5.4. Upravljanje Scout-om u stvarnom okruženju 29

6. Zaključak 32

Literatura 34

Sažetak 37

Abstract 38

1. Uvod

U današnjem svijetu, kada roboti sve više sudjeluju u globalnoj ekonomiji i postaju uobičajeni u okruženjima koja su donedavno bila isključivo ljudska, interakcija između ljudi i robota postaje sve važnija. Razvoj sučelja između robota i čovjeka, koje će tu interakciju učiniti što jednostavnijom i intuitivnijom, predstavlja velik izazov. [1], [2] Iako je krajnji cilj robotike napraviti robote koji će autonomno rješavati ljudske probleme, trenutno takva rješenja ne postoje (izuzev industrijskih robota). U većini slučajeva, mobilni roboti, poput bespilotnih letjelica, nisu potpuno autonomni, već njima upravlja čovjek. U takvim slučajevima je bitno da čovjek može efektivno upravljati mobilnim robotom. Koristeći uređaje virtualne stvarnosti (engl. *virtual reality*, skraćeno VR) moguće je napraviti sučelja koja omogućuju da čovjek vidi simulirano okruženje robota i upravlja robotom. U ovom radu razvijeno je jedno takvo sučelje, koristeći *Meta Quest3* VR-uređaj.

1.1. Struktura rada

Ovaj rad podijeljen je u 8 cjelina, uključujući i trenutnu cjelinu.

- 2. poglavlje daje pregled relevantnih radova iz područja istraživanja.
- 3. poglavlje daje čitatelju teorijsku podlogu za bolje razumijevanje čitavog sustava.
- 4. poglavlje pruža čitatelju pregled arhitekture sustava i korištenih alata.
- 5. i 6. poglavlje analiziraju rezultate eksperimenata i opisuju dovedene zaključke.

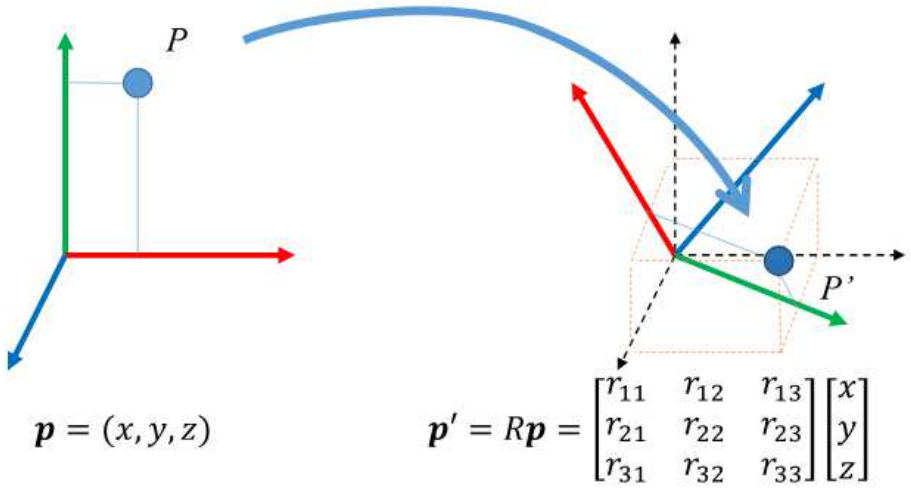
2. Pregled područja

U posljednjih pet godina područje robotike koje se bavi sučeljima čovjeka i robota doživjelo je ubrzani razvoj [3],[4],[5],[6],[7], kako zbog sve šire dostupnosti uređaja za virtualnu i proširenu stvarnost, tako i zbog ekspanzije modela dubokog učenja koji su prilagođeni raznim domenama [8] . Preslikavanje pokreta moguće je postići na nekoliko načina. Prvi se temelji na korištenju ručnih kontrolera koji dolaze uz većinu VR naočala, što ograničava preslikavanje na pokrete ruku. Drugi pristup koristi računalni vid, čime se omogućuje praćenje čitavog ljudskog tijela [9], [10].

Promatrajući postojeće sustave, [11] sugerira podjelu HRI (engl. *human-robot interaction*) sučelja u pet kategorija: (1) vizualizacija, (2) upravljanje robota i planiranje, (3) interakcija, (4) uporabljivost te (5) infrastruktura. S druge strane, noviji radovi popunjavaju niz specifičnih niša: primjerice, [12] istražuje poluautonomno planiranje i prikaz u virtualnoj stvarnosti, dok [13] razmatra optimizaciju sposobnosti upravljanja i u višeručnim teleoperacijama. U radu [14] se opisuje upotreba teleportiranja i intuitivnog teleupravljanja robota koristeći VR sučelja, a [15] i [2] sustavno obrađuju izazove i prednosti miješane stvarnosti (engl. *mixed-reality, XR*) u daljinskim interakcijama čovjeka i robota. Također vrijedi istaknuti istraživanja o suradnji ljudi i bespilotnih letjelica kroz mješovitu stvarnost [16], kao i napomene iz izvješća [1] o promjenama koje donosi interakcija čovjeka i robota na radnom mjestu. Neke od spomenutih promjena su rast produktivnosti te stvaranje kvalitetnih radnih mjesta.

3. Teorijska osnova

3.1. Rotacija koordinatnih sustava



Slika 3.1. Pomak točke P prilikom rotacije koordinatnog sustava oko točke ishodišta i pripadajuća matrična jednadžba, izvor: [17]

Rotacijske matrice u 3 dimenzije zadovoljavaju sljedeća svojstva:

- kompozicija dviju rotacijskih matrica $\mathbf{R_1 R_2}$ također je rotacijska matrica. Međutim, općenito vrijedi $\mathbf{R_1 R_2} \neq \mathbf{R_2 R_1}$, osim ako rotacije dijele istu os
- determinanta rotacijske matrice uvijek je $\det(\mathbf{R}) = \mathbf{1}$
- transponirana rotacijska matrica jednaka je njenom inverzu: $\mathbf{R^T} = \mathbf{R^{-1}}$, odnosno $\mathbf{R R^T} = \mathbf{R^T R} = \mathbf{I}$
- norme vektora ostaju nepromijenjene pod rotacijom.

Drugo tumačenje rotacijskih matrica je da svaki od 3 stupca matrice daje koordinate

osi nakon rotacije u odnosu na izvorni referentni sustav. Ako je matrica rotacije:

$$R = \begin{bmatrix} x_x & y_x & z_x \\ x_y & y_y & z_y \\ x_z & y_z & z_z \end{bmatrix} \tag{3.1}$$

tada $(\mathbf{x_x}, \mathbf{x_y}, \mathbf{x_z})$ daje koordinate nove X-osi, $(\mathbf{y_x}, \mathbf{y_y}, \mathbf{y_z})$ nove Y-osi, a $(\mathbf{z_x}, \mathbf{z_y}, \mathbf{z_z})$ nove Z-osi u starom sustavu. Ako je točka dana koordinatama $\mathbf{p} = (\mathbf{p_x}, \mathbf{p_y}, \mathbf{p_z})$, njezine nove koordinate nakon rotacije u izvornom referentnom sustavu su **Rp**.

3.2. Elementarne matrice rotacije

U trodimenzionalnom prostoru, rotacija oko osnovnih osi **X**, **Y** i **Z** može se opisati uz pomoć sljedećih matrica rotacije.

1. **Rotacija za kut θ oko X-osi:**

$$\mathbf{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \tag{3.2}$$

Matrica R_x rotira točku oko X-osi, pri čemu se mijenjaju samo **y** i **z** koordinate.

2. **Rotacija za kut θ oko Y-osi:**

$$\mathbf{R}_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{3.3}$$

Matrica R_y mijenja **x** i **z** koordinate, dok **y** ostaje nepromijenjena.

3. **Rotacija za kut θ oko Z-osi:**

$$\mathbf{R}_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

Matrica R_z utječe samo na **x** i **y** koordinate, dok **z** ostaje nepromijenjena.

3.3. Translacija koordinatnih sustava

Pretpostavimo dva koordinatna sustava: globalni koordinatni sustav G i lokalni koordinatni sustav L. Relacija između koordinata u ta dva sustava može se izraziti translacijom vektora.

Ako je vektor $\mathbf{p}_L$ u lokalnom sustavu povezan s vektorom $\mathbf{p}_G$ u globalnom sustavu, tada je odnos definiran kao:

$$\mathbf{p}_G = \mathbf{T} + \mathbf{p}_L \tag{3.5}$$

gdje je:

- $\mathbf{p}_G$ — koordinata točke u globalnom sustavu,
- $\mathbf{p}_L$ — koordinata točke u lokalnom sustavu,
- $\mathbf{T} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}$ — translacijski vektor između ishodišta globalnog i lokalnog koordinatnog sustava.

Primjerice, za 2D sustav translacija se može prikazati kao:

$$\begin{bmatrix} x_G \\ y_G \end{bmatrix} = \begin{bmatrix} T_x \\ T_y \end{bmatrix} + \begin{bmatrix} x_L \\ y_L \end{bmatrix}. \tag{3.6}$$

3.4. Skaliranje koordinatnih sustava

Ponekad je potrebno skalirati točke unutar koordinatnog sustava, kako bi se prilagodile određenim ograničenjima. U homogenoj matrici transformacije, skaliranje se obično

prikazuje dijagonalno, primjerice:

$$s = \begin{bmatrix} \alpha & 0 & 0 & 0 \\ 0 & \beta & 0 & 0 \\ 0 & 0 & \gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.7}$$

Na taj se način pojedine osi (x, y, z) zasebno skaliraju vrijednostima α, β i γ. Ovakva se matrica kasnije može kombinirati s rotacijom i translacijom u jedinstvenu homogenu transformaciju kako bi se istovremeno primijenilo više tipova transformacija.

3.5. Homogena matrica transformacije

Uzet je za primjer slučaj gdje postoje dva koordinatna sustava, koordinatni sustav svijeta i koordinatni sustav robota.

$$T = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & t_x \\ \sin\theta & \cos\theta & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.8}$$

Gornja matrica kombinira rotaciju oko osi **z** za kut θ s translacijom $\mathbf{t_x}$, $\mathbf{t_y}$, i $\mathbf{t_z}$. Proširenо za **3D** rotacije: za osi **x** i **y** treba zamijeniti odgovarajuće elemente matrice prema pravilima rotacijskih matrica za **3D** prostor.

Matrica koja se dobije množenjem elementalnih matrica rotacija $R_z(\psi)\,R_y(\theta)\,R_x(\phi)$, u slučaju kada se radi rotacija oko svake osi i translacija ishodišta:

$$T = \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi & t_x \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi & t_y \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.9}$$

Homogena matrica transformacije **T** koristi se u robotici i računalnoj geometriji za kom-

biniranje rotacija i translacija u jedinstveni sustav. Ona omogućuje prelazak između koordinatnih sustava objekata u prostoru, zadržavajući informacije o njihovoj orijentaciji i poziciji.

3.6. Kvaternioni

Kvaternioni su matematička struktura koja se koristi za predstavljanje rotacija u trodimenzionalnom prostoru. Kvaternioni su proširenje kompleksnih brojeva s četiri komponente:

$$q = w + xi + yj + zk \tag{3.10}$$

gdje su:

- w skalarni dio (realni broj),
- x, y, z imaginarni dijelovi, povezani s jedinicama i, j, k koje zadovoljavaju sljedeća pravila: $i^2 = j^2 = k^2 = ijk = -1$

3.6.1. Osnovne operacije s kvaternionima

1. **Zbrajanje:** Kvaternioni se zbrajaju komponentno:

$$q_1 + q_2 = (w_1 + w_2) + (x_1 + x_2)i + (y_1 + y_2)j + (z_1 + z_2)k \tag{3.11}$$

2. **Množenje:** Množenje kvaterniona nije komutativno, koristi se pravilo:

$$ij = k, \quad ji = -k \tag{3.12}$$

$$jk = i, \quad kj = -i \tag{3.13}$$

$$ki = j, \quad ik = -j \tag{3.14}$$

Opće pravilo za množenje dva kvaterniona $q_1 = (w_1, x_1, y_1, z_1)$ i $q_2 = (w_2, x_2, y_2, z_2)$:

$$q_1 q_2 = \begin{bmatrix} w_1 w_2 - x_1 x_2 - y_1 y_2 - z_1 z_2, \\ w_1 x_2 + x_1 w_2 + y_1 z_2 - z_1 y_2, \\ w_1 y_2 - x_1 z_2 + y_1 w_2 + z_1 x_2, \\ w_1 z_2 + x_1 y_2 - y_1 x_2 + z_1 w_2 \end{bmatrix} \tag{3.15}$$

3. **Konjugirani kvaternion:**

$$q^* = w - xi - yj - zk \tag{3.16}$$

4. **Norma kvaterniona:**

$$\|q\| = \sqrt{w^2 + x^2 + y^2 + z^2} \tag{3.17}$$

5. **Inverzni kvaternion:**

$$q^{-1} = \frac{q^*}{\|q\|^2} \tag{3.18}$$

Predstavljanje rotacije kvaternionima

Umjesto matrica ili Eulerovih kutova, rotacija se može predstaviti kvaternionom:

$$q = \cos\frac{\theta}{2} + \sin\frac{\theta}{2}(xi + yj + zk) \tag{3.19}$$

gdje je:

- θ kut rotacije,
- (x, y, z) jedinični vektor osi rotacije.

Za rotaciju vektora **v**, potrebno ga je zapisati kao čisti kvaternion:

$$v_q = 0 + v_x i + v_y j + v_z k \tag{3.20}$$

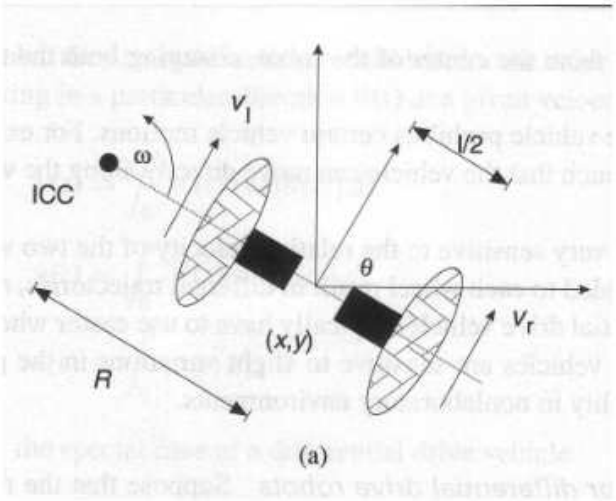
Rotacija se zatim računa kao:

$$v'_q = q v_q q^{-1} \tag{3.21}$$

Rezultat v_q' je novi vektor nakon rotacije.

3.7. Kinematika mobilnog robota s diferencijalnim pogonom

Većina mobilnih robota koristi pogonski mehanizam koji se sastoji od 2 kotača postavljenih na svaku osovinu, gdje se pojedinačni kotač može pokretati neovisno o drugom kotaču (naprijed-nazad). Uzimajući ovu pretpostavku, moguće je razviti jednadžbe gibanja za takav tip pogona. Kod pokretanja kotača, robot se nužno rotira oko središnje točke zakrivljenosti, oko koje se u određenom trenutku događa rotacija, a nalazi se na osi koja prolazi kroz središta kotača.



Slika 3.2. Prikaz diferencijalnog pogona s dva kotača

$$\omega\left(R+\frac{l}{2}\right)=V_r \tag{3.22}$$

$$\omega\left(R-\frac{l}{2}\right)=V_l \tag{3.23}$$

V_r, V_l su brzine desnog i lijevog kotača, R je udaljenost od središnje točke između kotača do ICC-a. Iz ovih formula lako se dobije R i ω

$$R=\frac{l\,(V_l+V_r)}{2\,(V_r-V_l)},\quad \omega=\frac{V_r-V_l}{l} \tag{3.24}$$

Postoje tri rubna slučaja koja su korisna za proučiti.

1. $V_L = V_R$: Robot se kreće naprijed u ravnoj liniji, trenutačno središte zakrivljenosti (ICC) nalazi se u beskonačnosti, kutna brzina je 0.

2. $V_L = -V_R$: Rotacija u mjestu jer je $R = 0$.

3. $V_L = 0, V_R \neq 0$: Robot se rotira oko lijevog kotača.

3.7.1. Unaprijedna kinematika mobilnog robota s omnidirekcionalnim pogonom

Kod svesmjernog (*omnidirekcionalnog*) pogona svaki od četiri kotača može se kontrolirati pojedinačno, što omogućuje kretanje u ravnini u svim smjerovima (komponente v_x i v_y) te rotaciju ω. Neka se kotači u lijevoj skupini okreću brzinom v_L (ili kutnom brzinom ω_L), a u desnoj skupini brzinom v_R (ili kutnom brzinom ω_R). Neka je R radijus kotača, a b udaljenost između lijevih i desnih kotača. Tada je:

$$v = \frac{v_R + v_L}{2} = \frac{R}{2}(\omega_R + \omega_L) \tag{3.25}$$

$$\omega = \frac{v_R - v_L}{b} = \frac{R}{b}(\omega_R - \omega_L) \tag{3.26}$$

Ako je konfiguracija robota opisana položajem (x, y) i orijentacijom θ, gibanje se opisuje diferencijalnim jednadžbama:

$$\dot{x} = v\cos\theta, \quad \dot{y} = v\sin\theta, \quad \dot{\theta} = \omega. \tag{3.27}$$

3.7.2. Inverzna kinematika mobilnog robota s omnidirekcionalnim pogonom

Inverzna kinematika određuje potrebne kutne brzine kotača kako bi robot postigao zadanu brzinu i orijentaciju kretanja. Potrebne kutne brzine kotača (v_x, v_y, ω), dobivaju se

koristeći inverznu relaciju:

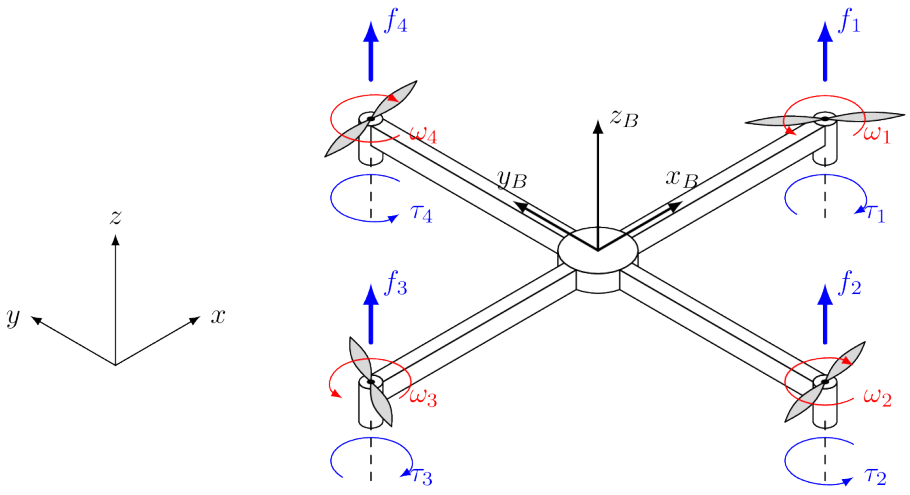
$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \frac{1}{R} \begin{bmatrix} 1 & -1 & -(l+w) \\ 1 & 1 & (l+w) \\ 1 & 1 & -(l+w) \\ 1 & -1 & (l+w) \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \tag{3.28}$$

gdje je:

- R - radijus kotača,
- l - udaljenost između središta robota i kotača u smjeru dužine,
- w - udaljenost između središta robota i kotača u smjeru širine.

3.8. Kvadkopteri

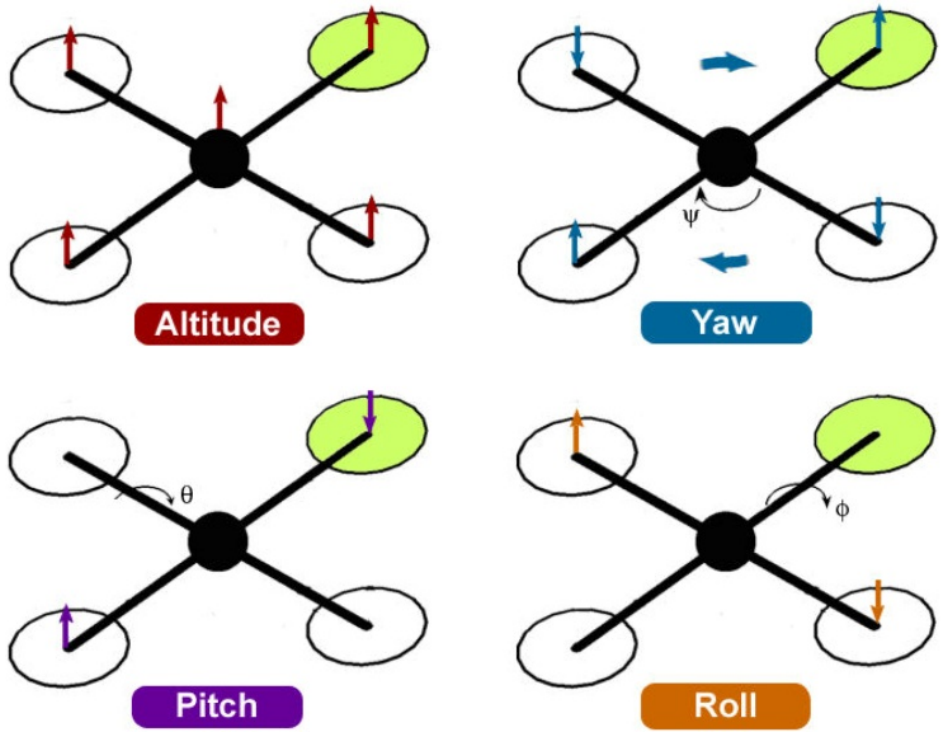
Kvadkopteri su danas najčešći oblik bespilotnih letjelica, prepoznatljivi po svoja četiri rotora. Svaki rotor može prilagođavati brzinu vrtnje propelera i time mijenjati vlastiti potisak. Iako kvadkopter ima **6** stupnjeva slobode (engl. *Degrees of freedom*, **DoF**), istovremeno se može upravljati samo s četiri, što ga čini podaktuiranim sustavom. Rotori su



Slika 3.3. Struktura kvadkoptera [18]

podijeljeni u dvije grupe, naprijed/nazad (rotori na osi x) te lijevo/desno (rotori na osi y). Rotori u istoj grupi imaju isti smjer zakretanja elisa. Svaki rotor uzrokoje potisnu silu koja djeluje u smjeru pozitivne z osi i okretni moment koji djeluje u smjeru suprotnom

od smjera zakretanja elise. Zakretanjem kvadkoptera oko pojedinačnih osi, moguće je ostvariti promjenom brzina vrtnje pojedinih rotora. Promjena kutne brzine bilo kojeg od rotora uzrokoje promjenu koju je nužno anulirati ako je cilj da letjelica zadrži svoju trenutnu orijentaciju. Primjerice, smanjenje kutne brzine rotora 1, uzrokuje smanjeni okretni moment u smjeru kazaljke na satu, što rotor 3 mora kompenzirati povećanjem svoje kutne brzine. Povećanjem kutne brzine rotora 3 kvadkopter se poćinje rotirati oko y osi jer je potisak na rotoru 1 manji od potiska na rotoru 3. Ukoliko je cilj održati trenutnu orijentaciju letjelice, smanjenje kutne brzine jednog rotora nužno podrazumijeva i smanjenje kutnih brzina drugih rotora. Promjenu visine jednostavno je napraviti, sma-



Slika 3.4. Promjene orijentacije kvadkoptera [19]

njenjem ili povećanjem brzina vrtnje svih rotora odjednom, što rezultira promjenom potiska (engl. *thrust*). Promjena nagiba, to jest zakretanje kvadkoptera oko y osi (engl. *pitch*) ostvaruje se promjenom potiska na prednjem i stražnjem rotoru, tako da jedan rotor smanjuje brzinu vrtnje dok drugi rotor povećava brzinu vrtnje, kako bi se poništio okretni moment koji je nastao smanjenjem brzine vrtnje rotora. Valjanje (zakretanje oko x osi, engl. *roll*) ostvaruje se na isti način kao i promjena nagiba, samo se u ovom slučaju mijenjaju brzine vrtnje lijevog i desnog rotora. Kod prethodnih manevara, zakretanje je ostvareno variranjem potiska s jedne strane rotacijske osi, a posljedica je promjena okretnog momenta koja se mora kompenzirati. Kako bi se kvadkopter zakrenuo oko **z**-

osi (engl. *yaw*), potrebno je promijeniti ukupan okretni moment, a ta promjena nužno podrazumijeva i promjenu potiska na nekim rotorima. Primjerice, smanji li se brzina vrtnje rotora 2 i 4, okretni moment koji gleda u smjeru kazaljke na satu će se smanjiti, što znači da ukupan okretni moment više nije nula, nego gleda u smjeru suprotnom od kazaljke na satu. Ovime je ostvarena rotacija oko z-osi, no drugi par rotora sada mora nadoknaditi izgubljeni potisak povećanjem vlastitih brzina vrntje.

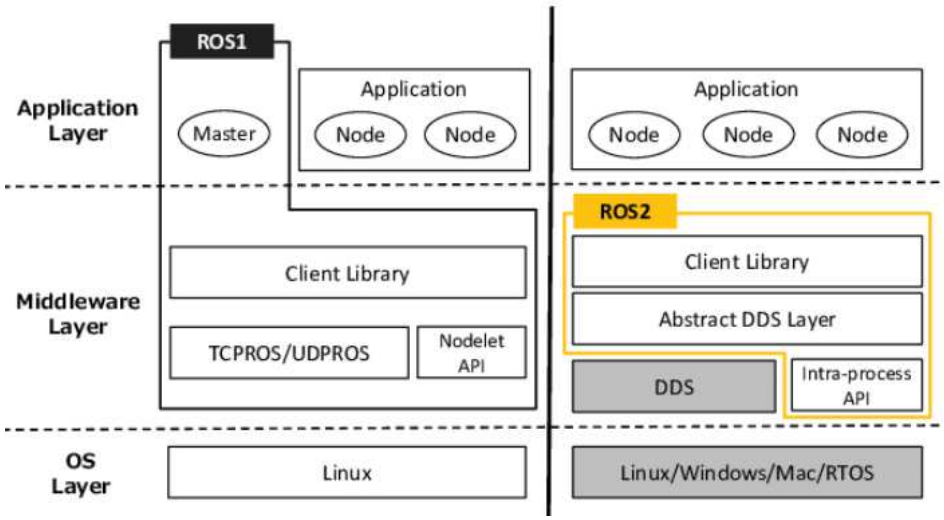
3.9. Upravljačka arhitektura kvadkoptera

Početni dio sustava, *Trajectory Planner*, generira referentnu putanju koju letjelica treba pratiti. Zatim postoji *Position Controller*, čija je zadaća usporediti željenu poziciju s aktualnom i proizvesti odgovarajuće pozicijske naredbe. *Attitude Planner* služi za planiranje orijentacije letjelice na temelju dobivenih pozicijskih podataka, vodeći računa o trenutnoj orijentaciji letjelice i drugim zahtjevima. Nadalje, *Attitude Controller* uspoređuje željene i stvarne vrijednosti orijentacije letjelice te izračunava korekcije za stabilan let. Te se korekcije zatim prosljeđuju u *Motor Controller*, koji upravlja pojedinačnim motorima kvadkoptera. *Motor Controller* prilagođava potisak svakog motora kako bi se održala željena visina, brzina i orijentacija.

4. Pregled sustava i alata

4.1. Robotski operacijski sustav (ROS)

ROS (***R**obot **O**perating **S**ystem*) je softver koji omogućuje jednostavno razvijanje aplikacija za različite vrste robota. Prednost *ROS*-a je njegova modularnost i velika ponuda gotovih paketa s otvorenim kodom, što znači da često nije potrebno raditi vlastita rješenje u cjelosti, već je dovoljno iskoristiti gotove pakete i nadograditi njihovu funkcionalnost za potrebe novog sustava. U sustavu koji je razvijen u sklopu ovog rada, korišten je *ROS* 1.



Slika 4.1. Pregled arhitekture ROS-a.

4.2. Sustav za virtualizaciju (Docker)

Docker je alat koji rješava problem distribucije softvera i postavljanje reproducibilnog razvojnog okruženja. Docker koristi izolirana okruženja (kontejner, engl. *container*, kako bi izolirao pojedinu aplikaciju od ostatka računala. Kontejneri su izolirana okruženja

koja imaju vlastiti datotečni sustav, mrežnu konfiguraciju, procese itd. Docker kontejneri se u ovom sustavu koriste kao razvojno okruženje ROS paketa za upravljanje letjelicom i mobilnim robotom.

4.3. Godot Engine

Godot je alat koji se koristi za razvoj 2D, 3D i VR igri. Olakšava razvoj igara time što proizvođači igara mogu kroz grafičko sučelje dodavati razne komponente, raditi scene, animacije, dodavati gotove modele iz drugih alata poput *blender*-a. Također je moguće dizajnirati igru sa skriptama koje mogu biti pisane u *C#* ili u nativnom skriptnom jeziku *Godot*-a, dopušteno je i raditi ekstenzije koje proširuju funkcionalnosti samog alata. Kao dio ovog sustava, razvijena je jednostavna godot ekstenzija koja dodaje funkcionalosti vezane za kompresiju i dekompresiju poruka u CBOR formatu.

4.4. Programsko okruženje za uređaj virtualne stvarnosti

U radu je korišten *Meta Quest 3* VR-uređaj, koji se sastoji od dva kontrolera i VR-naočala. Posebnost ovog uređaja je što ima kamere na prednjem dijelu naočala, što omogućuje AR (***A**ugmented **R**eality*), odnosno moguće je kroz naočale vidjeti virtualni svijet preklopljen sa stvarnim okruženjem korisnika. Uređaj ima velike mogućnosti praćenja pokreta glave, ruku i prstiju korisnika i daje pristup svim tim podacima putem OpenXR sučelja. *OpenXR* je "open-source" sučelje koje je trenutni standard u VR industriji i pruža podršku za sve tipove VR uređaja. Ovo sučelje rješava programere brige o pojedinostima različitih uređaja na tržištu time što sve funkcionalnosti izlaže programerima u standardiziranom formatu.

4.5. SCOUT mobilni robot

SCOUT MINI mobilni robot korišten je za testiranje teleoperacije i navigacije koristeći razvijeni sustava. Robot je poseban po tome što ima omnidirekcionalni pogon, što ga čini izuzetno manevrabilnim. Sa *Scout*-a se u VR može slati oblak točaka koji je dobiven iz dubinske kamere ili LiDAR senzora. Kako LiDAR ne daje informacije o boji okoline,

koristit će se podaci s dubinske kamere kako bi imali maksimalnu gustoću vizualnih informacija. Kinematika robota opisana je u poglavljima 3.7.2. i 3.7.1. *Scout*-om se može upravljati preko radio upravljača ili slanjem naredbi na */cmd_vel ROS* temu.

Slika 4.2. Mobilni robot *SCOUT MINI*

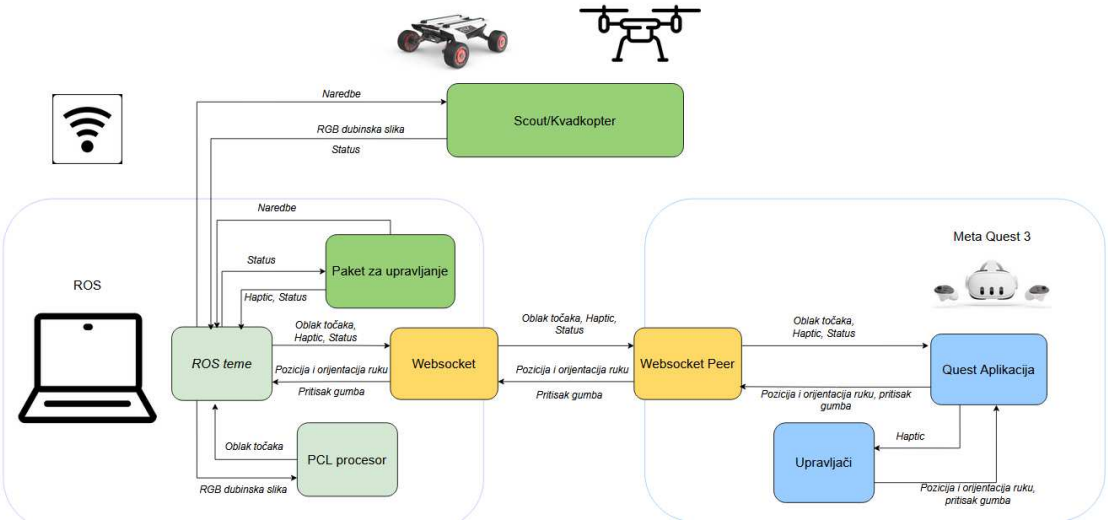
4.6. Kvadkopter

Kvadkopter se u ovom sustavu koristi isključivo u simulacijskom okruženju, kako bi se ubrzao proces razvoja i otklonio rizik oštećenja opreme. Kontrolni sustavi i načini manevriranja kvadkopterom opisani su u poglavlju 3.8.

Slika 4.3. Kvadkopter

4.7. Arhitektura sustava

Sustav za upravljanje robotom/bespilotnom letjelicom uz pomoć VR-naočala, sastoji se od *Quest 3* uređaja, robota ili letjelice i računala koje je povezano na robota ili letjelicu. Pretpostavka je da svi uređaji dijele istu *Wi-fi* mrežu. *Quest* uređaj povezan je s *ROS*-



Slika 4.4. Shema prijenosa ključnih informacija u sustavu

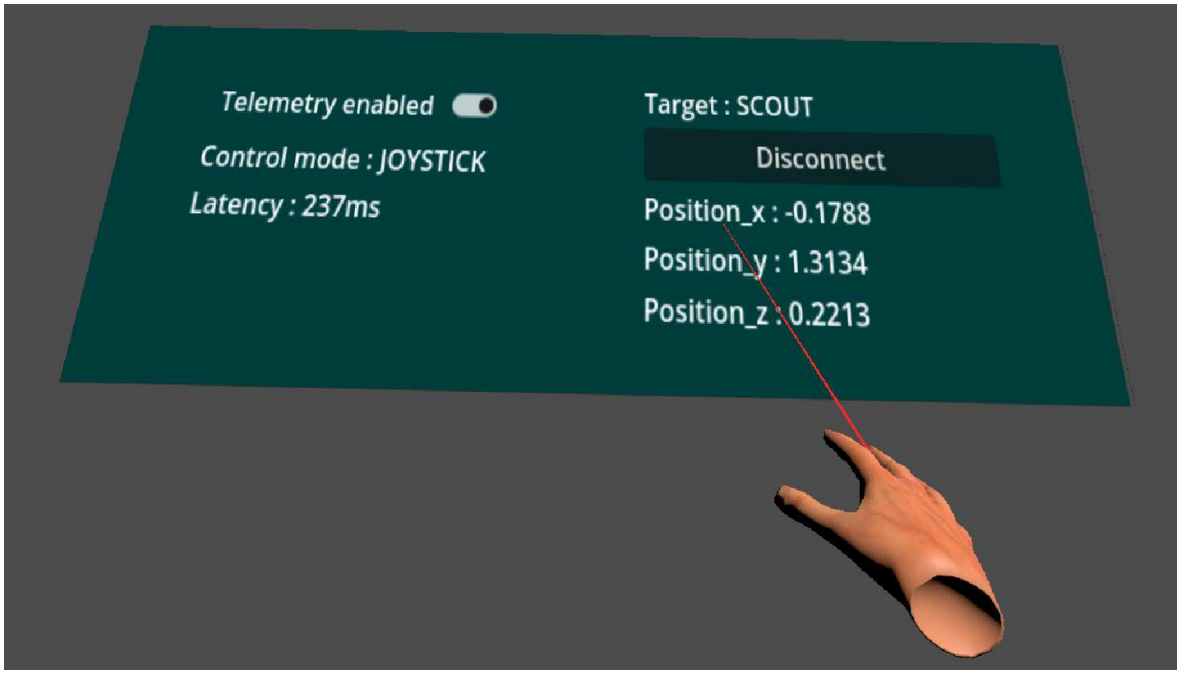
om na računalu preko *websocket*-a i može slati i primati poruke s *ROS*-a. U virtualnom svijetu prikazuje se oblak točaka sa pripadajućeg senzora na robotu/letjelici, to može biti LiDAR ili dubinska kamera. Računalo na kojemu se pokreće *ROS*, služi kao posrednik između *Questa* i robota, zaduženo je za primanje podataka o *Quest* kontrolerima, kontrolnu logiku, kalibraciju upravljačkih sistema, slanje podataka sa senzora i slanje povratnih informacija na *Quest* uređaj. U nastavku se nalazi popis tipova podataka koji se razmjenjuju u komunikaciji.

1. *Status* poruka - sadrži trenutni način upravljanja, status kalibracije, trenutnu poziciju robota (x, y, z)
2. *Latency* poruka - sadrži vrijeme odziva
3. *Pose* poruka - sadrži trenutnu pozu kontrolera
4. *Joystick* poruka - sadrži trenutno stanje *joysticka*
5. *Haptic* poruka - sadrži ciljana svojstva vibracije koja se odvija na kontroleru

6. *Point cloud* poruka - sadrži oblak točaka, zajedno sa svojstvima kao što su boja, veličina *voxel*-a

4.8. Korisničko sučelje

Korisničko sučelje sastoji se od 2D ekrana koji se prikazuje u virtualnoj stvarnosti. Na ekranu je prikazan trenutni status robota kojim upravljamo, zadani način upravljanja te trenutna pozicija robota. Prikazivanje oblaka točaka sa senzora robota odvija se direktno ispred korisnika, uzimajući za referencu orijentaciju VR naočala u trenutku postavljanja na glavu korisnika. Korisnik u određenim situacijama dobiva povratnu informaciju putem vibracije kontrolera (*haptic feedback*).



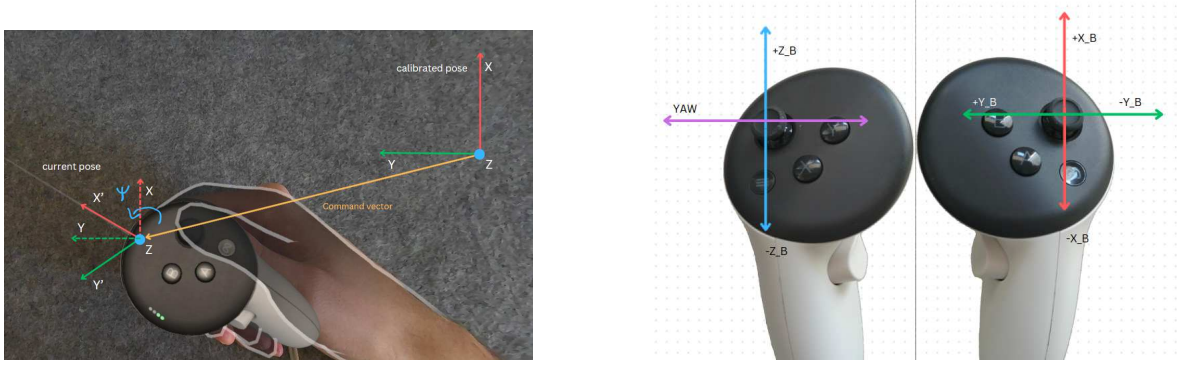
Slika 4.5. Kontrolna ploča u virtualnoj stvarnosti

4.9. Načini upravljanja

- **upravljanje pokretom ruke** - Ovaj način upravljanja oslanja se na praćenje pokreta kontrolera koji se nalazi u ruci korisnika. Prije početka upravljanja provodi se kalibracija sustava: korisnik prvo postavi ruku u proizvoljan početni položaj, zatim pokrene kalibraciju i drži ruku mirnom kako bi se spremila početna pozicija ruke. Jednom kada je kalibracija gotova, svaki pokret ruke koji prelazi upravljački

prag pretvara se u naredbu koja se šalje letjelici ili robotu. Također, na kontroler se šalje *haptic feedback*, što znači da kontroler vibrira u situacijama kada se aktivno šalju naredbe robotu.

- **upravljanje *joystick*-om** - Drugi način upravljanja oslanja se na već ugrađene gumbe i upravljačke palice na kontrolerima te nudi jednostavniju kontrolu. *Joystick* na lijevom kontroleru služi za pomicanje robota po z osi i za zakretanje robota oko z osi. Desni kontroler koristi se za pomicanje robota po x i y osi.



(a) Upravljanje pokretom ruke

(b) Upravljanje *joystick*-om

Slika 4.6. Načini upravljanja

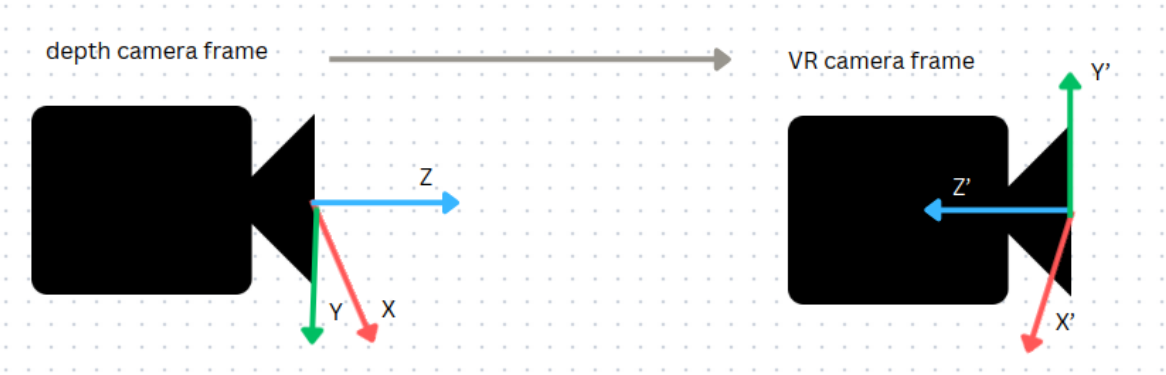
4.10. Komunikacijski sustavl

Komunikacija ROS-a i Quest-a ostvarena je koristeći *Websocket* sa strane ROS-a i gotovu funkcionalnost u Godot-u za spajanje na *Websocket*. *Websocket* koristi TCP kao transportni protokol, što daje sigurnost da je svaki poslani paket došao na svoje odredište i da je pritom očuvan dobar redoslijed paketa. Nedostatak ovog pristupa je što TCP protokol nije zamišljen za slanje velike količine podataka, primjerice očitanja s LIDAR-a, tu lako dolazi do zagušenja, a time i do velikog kašnjenja poruka. Za ovu svrhu, puno prikladniji je UDP protokol koji se koristi u slučajevima kada je cilj dobiti što manje kašnjenje (engl. *latency*). Nedostatak UDP-a je to što nemamo potvrdu da je paket uspješno zaprimljen, niti je garantiran pravilan redoslijed paketa. Poruke koje se šalju preko mreže pristižu u JSON formatu, što je pogodno za obradu s obje strane konekcije. Prilikom inicijalnih testiranja, slanje oblaka točaka predstavljalo je veliki problem jer bi se mreža ubrzo zagušila. Kako bi se povećala propusnost mreže, uspostavljena je kompresija tog tipa poruka. Umjesto JSON formata, za slanje velikih podataka koristi se *CBOR* (skra-

ćeno od engl. *Concise Binary Object Representation*) format, koji značajno poboljšava performanse sustava.

4.11. Prikaz oblaka točaka u virtualnoj stvarnosti

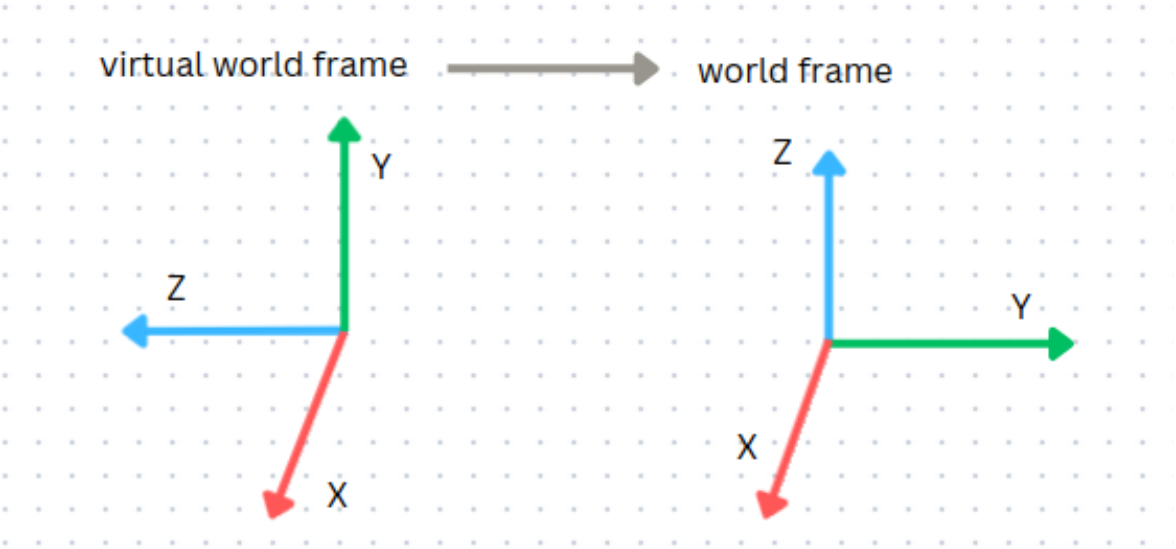
Za prikaz oblaka točaka u virtualnoj sceni potrebno je raspakirati oblak točaka koji je poslan s kamere, pritom je nužno napraviti transformaciju nad točkama kako bi se pravilno prikazale u virtualnoj sceni. Ova transformacija odgovara rotaciji sustava oko x osi za 180 stupnjeva, prikazanoj u jednadžbi 3.2.



Slika 4.7. Prijelaz u koordinatni sustav virtualne scene

4.12. Preslikavanje pokreta

Remapiranje ljudskog pokreta, odnosno pretvorba pokreta ruku u naredbe koje se šalju robotu ključni je dio sustava. Kako bi pokreti čovjeka odgovarali naredbama provedenim na robotu, bitno je uzeti u obzir razliku između referentnih koordinatnih sustava čovjeka (*VR* uređaja) i robota. Primjerice, osi koordinatnog sustava virtualne scene ne odgovaraju osima globalnog referentnog sustava svijeta robota. Nadalje, koordinatni sustav tijela robota se razlikuje od koordinatnog sustava svijeta robota. Relaciju između prvog spomenutog para sustava relativno je jednostavno dobiti rotacijom oko x osi za 90 stupnjeva (3.2.).



Slika 4.8. Prijelaz u koordinatni sustav svijeta robota

Relacija između svijeta robota i tijela robota, dostupna je u obliku kvaterniona koji robot objavljuje uz pomake po x, y, z osi koordinatnog sustava svijeta. Koristeći izraz 3.6.1. mapiraju se naredbe zadane u sustavu svijeta na sustav robota, pritom se trenutna orijentacija (q) kontinuirano ažurira. Ovaj postupak nužan je za uspješno upravljanje robotom.

4.12.1. Generiranje naredbi, ažuriranje pozicija

Za generiranje naredbi koristi se trenutna poza ruke ili kontrolera, koja se zatim uspoređuje s kalibriranom pozom. Prilikom ažuriranja poze ruke, ako se radi o prethodno kalibriranoj ruci na ulaz funkcije za generiranje naredbi dovode se (dx, dy, dz), translacijski pomaci u odnosu na kalibriranu referentnu poziciju.

$$dx = x_{\text{msg}} - x_{\text{calib}} \tag{4.1}$$

$$dy = y_{\text{msg}} - y_{\text{calib}} \tag{4.2}$$

$$dz = z_{\text{msg}} - z_{\text{calib}} \tag{4.3}$$

Udaljenost između trenutne i kalibrirane pozicije izračunava se koristeći Euklidsku normu:

$$d = \sqrt{dx^2 + dy^2 + dz^2} \tag{4.4}$$

Gdje su:

$x_{\text{msg}}, y_{\text{msg}}, z_{\text{msg}}$ koordinate dolazne poruke, $x_{\text{calib}}, y_{\text{calib}}, z_{\text{calib}}$ kalibrirane koordinate ruke. Za izračunavanje kuta zakretanja (engl. yaw) koristi se transformacija kvaterniona u Eulerove kutove:

Zadani kut zakretanja dobivamo kao: $\psi = \psi_{\text{current}} - \psi_{\text{calib}}$, gdje su ψ_{current} — trenutan kut zakretanja ruke/kontrolera. ψ_{calib} - prosječan kut zakretanja dobiven tijekom kalibracije Dobiveni vektor pomaka (dx, dy, dz) množi se transformacijskom matricom ${}^{b}\mathbf{R}_{w}$

$$cmd_vec = {}^{b}\mathbf{R}_{w} \cdot \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} \tag{4.5}$$

Nove koordinate položaja izračunavaju se uz pomoć koraka $step_size$ i prethodne referentne pozicije:

$$x_{\text{new}} = x_{\text{prev}} + \Delta_s \cdot cmd_vec_1 \tag{4.6}$$

$$y_{\text{new}} = y_{\text{prev}} + \Delta_s \cdot cmd_vec_2 \tag{4.7}$$

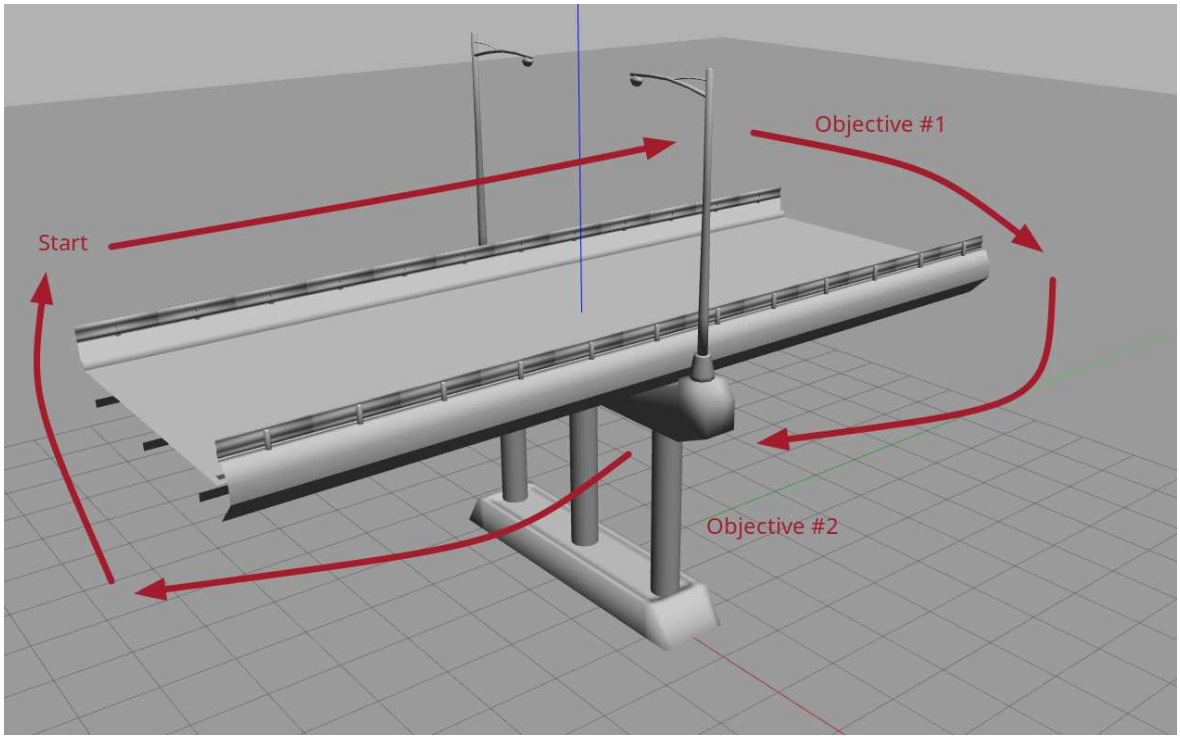
$$z_{\text{new}} = z_{\text{prev}} + \Delta_s \cdot cmd_vec_3 \tag{4.8}$$

Konačno, robotu se šalje naredba koja sadrži nove vrijednosti $(x_{\text{new}}, y_{\text{new}}, z_{\text{new}}, \psi)$.

5. Eksperimentalni rezultati

5.1. Upravljanje letjelicom u simulaciji

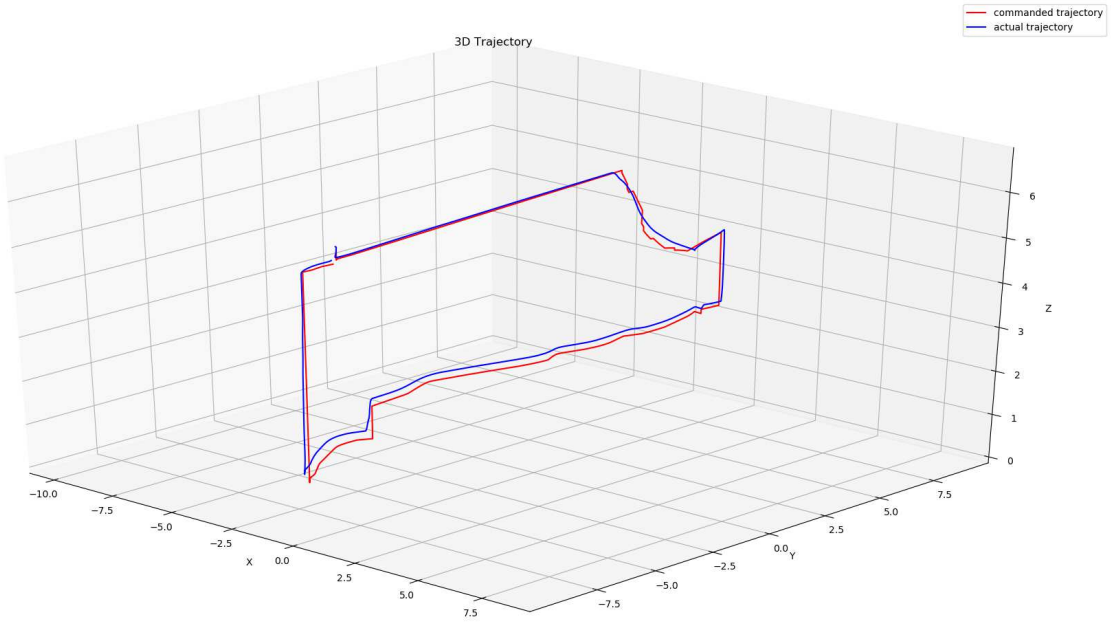
Za ovaj eksperiment odabran je jednostavan svijet u *gazebu* koji se sastoji od segmenta mosta u inače praznom svijetu. Kako bi postojalo nekakvo mjerilo za usporedbu različitih načina upravljanja, odabrana je putanja koju letjelica mora pratiti, pritom zadovoljavajući nekoliko ciljeva. Zadatak je preletiti letjelicom iznad mosta po njegovoj čitavoj dužini, zatim spustiti letjelicu ispod mosta te preletjeti ispod mosta u suprotnom smjeru, pritom prolazeći između nosivih stupova. Konačno, letjelicu treba vratiti u početnu poziciju i orijentaciju. Eksperiment je potrebno napraviti prvo koristeći način upravljanja pozom ruke, zatim koristeći upravljanje *joystick*-om.



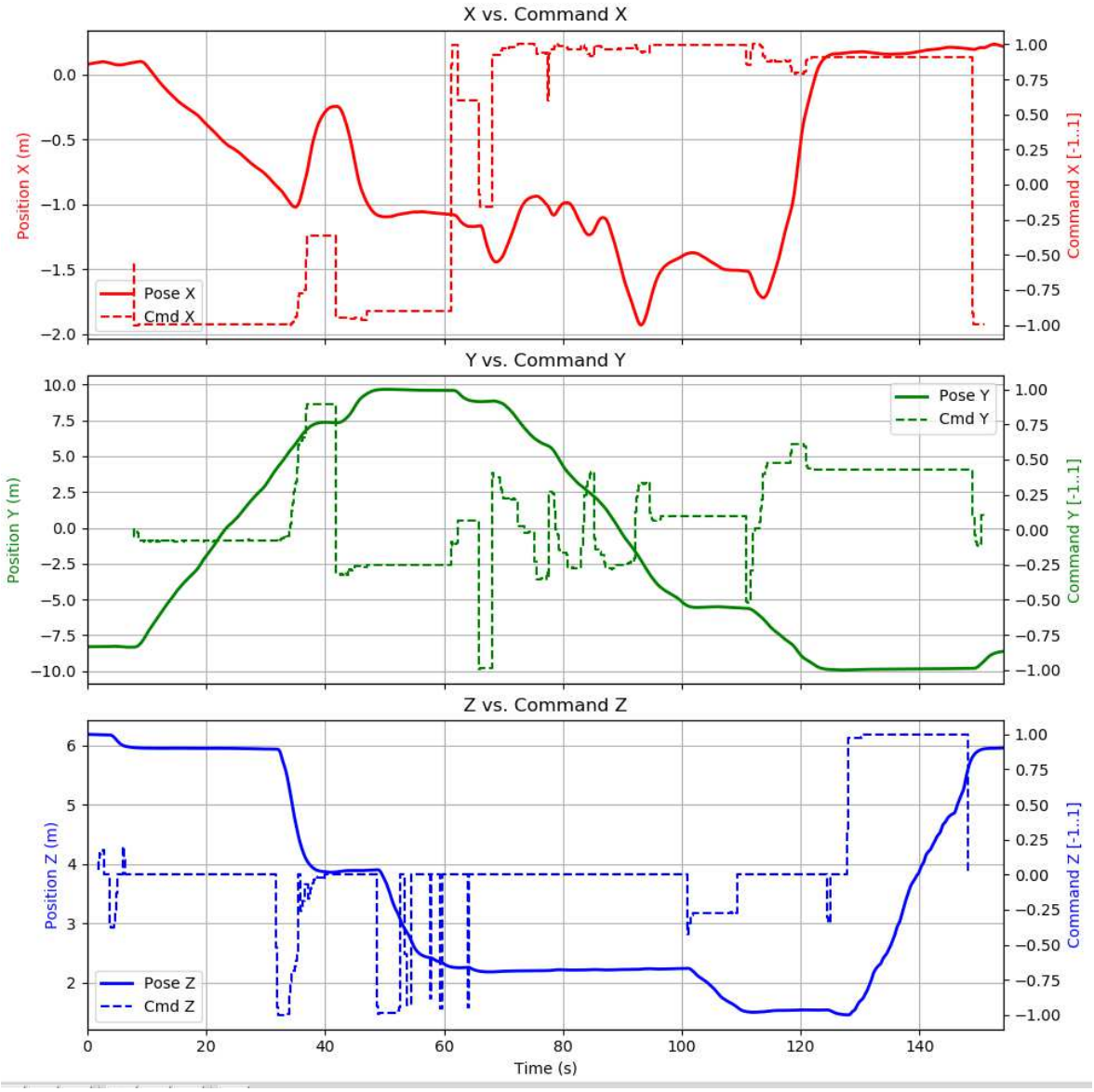
Slika 5.1. Planirana putanja letjelice

5.2. Upravljanje letjelicom kontrolerom u simulaciji

Test upravljanja *joystickom* je uspješno obavljen. Na grafu je prikazana putanja letjelice dobivena iz zadanih naredbi (u crvenom) i stvarna putanja letjelice (u plavom). Vidljivo je da nema velikih odstupanja, što ukazuje na to da se letjelicom može sigurno upravljati.



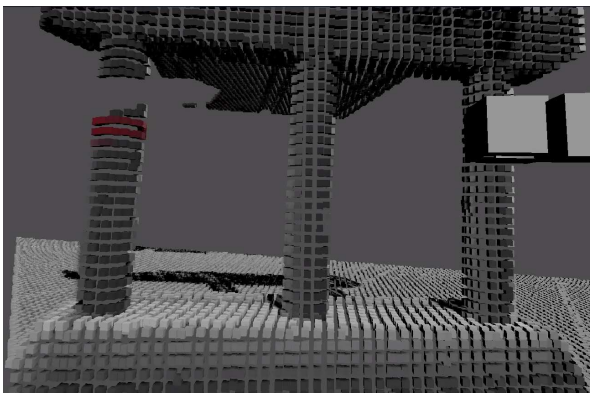
Slika 5.2. Putanja letjelice u simulaciji



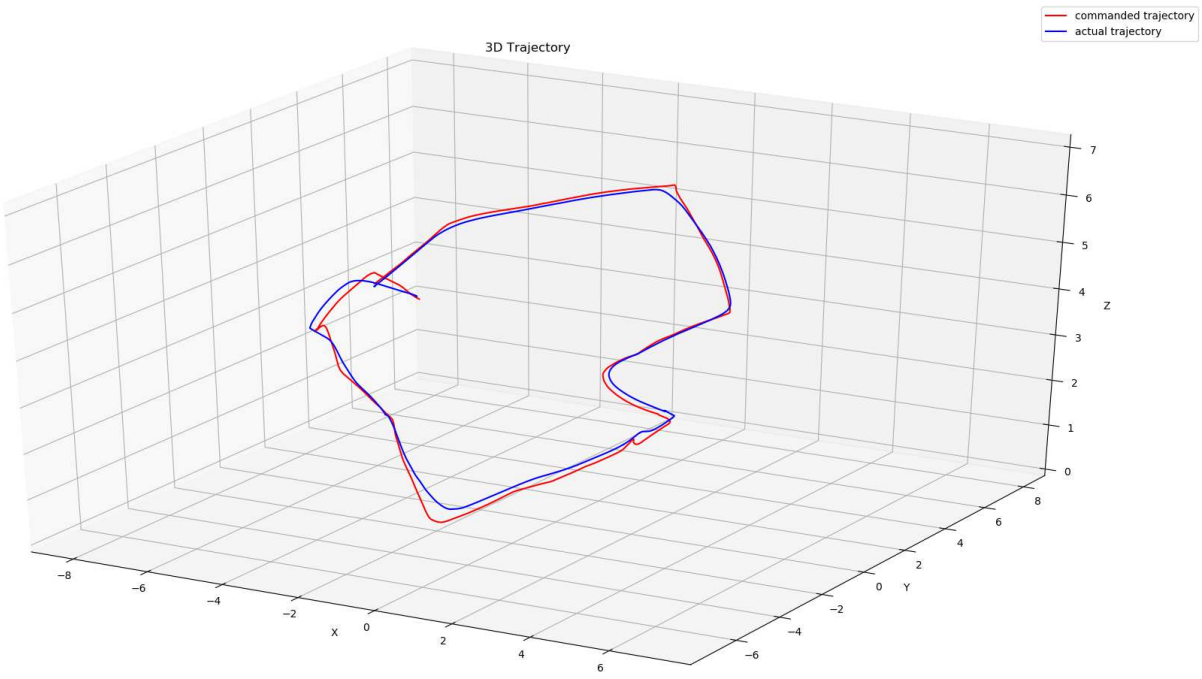
Slika 5.3. Prikaz komponenti naredbi i pozicija po x, y, z kroz vrijeme.

5.3. Upravljanje letjelicom pokretom ruku u simulaciji

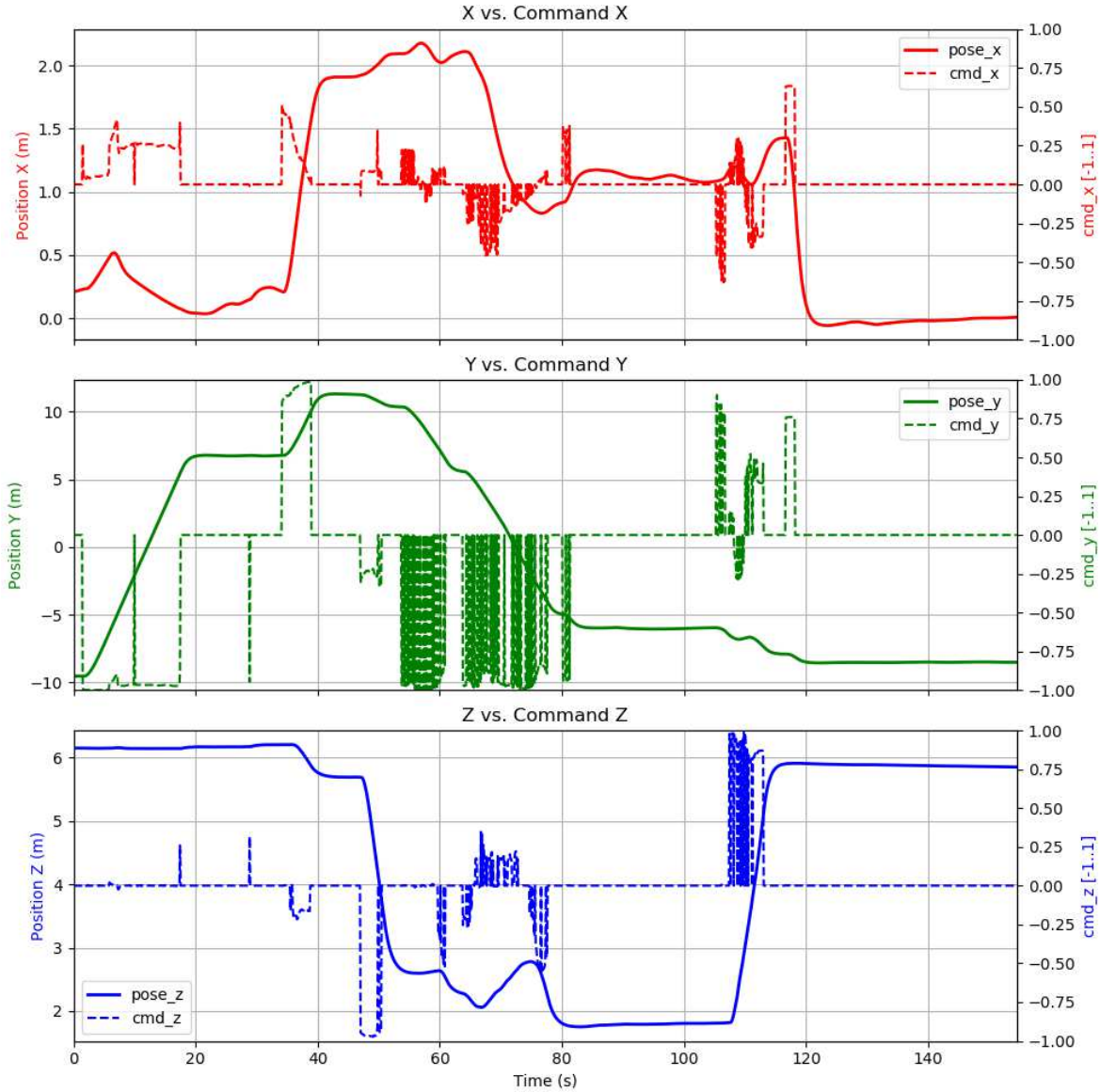
Test letjelice odrađen je uspješno, letjelica je zadovoljila sve uvjete određene u 5.1., uključujui i prolazak letjelice između stupova mosta. Putanja letjelice u ovom slučaju izgleda glatko, zbog toga što se naredbe šalju letjelici kontinuirano kako pristižu podaci o poziciji ruke. Prilikom testiranja korisnici su imali pozitivna mišljenja o kontroli pozicijom ruke, neki čak preferiraju ovaj način upravljanja naspram upravljanja *joystickom*.



Slika 5.4. Pogled s dubinske kamere letjelice u virtualnoj stvarnosti



Slika 5.5. Putanja letjelice u simulaciji



Slika 5.6. Prikaz komponenti naredbi i pozicija po x, y, z kroz vrijeme. Napomena: podaci su uzeti iz različite snimke, ne odgovaraju gornjoj putanji

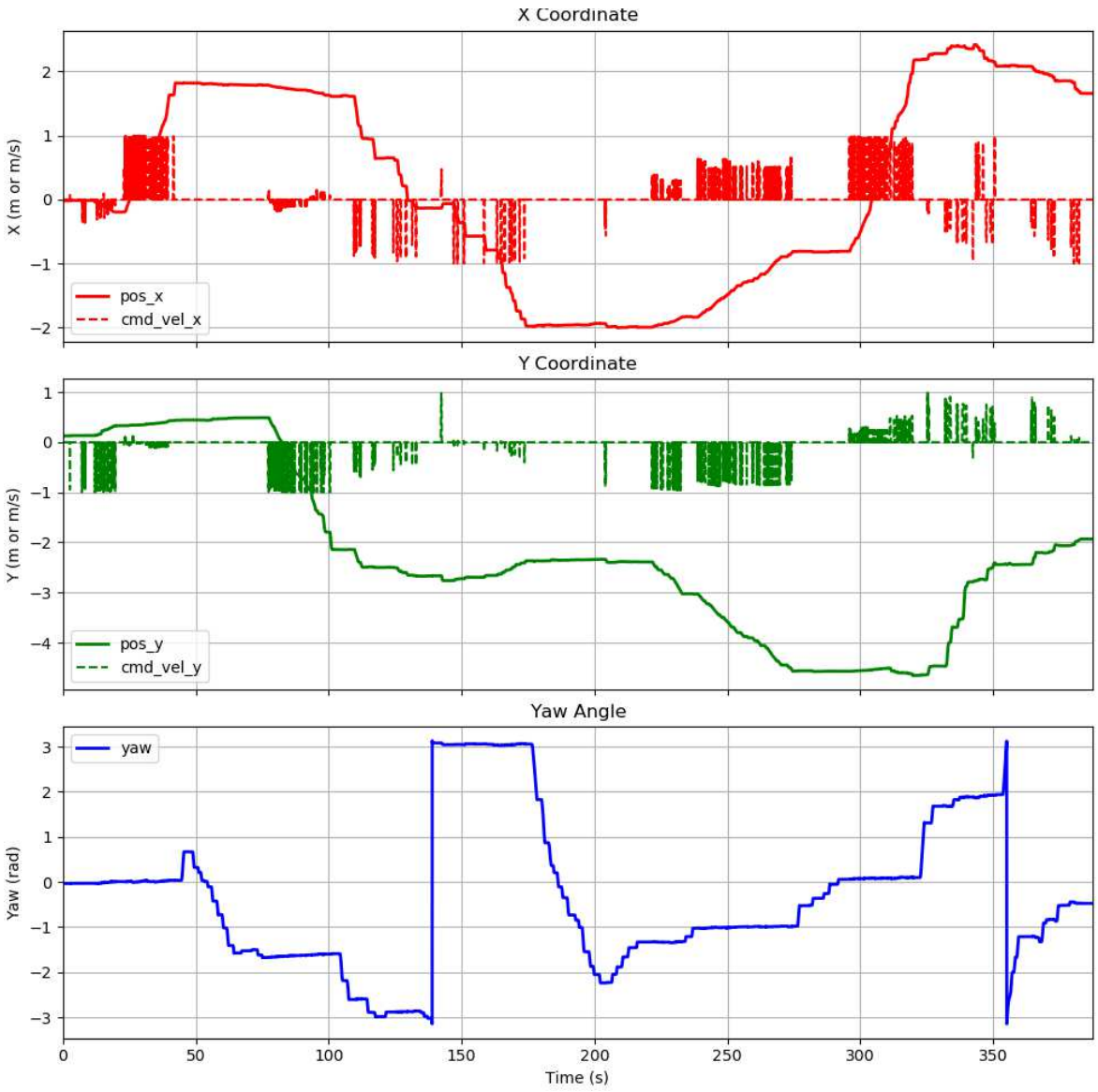
5.4. Upravljanje Scout-om u stvarnom okruženju

Cilj ovog eksperimenta je pokazati sposobnost uspješnog navigiranja kroz zatvoreni prostor, koristeći samo informacije sa Scout-ovih senzora prikazanih u virtualnoj stvarnosti. Za uspješno rješavanje zadatka potrebno je :

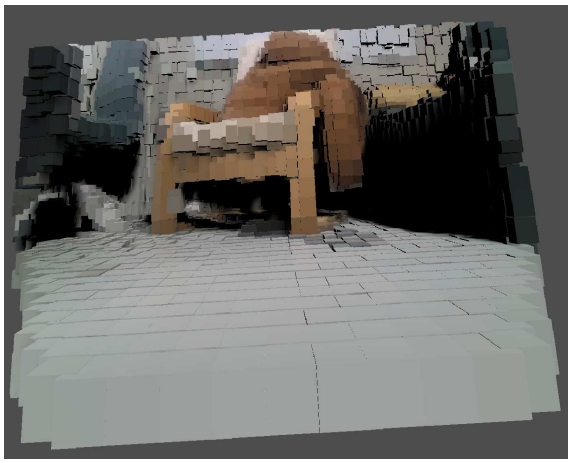
- Navigirati *Scout*-a od početka do kraja sobe, pritom izbjegavajući bilokakve prepreke

Može se reći da je i ovaj eksperiment zadovoljio zahtjeve, operater je vozio *Scout*-a kroz čitavu prostoriju i istovremeno izbjegavao prepreke (pokretne i nepokretne). U nas-

tavku su priloženi grafovi koji prikazuju podatke o poziciji *Scout*-a tijekom eksperimenta i naredbe koje je *Scout* primio.



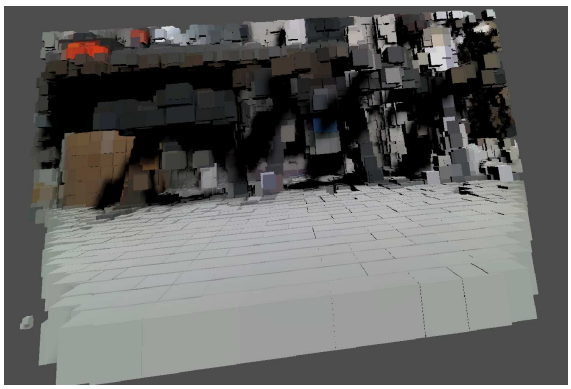
Slika 5.7. Prikaz podataka o poziciji i orijentaciji *Scout* robota, preklopljen sa prikazom odaslanih naredbi



(a) Oblak točaka sa Scout-a u virtualnoj stvarnosti

(b) Stvarna slika iz slične perspektive

Slika 5.8. Usporedba VR-a i stvarnosti



(a) Oblak točaka sa scout-a u virtualnoj stvarnosti

(b) Stvarna slika iz slične perspektive

Slika 5.9. Usporedba VR-a i stvarnosti

6. Zaključak

U sklopu rada razvijeno je sučelje čovjeka i robota u virtualnoj stvarnosti koje omogućuje teleoperaciju različitih vrsta robota u stvarnom svijetu preslikavanjem ljudskog pokreta. Opisane su funkcionalnosti sustava, njegova arhitektura i teorijska podloga.

Mogućnosti:

- Sustav omogućuje upravljanje letjelicom i mobilnim robotom u stvarnom vremenu.
- Operater u virtualnoj stvarnosti ima dovoljno podataka kako bi uspješno izbjegavao prepreke i upravljao letjelicom/mobilnim robotom.
- Operater može upravljati na dva načina, te može mijenjati način upravljanja tijekom leta/vožnje.
- Sustav održava stabilnu komunikaciju između operatera i robota i osigurava protok kritičnih informacija.

Nedostatci:

- Slanje oblaka točaka uvelike je ograničeno s propusnošću mreže, stoga se nerijetko kad mora smanjiti rezolucija oblaka točaka kako bi se održala prihvatljiva razina responzivnosti sustava.
- Prosječno vrijeme odziva je oko 200-300ms (milisekundi), što bi se trebalo smanjiti na manje od 100ms kako bi sustav bio responzivniji.

Razmatrajući eksperimente, utvrđeno je nekoliko područja u kojima je moguće poboljšati sustav. Jedna od mogućnosti je dodavanje funkcionalnosti za slanje vibracija na kontroler ovisno o blizini prepreka ili očitanja sa senzora robota (u slučaju kada robot do-

takne neku prepreku) [20]. Stvar koja najviše može poboljšati korisničko iskustvo operatera je povećanje rezolucije oblaka točaka koji se prikazuje i smanjenje vremena odziva. Za velik napredak u tom području potrebo je akumulirati puno malih unaprijeđenja poput bolje kompresije, boljeg iskorištavanja mreže i prioritiziranja poruka. Daleko najveća poluga u tom smislu je sama brzina mreže, stoga bi se u budućnosti trebalo razmotriti korištenje stvari poput 5G mreža.

Literatura

[1] D. A. Sara Riso, "eurofound.europa.eu", 2024. [Mrežno]. Adresa: https://www.eurofound.europa.eu/hr/publications/2024/interakcija-covjeka-i-robota-sto-se-mijenja-na-radnom-mjestu?

[2] Y. Lei, Z. Su, i C. Cheng, "Virtual reality in human-robot interaction: Challenges and benefits", *Electronic Research Archive*, sv. 31, br. 5, str. 2374–2408, 2023.

[3] X. Cheng, J. Li, S. Yang, G. Yang, i X. Wang, "Open-television: Teleoperation with immersive active visual feedback", *arXiv preprint arXiv:2407.01512*, 2024.

[4] E. Rosen i D. K. Jha, "A virtual reality teleoperation interface for industrial robot manipulators", *arXiv preprint arXiv:2305.10960*, 2023.

[5] H. Kasaei i M. Kasaei, "Vital: Visual teleoperation to enhance robot learning through human-in-the-loop corrections", *arXiv preprint arXiv:2407.21244*, 2024.

[6] F. De Pace, G. Gorjup, H. Bai, A. Sanna, M. Liarokapis, i M. Billinghurst, "Leveraging enhanced virtual reality methods and environments for efficient, intuitive, and immersive teleoperation of robots", u *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021., str. 12 967–12 973.

[7] L. Meng, J. Liu, W. Chai, J. Wang, i M. Q.-H. Meng, "Virtual reality based robot teleoperation via human-scene interaction", *Procedia Computer Science*, sv. 226, str. 141–148, 2023.

[8] S. Lynch, "Ai index: State of ai in 13 charts", 2024. [Mrežno]. Adresa: https://hai.stanford.edu/news/ai-index-state-ai-13-charts

[9] Z. Fu, Q. Zhao, Q. Wu, G. Wetzstein, i C. Finn, "Humanplus: Humanoid shadowing and imitation from humans", *arXiv preprint arXiv:2406.10454*, 2024.

[10] F. Zorić i M. Orsag, "H2ami: Intuitive human to aerial manipulator interface", u *2023 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2023., str. 1226–1232. https://doi.org/10.1109/ICUAS57906.2023.10156265

[11] M. Wonsick i T. Padir, "A systematic review of virtual reality interfaces for controlling and interacting with robots", *Applied Sciences*, sv. 10, br. 24, str. 9051, 2020.

[12] G. LeMasurier, J. Allspaw, i H. A. Yanco, "Semi-autonomous planning and visualization in virtual reality", *arXiv preprint arXiv:2104.11827*, 2021.

[13] F. Kennel-Maushart, R. Poranne, i S. Coros, "Manipulability optimization for multi-arm teleoperation", u *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021., str. 3956–3962.

[14] A. Naceri, D. Mazzanti, J. Bimbo, Y. T. Tefera, D. Prattichizzo, D. G. Caldwell, L. S. Mattos, i N. Deshpande, "The vicarios virtual reality interface for remote robotic teleoperation: Teleporting for intuitive tele-manipulation", *Journal of Intelligent & Robotic Systems*, sv. 101, str. 1–16, 2021.

[15] X. Wang, L. Shen, i L.-H. Lee, "A systematic review of xr-based remote human-robot interaction systems", *arXiv preprint arXiv:2403.11384*, 2024.

[16] L. Morando i G. Loianno, "Spatial assisted human-drone collaborative navigation and interaction through immersive mixed reality", 2024. [Mrežno]. Adresa: https://arxiv.org/abs/2402.04070

[17] [Mrežno]. Adresa: https://motion.cs.illinois.edu/RoboticSystems/CoordinateTransformations.html

[18] [Mrežno]. Adresa: https://py.collimator.ai/examples/01_quadcopter_modelling/

[19] S. A. Raza i W. Gueaieb, "Intelligent flight control of an autonomous quadrotor", u *Motion Control*, F. Casolo, Ur. Rijeka: IntechOpen, 2010., pog. 12. https://doi.org/10.5772/6968

[20] G. A. Yashin, D. Trinitatova, R. T. Agishev, R. Ibrahimov, i D. Tsetserukou, “Aerovr: Virtual reality-based teleoperation with tactile feedback for aerial manipulation”, u *2019 19th International Conference on Advanced Robotics (ICAR)*. IEEE, 2019., str. 767–772.

Sažetak

Razvoj sučelja čovjeka i robota predstavlja velik izazov, ujedno i priliku za olakšavanje suradnje čovjeka i robota. Tradicionalna sučelja čovjeka i robota prikazuju podatke na 2D ekranima, najčešće u obliku različitih grafova i video prijenosa s kamera. Neiskusnim korisnicima ovakva sučelja djeluju izazovno i nepraktično. Ovaj rad istražuje primjenu virtualne stvarnosti za razvijanje intuitivnog sučelja za teleoperaciju robota. Uspoređuju se dvije metode: jedna se oslanja na praćenje pokreta čovjeka, a druga koristi *joystick* kontrolere dostupne uz VR naočale. Iako je sustav do sada testiran na samo dva tipa robota, njegova prilagodljivost omogućuje upravljanje širokim spektrom mobilnih robota i zračnih vozila, a rezultati pokazuju da se uspješno može koristiti u stvarnim uvjetima.

Ključne riječi: virtualna stvarnost; sučelje robota i čovjeka; teleoperacija

Abstract

Developing intuitive human-robot interfaces presents a great opportunity to make human-robot interaction more practical, as conventional human-robot interfaces typically involve looking at telemetry info on a 2D screen which makes it difficult for untrained people to make sense of the data. This paper investigates developing more intuitive interfaces for robot teleoperation using virtual reality (VR) headset with hand held controllers. We compare two ways of operating robots in VR, one using human motion tracking, an the other using familiar joystick controllers that come with our Meta Quest 3 headset. Although our system is currently focused and tested on just two types of robots, it can be adapted to support teleoperation of a wide range of mobile robots and aerial vehicles. Our experimental results suggest that the system can be succesfully used for operating mobile robots in the real-world scenarios.

Keywords: robot teleoperation; human robot interaction (HRI); virtual reality (VR); robotics; mobile robots