

UNIVERZA V LJUBLJANI
FAKULTETA ZA ELEKTROTEHNIKO

Aljaž Baumkircher

**OBOGATENO KINESTETIČNO
UČENJE FINIH ROBOTSKIH GIBOV**

DOKTORSKA DISERTACIJA

Mentor: prof. dr. Matjaž Mihelj

Ljubljana, 2022

Zahvala

Za vse nasvete, pomoč in usmerjanje na moji raziskovalni poti kakor tudi med nastajanjem te doktorske disertacije se iskreno zahvaljujem mentorju prof. dr. Matjažu Mihelju ter ostalim kolegom iz Laboratorija za robotiko (ULFE).

Nadalje se zahvaljujem prof. dr. Katji Seme, dr. med. iz Inštituta za mikrobiologijo in imunologijo (ULMF) za omogočen dostop do opreme in strokovno podporo v sklopu študije identifikacije bakterijskih kolonij. Zahvala gre seveda tudi vsem ostalim kolegicam iz Laboratorija za bakteriološko diagnostiko respiratornih infekcij za pripravo (pre)mnogih petrijevkv z vzorci in kakšno spodbudno besedo.

Zahvala gre Javni agenciji za raziskovalno dejavnost Republike Slovenije (ARRS), ki mi je omogočila financiranje doktorskega študija.

Nazadnje bi se rad zahvalil še svojim staršem za vso izkazano podporo in pripravljenost subvencioniranja mojega življenja "ker nimam prave službe" (citirano mami Mojce, Corner Pub, datum neznan).

Seveda pa gre zahvala tudi mojim prijateljem in boljši polovici Nevi, s pomočjo katerih sem prebrodil ta štiri leta brez resne duševne krize.

Hvala!

Kazalo

Seznam uporabljenih kratic	xiii
Povzetek	1
Abstract	3
1 Uvod	5
1.1 Cilji doktorske disertacije	8
2 Robotski sistem	11
2.1 Sodelujoči robot Franka Emika Panda	11
2.2 Vodenje robotov	13
2.2.1 Sistem vodenja	13
2.2.2 Kinestetično vodenje	15
2.2.3 Teleoperacija	16
2.2.4 Kooperativno robotsko orodje	17
2.2.5 Programsko generiranje gibov	19
2.3 Kalibracija kinematičnega modela	20
3 Demonstriranje finih robotskih gibov	25
3.1 Materiali in metode	26
3.1.1 Postavitev študije	26
3.1.2 Naloge	29

3.1.3	Protokol študije	31
3.1.4	Analiza podatkov	32
3.1.5	Statistična analiza	33
3.1.6	Vprašalnik	34
3.2	Rezultati	34
3.2.1	Naloga 1: premik od točke do točke	35
3.2.2	Naloga 2: sledenje poti	38
3.2.3	Vprašalnik	42
3.3	Diskusija	43
4	Analitični zapis finih robotskih gibov	47
4.1	Izbira in simulacija trajektorij za analizo metod	48
4.2	Metoda dinamičnih generatorjev gibov	49
4.3	Metoda mešanih Gaussovih modelov	50
4.4	Nadgradnja metode GMM s prilagajanjem gostote komponent na podlagi frekvenčne analize	51
4.5	Rezultati	54
4.5.1	Dinamični generatorji gibov	54
4.5.2	Mešani Gaussovi modeli	58
4.5.3	Nadgradnja metode GMM s prilagajanjem gostote komponent na podlagi frekvenčne analize	61
4.6	Diskusija	65
5	Študija kinestetičnega učenja finih robotskih gibov z uporabo učnega vmesnika v sklopu laboratorijske aplikacije	69
5.1	Eksperimentalni sistem	71
5.2	Učni vmesnik	72
5.2.1	Obogatena resničnost kot pomoč pri izvajanju demonstracij	74
5.2.2	Učenje na podlagi demonstracij	76

5.3	Identifikacija bakterijskih kolonij v procesu MALDI-TOF	77
5.3.1	Eksperimentalni protokol	78
5.3.2	Rezultati	79
5.3.2.1	Odvzem kolonij	79
5.3.2.2	Nanos kolonij	80
5.3.2.3	Identifikacija kolonij	81
5.4	Validacija učnega vmesnika	82
5.5	Diskusija	84
6	Zaključek	87
	Izvirni prispevki doktorske disertacije	91
	Literatura	94
6.1	Eksperimentalni sistem in protokol študije	103
6.2	Rezultati	104
6.3	Diskusija	109

Slike

2.1	Shematski prikaz splošnega eksperimentalnega sistema, ki prikazuje hierarhijo ter tok podatkov med robotom, krmilnikom, napravami za vodenje robota (npr. haptični vmesnik pri teleoperaciji), specifičnimi senzorji za delovanje aplikacije (npr. RGB kamera) in obogatnim okoljem potrebnim za ustrezen prikaz poteka aplikacije.	12
2.2	PD regulacija lege v zunanjih koordinatah s kompenzacijo gravitacije ter Coriolisovih in centripetalnih sil.	15
2.3	Shematski prikaz operaterjevega nadzorovanja gibanja robota s pristopom kinestetičnega vodenja.	15
2.4	Shematski prikaz operaterjevega nadzorovanja gibanja robota s pristopom teleoperacije.	16
2.5	Shematski prikaz operaterjevega nadzorovanja gibanja robota s pristopom kooperativnega robotskega orodja.	18
2.6	Prikaz gladkosti časovnega poteka pozicije, hitrosti in pospeška pri uporabi metode minimalnega sunka.	19
2.7	Prikaz različnih kalibracijskih mrež glede na študijo. Leva slika prikazuje kalibracijsko mrežo študije demonstracije finih gibov, ki je bila prikazana na LCD zaslonu, medtem ko desna slika prikazuje kalibracijsko mrežo študije kinestetičnega učenja v sklopu laboratorijske aplikacije, ki je bila definirana z izvrtinami na aluminjasti plošči.	22
2.8	Prikaz mehanske strukture robota Panda, pripadajočih koordinatnih sistemov ter izračunanih vrednosti DH parametrov za primer študije finih gibov.	23

3.1	Prikaz postavitve študije (a) in način uporabe naprav pri upravljanju z robotom Panda s pristopom kinestetičnega učenja (b), teleoperacije (c) in kooperativnega robotskega orodja (d).	27
3.2	Shematski prikaz postavitve študije.	28
3.3	Podroben prikaz uporabniškega vmesnika - naloga premika od točke do točke (a) in sledenja poti (b), funkcija vizualnega skaliranja (c) in indikator trenutne vrednosti prostorskega skaliranja (d).	28
3.4	Prikaz izvedenih demonstracij operaterja v posamezne končne točke (1 - 5) v sklopu naloge giba od točke do točke.	29
3.5	Prikaz izvedene demonstracije operaterja v sklopu naloge gibanja vzdolž poti.	30
3.6	Prikaz pogreškov pozicioniranja pri odmiku od začetne točke in med približevanjem končni točki.	35
3.7	Hitrost in gladkost demonstriranih gibov med vmesnim premikom in pri približevanju končni točki.	37
3.8	Pregled pogreškov pozicioniranja, hitrosti in gladkosti gibanja v vzdolž posameznih segmentov poti.	39
3.9	Pregled subjektivnih ocen udeležencev o pristopih in vplivu funkcij vizualnega in prostorskega skaliranja. Rdeči + znaki predstavljajo podatkovne osamelce.	42
4.1	Prikaz trajektorije s finimi gibi (a) in trajektorije brez finih gibov (b).	48
4.2	Prikaz frekvenčne analize izbrane trajektorije z uporabo Welcheve transformacije (a), normiranega spektrograma $\hat{\mathbf{S}}(t, \omega)$ (b), potek srednje vrednosti dominantnih frekvenc $\bar{f}_d(t)$ (c) in porazdelitev komponent mešanega Gaussovega modela vzdolž trajektorije (d).	53
4.3	Prikaz zapisov trajektorije brez finih gibov pri uporabi metode DMP z N baznimi funkcijami.	55
4.4	Prikaz generiranih pogreškov v primeru trajektorije brez finih gibov vzdolž posameznih segmentov (A_1, A_2, A_3) pri uporabi metode DMP z N baznimi funkcijami.	56

4.5	Prikaz zapisov trajektorije s finimi gibi pri uporabi metode DMP z N baznimi funkcijami.	57
4.6	Prikaz generiranih pogreškov v primeru trajektorije s finimi gibi vzdolž posameznih segmentov ($A_1 - A_5$) pri uporabi metode DMP z N baznimi funkcijami.	57
4.7	Prikaz poteka vrednosti BIC in AIC glede na število komponent pri trajektoriji brez finih gibov (a) in trajektoriji s finimi gibi (b).	58
4.8	Prikaz zapisov trajektorije brez finih gibov pri uporabi metode GMM z N komponentami.	59
4.9	Prikaz generiranih pogreškov v primeru trajektorije brez finih gibov vzdolž posameznih segmentov (A_1, A_2, A_3) pri uporabi metode GMM z N komponentami.	59
4.10	Prikaz zapisov trajektorije s finimi gibi pri uporabi metode GMM z N komponentami.	60
4.11	Prikaz generiranih pogreškov v primeru trajektorije s finimi gibi vzdolž posameznih segmentov ($A_1 - A_5$) pri uporabi metode GMM z N komponentami.	61
4.12	Prikaz zapisov trajektorije brez finih gibov pri uporabi metode GMM z $N = 20$ komponentami ob različnih pristopih razporeditve komponent vzdolž podatkov.	63
4.13	Prikaz generiranih pogreškov v primeru trajektorije brez finih gibov vzdolž posameznih segmentov (A_1, A_2, A_3) pri uporabi metode GMM z enakomerno (E), linearno naraščajočo (L) ali frekvenčno (F) razporeditvijo komponent.	63
4.14	Prikaz zapisov trajektorije brez finih gibov pri uporabi metode GMM z N komponentami.	64
4.15	Prikaz generiranih pogreškov v primeru trajektorije s finimi gibi vzdolž posameznih segmentov ($A_1 - A_5$) pri uporabi metode GMM z enakomerno (E), linearno naraščajočo (L) ali frekvenčno (F) razporeditvijo komponent.	64

4.16	Prikaz porazdelitve komponent vzdolž podatkov glede na izbran pristop razporeditve komponent. Prikaz je ločen na porazdelitev komponent v primeru trajektorije brez finih gibov (a) in porazdelitev komponent v primeru trajektorije s finimi gibi (b).	67
5.1	Prikaz poteka identifikacije organizmov z uporabo MALDI-TOF postopka v sklopu klinične mikrobiologije.	70
5.2	Prikaz eksperimentalnega sistema (a) in orodja na vrhu manipulatorja (b).	72
5.3	Prikaz poteka eksperimenta - slika petrijevke zajete z RGB kamero (a), 3D model izbrane bakterijske kolonije generiran z uporabo linijskega laserskega merilnika (b), prikaz procesa odvzema kolonije (c) in prikaz procesa nanosa kolonije (d).	73
5.4	Prikaz operaterja pri demonstriranju giba z uporabo obogatene resničnosti (a), bakterijskih kolonij na podlagi katerih je bil generiran 3D model kolonije (b) ter pogled operaterja na obogateno delovno okolje ob oddaljenosti od tarčne kolonije (c) in pogled na obogateno delovno okolje v bližini kolonije (d).	75
5.5	Prikaz zapisa referenčnih trajektorij procesa odvzema (a) in nanosa (b) z uporabo metode DMP.	77
5.6	Ravninski prikaz želenega poteka trajektorije vrha robota (orodja) pri odvzemu kolonije bakterij.	78
5.7	<i>Ravninski toplotni prikaz</i> 3D modelov kolonij pred in po odvzemu. Kolonije so razvrščene od (1) do (6) glede na začetni volumen.	80
5.8	Prikaz kontaktne sile med vrhom robota in tarčno ploščo in posledična kompenzacija pozicije vzdolž osi z (a) in fotografija tarčne plošče pred postopkom identifikacije (b).	81
5.9	Prikaz plošče z referenčnimi točkami.	82
5.10	Prikaz vpliva obogatene resničnosti na natančnost pozicioniranja (a), hitrost (b) in gladkost (c) gibanja kinestetičnega učenja. Prikazani so tudi rezultati statistične primerjave, pri čemer * predstavlja statistično signifikantno razliko.	83

6.1	Prikaz demonstriranega dinamičnega giba z uporabo kinestetičnega učenja pri referenčni frekvenci f_1 . Graf (a) prikazuje časovni potek pozicije vrha robota v primerjavi z referenco, medtem ko graf (b) prikazuje pospeške vrha robota in vrednost vsiljenih sil.	105
6.2	Prikaz demonstriranega dinamičnega giba z uporabo kinestetičnega učenja pri referenčni frekvenci f_2 . Graf (a) prikazuje časovni potek pozicije vrha robota v primerjavi z referenco, medtem ko graf (b) prikazuje pospeške vrha robota in vrednost vsiljenih sil.	105
6.3	Prikaz demonstriranega dinamičnega giba z uporabo pristopa CRT pri referenčni frekvenci f_1 . Graf (a) prikazuje časovni potek pozicije vrha robota v primerjavi z referenco, medtem ko graf (b) prikazuje pospeške vrha robota in vrednost vsiljenih sil.	106
6.4	Prikaz demonstriranega dinamičnega giba z uporabo pristopa CRT pri referenčni frekvenci f_2 . Graf (a) prikazuje časovni potek pozicije vrha robota v primerjavi z referenco, medtem ko graf (b) prikazuje pospeške vrha robota in vrednost vsiljenih sil.	107
6.5	Prikaz demonstriranega dinamičnega giba z uporabo pristopa teleoperacije pri referenčni frekvenci f_1 . Graf (a) prikazuje časovni potek pozicije vrha robota v primerjavi z referenco, medtem ko graf (b) prikazuje pospeške vrha robota in vrednost vsiljenih sil.	108
6.6	Prikaz demonstriranega dinamičnega giba z uporabo pristopa teleoperacije pri referenčni frekvenci f_2 . Graf (a) prikazuje časovni potek pozicije vrha robota v primerjavi z referenco, medtem ko graf (b) prikazuje pospeške vrha robota in vrednost vsiljenih sil.	108
6.7	Predloga grafičnega prikaza statistične analize.	111
6.8	Premik od točke do točke - statistični pregled testiranja ničtih hipotez primerjave različnih pristopov demonstriranja ter vpliva prostorskega in vizualnega skaliranja.	112
6.9	Sledenje poti - statistični pregled testiranja ničtih hipotez primerjave različnih pristopov demonstriranja ter vpliva prostorskega in vizualnega skaliranja.	113

Tabele

5.1	Ocenjena sprememba volumna kolonije pri procesu odvzema.	79
5.2	Rezultati identifikacije kolonij.	81
6.1	Prikaz izračunanih značilnk, ki omogočajo oceno kvalitete izvedbe dinamičnega giba z uporabo kinestetičnega učenja.	106
6.2	Prikaz izračunanih značilnk, ki omogočajo oceno kvalitete izvedbe dinamičnega giba z uporabo pristopa CRT.	107
6.3	Prikaz izračunanih značilnk, ki omogočajo oceno kvalitete izvedbe dinamičnega giba z uporabo teleoperacije.	109

Seznam uporabljenih kratic

KRATICA	OPIS
CRT	kooperativno robotsko orodje (angl. cooperative robot tool)
DMP	dinamični generatorji gibov (angl. dynamic movement primitives)
GMM	mešani Gaussovi modeli (angl. Gaussian mixture models)
KMP	jedrni generatorji gibov (angl. kernelized movement primitives)
ProMP	verjetnostni generatorji gibov (angl. probabilistic movement primitives)
GMR	Gaussova mešana regresija (angl. Gaussian mixture regression)
EM	oceni-maksimiziraj (angl. expectation-maximization)
MLE	največje verjetje (angl. maximum likelihood estimation)
DTW	nelinearna časovna poravnava (angl. dynamic time warp)
AIC	Akaikov informacijski kriterij (angl. Akaike information criterion)
BIC	Bayesov informacijski kriterij (angl. Bayesian information criterion)
SAL	gladkost giba (angl. spectral arc length)
DH	Denavit-Hartenberg parametri
AR	obogatena resničnost (angl. augmented reality)
MS	masna spektrometrija (angl. mass spectrometry)

MALDI	ionizacija v nosilcu z lasersko desorpcijo (angl. matrix-assisted laser desorption/ionization)
TOF	čas preleta (angl. time-of-flight)

Povzetek

V doktorski disertaciji predstavljamo učenje finih robotskih gibov z uporabo pristopa kinestetičnega učenja. Kinestetično učenje je uveljavljen pristop učenja z demonstracijo saj operaterjem omogoča intuitivno izvedbo giba brez dodatnih krmilnih naprav. Operater lahko namreč izvede želen gib tako, da prime posamezne segmente robota in jih premakne v želeno lego. Primernost kinestetičnega učenja je tako že bila preučena v sklopu aplikacij, ki so zahtevale grobe gibe, ne pa v sklopu aplikacij, ki so zahtevale učenje finih gibov. Fini gibi so namreč gibi, pri katerih je zahtevana natančnost pozicioniranja znotraj velikostnega reda milimetra, za kar se pogosto uporablja teleoperacijo in kooperativno robotsko orodje, ki sta uveljavljena pristopa za demonstracijo finih gibov.

Tako smo v prvem delu disertacije delovanje kinestetičnega učenja primerjali z omenjenima pristopoma, pri čemer smo pristope primerjali na dveh skrbno načrtovanih nalogah. Nalogi sta se razlikovali glede na tip giba, pri čemer je prva zahtevala natančen premik od točke do točke, druga pa natančno sledenje referenčni poti. Cilj študije je bila, poleg določanja primernosti posameznega pristopa za demonstracijo finih gibov, tudi analiza vpliva vizualnih modalitet na natančnost izvedbe. Razvili smo namreč neke vrste virtualni mikroskop, ki je omogočal slikovno povečavo delovnega območja pod vrhom robota in posledično izboljšal vizualno zaznavanje pozicijskih odstopanj med izvajanjem demonstracije. Operaterji so tako izvedli demonstracije brez in z uporabo vizualne povečave.

V sklopu te študije smo vzporedno pripravili tudi manjšo študijo osredotočeno na izvajanje finih dinamičnih gibov. Pri teh gibih je za uspešno izvedbo potreben ustrezen dinamičen potek, pri čemer se sam gib izvede na relativno kratki prostorski razdalji velikostnega reda centimetra. Ugotovitve te študije so predstavljene v sklopu priloge Dodatek A, saj ugotovitve niso tako pomembne kot v primeru ostalih študij.

Nadalje smo v drugem delu doktorske disertacije preučili metode zapisa finih gibov. Poleg primerne demonstracije je ustrezen zapis demonstracij namreč druga ključna

stvar pri pristopu učenja z demonstracijo. Med seboj smo primerjali metodi DMP in GMM, ki sta uveljavljeni metodi zapisa demonstracij. Dodatno smo predlagali tudi nadgradnjo metode GMM na podlagi frekvenčne analize, ki omogoča ustrezen zapis finih gibov brez občutnega povečanja računske kompleksnosti metode.

Disertacija se zaključuje s preizkusom kinestetičnega učenja finih gibov na realnem procesu v kliničnem mikrobiološkem okolju. Z vpeljavo sodelujočih robotov v to delovno okolje je namreč potrebno vedeti, ali njihova uporaba omogoča primerljive rezultate v primerjavi z izkušenimi delavci. Tako smo uporabili sodelujočega robota za postopek zaznave, odvzema in nanosa bakterijskih kolonij v sklopu procesa identifikacije bakterijskih kolonij z uporabo masne spektrometrije. Delovanje sistema smo ocenili na podlagi rezultatov posameznega vmesnega postopka ter uspešnosti identifikacije kolonij, rezultate pa primerjali z objavljenimi podatki o uspešnosti izkušenih laboratorijskih tehnikov. Na podlagi prvih dveh študij, pa smo za namen te aplikacije pripravili tudi t.i. učni vmesnik. Vmesnik je bil sestavljen iz dveh ločenih delov. Prvi del je predstavljal obogateno okolje delovnega prostora, ki je operaterju, z uporabo očal za navidezno resničnost, omogočil bolj natančno izvedbo demonstracije. Drugi del pa je predstavljal sistem, ki je na podlagi metode DMP zapisal izvedeno demonstracijo ter jo prilagodil glede na trenutne zahteve procesa.

Ključne besede: kinestetično učenje, fini gibi, učenje z demonstracijo, sodelujoči roboti

Abstract

KINESTHETIC teaching is a well-established learning by demonstration (LfD) approach as it allows operators to intuitively generate the robot motion without additional control devices. That is so because the operator can perform the desired motion by grasping individual robot segments and moving them to the desired pose. The performance of kinesthetic teaching has thus already been studied in the context of an application requiring coarse movements, while the performance of generating fine movements has yet to be studied. Fine movements require high positional precision, for which teleoperation and cooperative robot tool are the two established LfD approaches.

Thus, in the first part of the thesis, we compare the performance of kinesthetic teaching to the two approaches mentioned above. For comparison, we carefully designed two tasks based on the required motion, with the first task requiring a precise movement from point to point and the second task requiring a precise tracking of a reference trajectory. In addition, to determine the suitability of each LfD approach for fine movements, we also analyzed the influence of visual modalities on the operator's performance. Specifically, we developed a visual enhancement tool that allowed us to visually zoom in on the work area under the robot's end-effector and consequently improve the visual detection of positioning errors during the demonstration. Thus, operators performed demonstrations using each LfD approach with and without the use of the visual enhancement tool.

As part of this study, a smaller parallel study which focused on the execution of fine dynamic movements was also performed. For these movements, the dynamics of the movement have to be appropriate in order for successful demonstration. Usually, these movements are also generated over a relatively short distance. The findings of this study are presented in Appendix A, as the findings are not as significant as it is the case with other studies.

In the second part of the thesis, we analyzed the performance of different methods which are used for demonstration generalization. Apart from an appropriate

demonstration, motion generalization is the other important issue regarding LfD. We compared DMP and GMM, which are both established methods for movement generalization. Additionally, we proposed a novel addition to the GMM method that improves the generalization of fine movements without increasing the computational complexity of the model.

The thesis concludes with a study that implements kinesthetic teaching into a real-world environment. We used a collaborative robot to detect, collect and deposit bacterial colonies as part of a bacterial colony identification process using mass spectrometry. The system's performance was evaluated based on each intermediate procedure's results and colony identification's success rate. The identification results were then compared with published data on the success rate of experienced laboratory technicians. We have also developed a so-called teaching agent for this application based on the findings from the first two studies. The agent consisted of two separate functionalities. The first was an augmented reality environment that allowed the operator to perform the demonstration more precisely using the virtual reality ovals. The second functionality was a system that, based on the DMP method, generalized the given demonstration and adapted it to the current process requirements.

Key words: kinesthetic teaching, fine movement, learning by demonstration, collaborative robots

1 Uvod

SODELUJOČI roboti so odgovor proizvajalcev na povpraševanje industrije po robotih, ki operaterju olajšajo upravljanje robota ter hkrati s tem povečajo prilagodljivost proizvodnega procesa. Takšni roboti so zasnovani za varno delo v bližini operaterja. Vgrajeni senzorji sile, poleg zaznavanja morebitnih trkov, omogočajo tudi kinestetično učenje robota. To je eden izmed pristopov učenja z demonstracijo pri katerem operater, preko ročnega premikanja robota, določa potek giba. S tem se poenostavi učenje trajektorij, postopek pa je primeren tudi za operaterje, ki imajo manj izkušenj s programiranjem robotov. Operater lahko demonstrira referenčne lege (via točke vključno z začetno in končno lego), skozi katere se mora robot gibati, ali celotne poteke trajektorij [1]. Kinestetično učenje je učinkovito za določanje referenčnih leg, kar pripomore k prilagodljivosti proizvodnega procesa [2].

Poleg kinestetičnega učenja, je pogosto uporabljen način upravljanja z robotom teleoperacija. Za razliko od kinestetičnega učenja, operater v primeru teleoperacije upravlja z robotom preko vmesne (haptične) naprave. Teleoperacija daje rezultate primerljive s kinestetičnim učenjem, a so ti odvisni od naprave, s katero operater upravlja z robotom [3]. Potrebno je poudariti, da so bile primerjave med kinestetičnim učenjem in teleoperacijo opravljene v aplikacijah, osnovanih na grobih gibih. Dovoljena odstopanja pri grobih gibih so namreč dovolj velika, da je s kinestetičnim učenjem operater zmožen prenosa giba na robota z ustrezno natančnostjo. Primeri grobih gibov so postavljanje enega objekta na drugega, zajemanje z žlico, prestavljanje tiskanega vezja, udarec žogice z namiznoteniškim loparjem in pisanje [4–9]. Pri izvajanju finih gibov, kjer je zahtevana natančnost velikostnega reda milimetra, pa se teleoperacija izkaže kot učinkovitejši pristop v primerjavi s kinestetičnim učenjem [10]. To je razvidno tudi na področju kirurške robotike, kjer se za učenje finih gibov pogosto uporablja teleoperacija [11, 12]. Izvajanju finih gibov so namenjena tudi kooperativna robotska orodja (angl. cooperative robot tools - CRT) [13, 14]. V tem primeru operater upravlja z orod-

jem, vpetim v robotski mehanizem. Med interakcijo operaterja z orodjem se izmerijo sile, na podlagi katerih se izvede premik orodja. Premiki so ustrezno skalirani glede na silo, s katero operater deluje na orodje (osnova je admitančno vodenje običajno samozapornega robotskega sistema). Tak sistem omogoča izvajanje finih gibov, saj odpravlja tresavico operaterja in vhodne sile skalira v ustrezno majhne premike [15, 16]. Zaradi potrebe po sistemu, ki omogoča učenje finih gibov, a hkrati ne zahteva uporabe dodatnih naprav za upravljanje z robotom, smo v sklopu doktorske disertacije raziskali pristope, ki omogočijo in olajšajo kinestetično učenje finih gibov. Učenje finih gibov je lažje, če so gibi robota ustrezno prostorsko skalirani glede na demonstracijo (primer teleoperacije). V primeru kinestetičnega učenja to ni mogoče, saj operater neposredno premika segmente in posledično vrh robota. Tako je potrebno uporabiti druge pristope, ki vplivajo na natančnost demonstracije operaterja. Uporaba virtualne oziroma obogatene vizualne modalnosti omogoča operaterju podrobnejši in povečan prikaz poteka finega giba, zaradi česar bolje zaznava odstopanja od želenega poteka giba. Vpliv obogatene vizualne modalnosti na kinestetično učenje finih gibov smo tako preučili tudi v sklopu doktorske disertacije.

Opravila, ki jih operater želi demonstrirati, niso nujno pogojena s potekom poti, temveč s časovnimi odvodi poti (hitrost, pospešek, sunek). Tako smo v sklopu doktorske disertacije preučili tudi primernost kinestetičnega učenja za izvedbo demonstracij finih dinamičnih gibov. To so gibi, pri katerih je pomemben dinamični potek, ki se izvede na relativno kratki poti.

Poleg izvedbe demonstracije je pri učenju z demonstracijo pomembna tudi izbira metode zapisa izbranega nabora demonstracij. Te metode so namenjene posrednemu zapisu trajektorij, posploševanju iz množice demonstracij in nadaljnjemu prilagajanju naučenih trajektorij. Razvitih je bilo že več metod (t.i. generatorji giba), med katerimi so pogosto uporabljene dinamični generatorji gibov (angl. dynamic movement primitives - DMP) in mešani Gaussovi modeli (angl. Gaussian mixture models - GMM), razvijajo pa se tudi nove, kot so jedrni generatorji gibov (angl. kernelized movement primitives - KMP) [6, 17–19]. Vsaka izmed metod ima svoje prednosti kot tudi slabosti, zato smo v sklopu doktorske disertacije raziskali primernost uporabe metod pri zapisu finih gibov.

Laboratorijsko okolje vključuje različna opravila, ki zahtevajo izvajanje finih gibov [20]. Ta opravila trenutno izvajajo laboranti oziroma naprave, prilagojene specifičnemu opravilu [21–23]. Uporabo takih naprav bi lahko smatrali že za laboratorijsko

avtomatizacijo, a temu ni tako. Laboratorijska avtomatizacija se namreč razvršča glede na stopnjo integracije večih, ločenih inštrumentov oziroma naprav. Če so te naprave samostojne se laboratorij smatra za neavtomatiziranega, v kolikor pa so naprave medsebojno povezane in delno integrirane pa že lahko govorimo o delno avtomatiziranem laboratoriju [24, 25]. O celoviti laboratorijski avtomatizaciji (angl. total laboratory automation - TLA) pa lahko govorimo takrat, ko je večina naprav, ki opravlja različne vrste testov fizično integriranih na različnih matricah vzorcev (npr. kri, serum, heparinizirana ali citrirana plazma) [26]. Taki sistemi opravljajo tudi veliko predanalitičnih in postanaličnih korakov (npr. vnos vzorca, prijava, shranjevanje, itd.), ki jih učinkovito upravljajo računalniški programi. Učinkovita implementacija TLA sistemov omogoča, kljub visokim začetnim stroškom, nižanje stroškov na dolgi časovni rok, saj v primeru testiranja vzorcev v velikem obsegu zmanjšuje potrebo po laboratorijskih tehnikih, hkrati pa niža tudi pred in postanalične stroške [27, 28]. Z učinkovito zasnovanim TLA sistemom se tako lahko zmanjša čas obdelave in posledično poveča produktivnost laboratorija [29, 30], zmanjša možnost za napake pri označevanju vzorcev in posledično poveča zanesljivost diagnoz [31]. Implementacija sistema TLA pa ima tudi svoje slabosti kot so visoki kratkoročni stroški, dodatni stroški vzdrževanja in zahteva po ustrezno velikem prostoru [32, 33]. Ob visoki stopnji avtomatizacije in velikem številu analiziranih vzorcev se pojavi tudi problem vzpostavitve in vzdrževanja delovnih procesov ob primeru kritične napake, ki zaustavi delovanje takega sistema. Končni vpliv avtomatizacije je tako odvisen od implementacije, kot tudi od izbranih procesov, zato ne obstaja enolična rešitev za izbiro stopnje avtomatizacije.

V kolikor v sklopu laboratorija ni smotrna implementacija TLA sistemov, obstaja možnost delne avtomatizacije, kjer so posamezni manjši procesi med seboj povezani in integrirani. Z razvojem sodelujočih robotov je postal razvoj delne avtomatizacije, ki zahteva uporabo robotskih manipulatorjev lažji, hkrati pa njihova implemetacija omogoča dodatno fleksibilnost procesov [34]. Takšni roboti so že bili vključeni v robotske aplikacije priprave vzorcev, kjer so izvajali pipetiranje in preprijemanje [35–37]. Vendar pa tako definirana robotska aplikacija ne omogoča prilagodljivosti različnim aplikacijam. Razviti bi bilo potrebno sistem, ki omogoča prilagajanje funkcionalnosti robota trenutnim potrebam laboratorija, kjer osrednjo enoto predstavlja sodelujoč robot okrog katerega se modularno spreminjajo tipi vzorcev in zahtevani inštrumenti [38]. Hkrati pa operaterji prilagodijo tudi orodje manipulatorja in naučijo robota nove zahtevane gibe. Ti gibi bi lahko bili naučeni z uporabo kinestetičnega učenja saj je intuitiven

za uporabo v primeru neizkušenih operaterjev [3,39,40]. Ker primernost uporabe sodelujočih robotov in kinestetičnega učenja v laboratorijskem okolju še ni bila raziskana, smo v sklopu doktorske disertacije analizirali primernost kinestetičnega učenja finih gibov v kliničnem mikrobiološkem laboratoriju, kjer smo za študijo primera izbrali odvzem kolonije bakterij za analizo z masnim spektrometrom [41].

1.1 Cilji doktorske disertacije

Učenje finih gibov z uporabo kinestetičnega učenja je neraziskan in širok problem. V sklopu doktorske disertacije smo tako izvedli primerjavo med različnimi pristopi učenja z demonstracijo finih gibov, raziskali vpliv obogatene resničnosti na kvaliteto izvedene demonstracije, izbrali ustrezne metode zapisa izvedenih demonstracij in predlagali algoritem za optimizacijo postavitev komponent mešanih Gaussovih modelov z namenom boljše zaznave finih delov giba. Poleg tega smo preizkusili primernost kinestetičnega učenja za demonstracijo finih dinamičnih gibov in vpeljali sodelujočega robota v klinični mikrobiološki proces, kjer smo primerjali učinkovitost kinestetičnega učenja z delom laboranta. Tekom teh raziskav smo izvedli sledeče vmesne cilje:

1. **Študija natančnosti in uporabniške izkušnje učenja finih robotskih gibov z različnimi pristopi učenja z demonstracijo.** V sklopu študije smo analizirali primernost kinestetičnega učenja za učenje vnaprej definiranih, finih gibov. Primerjali smo različne pristope učenja z demonstracijo (kinestetično učenje, teleoperacija, CRT) ter preučili vpliv nadgradnje interakcije med operaterjem in robotom z uporabo vizualnih modalitet. Ustreznost pristopov smo določili na osnovi natančnosti demonstriranih gibov, časa potrebnega za izvedbo učenja in uporabniške izkušnje, ki smo jo ocenjevali na podlagi vprašalnika.
2. **Študija zapisa finih robotskih gibov.** Primerjali smo različne metode zapisa in ocenili primernost posamezne metode za posplošitev finih gibov ter analizirali vpliv prilagajanja parametrov posameznih metod na kakovost zapisa. Predlagali smo tudi novo metodo namenjeno zapisu finih gibov.
3. **Študija kinestetičnega učenja finih robotskih gibov z uporabo učnega vmesnika v sklopu laboratorijske aplikacije.** V sklopu te študije smo robota umestili v izbran laboratorijski proces, z namenom nadomestitve dela operaterja ali specifičnemu opravilu namenjene naprave, pri čemer je operater naučil

robota izbrano opravilo z uporabo kinestetičnega učenja. Rezultate izvedbe laboratorijskega procesa smo primerjali z že raziskanimi rezultati ročne izvedbe laboratorijskega procesa ter na podlagi teh rezultatov določili uspešnost vpeljave sodelujočega robota v laboratorijsko okolje. Poleg tega smo preučili tudi primer-
nost uporabe obogatene vizualne modalnosti za demonstracijo finih gibov.

2 Robotski sistem

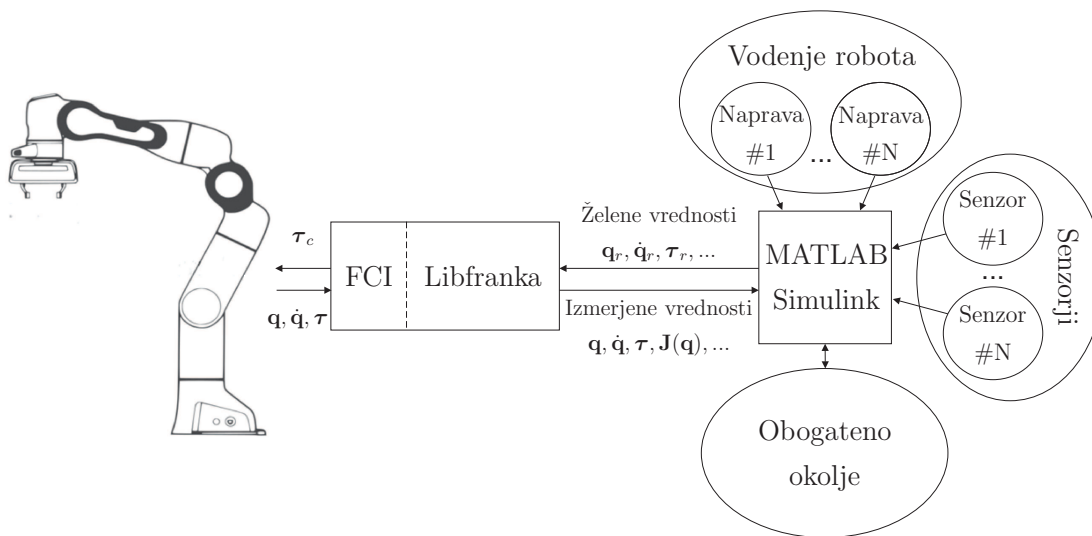
V sklopu doktorske disertacije smo uporabljali robotski manipulator Franka Emika Panda, za katerega smo implementirali ustrezen sistem vodenja in različne generatorje trajektorij, ki so omogočali zeleno gibanje robota. Robot se je namreč gibal tako na podlagi programske interpoliranega gibanja od začetne do končne točke, kot tudi na podlagi operaterjevih ukazov tekom kinestetičnega vodenja, teleoperacije, ali izvedbe kooperativnega robotskega orodja (angl. cooperative robot tool - CRT). Poleg tega smo za minimiziranje pozicijskega pogreška vrha robotskega manipulatorja kalibrirali parametre Denavit-Hartenberg (DH). Opis robotskega manipulatorja, sistem vodenja in kalibracija DH parametrov bodo tako predstavljeni tekom tega poglavja.

2.1 Sodelujoči robot Franka Emika Panda

Robotski manipulator Panda (Franka Emika, GmbH) je sodelujoč robot s sedmimi prostostnimi stopnjami, kjer vse stopnje predstavljajo rotacijski sklepi. Sedem prostostnih stopenj omogoča kinematično redundanco, zaradi katere lahko robot ohranja lego vrha kljub spreminjanju konfiguracij sklepov. Robot ima nosilnost 3 kg in maksimalen doseg 855 mm. Varno zaustavitev v primeru trka (angl. power-force limiting, ISO/TS 15066) omogočajo senzorji navora v vsakem sklepu, na podlagi katerih se zaznava potencialne trke. Ena izmed glavnih prednosti robotskih manipulatorjev je dobra ponovljivost pozicioniranja. Ponovljivost pozicioniranja vrha znaša pri robotu Panda 0,1 mm. V splošnem je točnost pozicioniranja robotskih manipulatorjev slabša. Vzrok je v odstopanju kinematičnega modela robotskega manipulatorja, ki je posledica odstopanj v strukturi robotskega manipulatorja in pogreškov pri izmerjenih pozicijah posameznih sklepov. Za izboljšanje točnosti robota Panda smo izvedli kalibracijo DH parametrov (predstavljeno v poglavju 2.3), s katero smo dosegli točnost pozicioniranja 0,1 mm.

Krmilnik manipulatorja je namenjen generiranju ustreznih tokov za aktuacijo po-

sameznih sklepov robota. Nadzorovan je z računalnikom preko Ethernet povezave. Za vodenje robota smo uporabili knjižnico *libfranka* podano s strani proizvajalca Franka Emika in osnovano na programskem jeziku C++, ki je omogočala komunikacijo s programsko zaprtim delom krmilnika. Tako imenovan FCI (angl. Franka Control Interface) je zaprti vmesnik na strani krmilnika, ki zagotavlja realnočasno izvedbo ukazov pri frekvenci 1 kHz in branje trenutnih vrednosti kinematičnih in dinamičnih spremenljivk robota, kot so pozicije \mathbf{q} , hitrosti $\dot{\mathbf{q}}$ in navori $\boldsymbol{\tau}$ v posameznih sklepih (Slika 2.1).



Slika 2.1: Shematski prikaz splošnega eksperimentalnega sistema, ki prikazuje hierarhijo ter tok podatkov med robotom, krmilnikom, napravami za vodenje robota (npr. haptični vmesnik pri teleoperaciji), specifičnimi senzorji za delovanje aplikacije (npr. RGB kamera) in obogatenim okoljem potrebnim za ustrezen prikaz poteka aplikacije.

Za namene visokonivojskega odločanja in določanja referenčnih vrednosti (npr. končna lega, potek trajektorije) smo uporabili okolje MATLAB Simulink 2019b (The MathWorks, Inc.), ki je uveljavljeno programsko orodje za namene bločnega programiranja in je zaradi široke funkcionalnosti pogosto uporabljeno na področju robotike. Uporabili smo ga namreč ne le za nadzorovanje robota, ampak tudi za komunikacijo in nadzor ostalih sestavnih delov aplikacije, ki so bili namenjeni vodenju robota (senzor sile in haptični vmesnik), zaznavi okolice za namene aplikacije (RGB kamera, laserski skener), ali prikazovanju uporabniškega vmesnika z igralnega pogona Unity (Unity Technologies, Inc.). Za komunikacijo prek omrežja Ethernet smo izbrali protokol UDP, saj omogoča konsistentno pošiljanje in branje podatkov, ki je ključno za

robotsko aplikacijo s frekvenco regulacijske zanke 1 kHz.

2.2 Vodenje robotov

Gibanje posameznih segmentov robotskega manipulatorja je posledica generiranih tokov v motorjih, ki zagotavljajo doseganje želenega navora v posameznih sklepih. Zato je za nadzorovano gibanje robota pomembno, da so želeni navori določeni pravilno. Trajektorija gibanja je lahko generirana programsko (npr. pri gibu od točke do točke po principu minimalnega sunka) ali pa je generirana na podlagi signalov iz nadzornih naprav (npr. pri teleoperaciji). V obeh primerih je potrebno uporabiti ustrezen sistem vodenja, ki želene lege pretvori v sklepne napore, potrebne za doseg lege.

Tekom doktorske disertacije smo gibanje robota generirali na podlagi različnih pristopov, ki so bili potrebni za namene učenja z demonstracijo (kinestetično vodenje, teleoperacija, kooperativno robotsko orodje) in za programsko generiranje premikov vrha robota v želeno lego. Sistem vodenja in izbrane pristope generiranja trajektorij bomo tako predstavili v sledečih podpoglavjih.

2.2.1 Sistem vodenja

Odvisnost med gibanjem posameznih segmentov robotskega manipulatorja in generiranimi tokovi (ter posledično navori) v posameznih sklepih je pogojena z mehansko strukturo manipulatorja in opisana z dinamičnim modelom robota. Zvezo lahko tako opišemo z enačbo inverznega dinamičnega modela

$$\mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}_v\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}, \quad (2.1)$$

kjer $\mathbf{B}(\mathbf{q})$ predstavlja vztrajnostno matriko manipulatorja, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ matriko Coriolisovih in centripetalnih prispevkov, \mathbf{F}_v matriko elementov viskoznega dušenja, $\mathbf{g}(\mathbf{q})$ vektor gravitacijskih navorov in $\boldsymbol{\tau}$ vektor generiranih sklepnih navorov. Za boljšo predstavo o medsebojni odvisnosti lahko nadalje iz enačbe (2.1) izrazimo pospešek

$$\ddot{\mathbf{q}} = \mathbf{B}^{-1}(\mathbf{q})(\boldsymbol{\tau} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{F}_v\dot{\mathbf{q}} - \mathbf{g}(\mathbf{q})). \quad (2.2)$$

Za želeno gibanje manipulatorja, je tako potrebno ustrezno določiti potrebne napore $\boldsymbol{\tau}$. V sklopu doktorske disertacije smo potrebne napore izračunavali na podlagi odstopanj

od zelene lege vrha robota (v t.i. zunanjih koordinatah) z uporabo proporcionalno-diferencirnega (PD) regulatorja s kompenzacijo gravitacije ter prispevka Coriolisovih in centripetalnih sil (Slika 2.2). Potrebne navore $\boldsymbol{\tau}$ lahko zapišemo kot

$$\boldsymbol{\tau} = \mathbf{J}_A^T(\mathbf{q})(\mathbf{K}_p \tilde{\mathbf{x}} - \mathbf{K}_d \mathbf{J}_A(\mathbf{q})\dot{\mathbf{q}}) + \widehat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \widehat{\mathbf{g}}(\mathbf{q}), \quad (2.3)$$

kjer $\tilde{\mathbf{x}}$ predstavlja odstopanje med želeno \mathbf{x}_r in dejansko lego vrha robota \mathbf{x}

$$\tilde{\mathbf{x}} = \mathbf{x}_r - \mathbf{x}, \quad (2.4)$$

$\mathbf{J}_A(\mathbf{q})$ predstavlja analitično Jacobijevo matriko, medtem ko $\widehat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})$ predstavlja model prispevkov Coriolisovih in centripetalnih sil, $\widehat{\mathbf{g}}(\mathbf{q})$ pa model prispevkov gravitacijskih sil. V primeru robota Panda smo imeli dostop do modelov $\widehat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})$ in $\widehat{\mathbf{g}}(\mathbf{q})$, ki sta bila podana s strani proizvajalca. Matriki \mathbf{K}_p in \mathbf{K}_d predstavljata diagonalni matriki proporcionalnih in diferencirnih ojačenj. Ker je lega vrha robota \mathbf{x} v splošnem opisana s pozicijo \mathbf{p} in orientacijo $\boldsymbol{\phi}$

$$\mathbf{x} = \begin{bmatrix} \mathbf{p} \\ \boldsymbol{\phi} \end{bmatrix}, \quad (2.5)$$

sta tudi matriki \mathbf{K}_p in \mathbf{K}_d sestavljeni iz dveh podmatrik pozicijskih in orientacijskih ojačenj

$$\mathbf{K}_p = \begin{bmatrix} \mathbf{K}_{pp} & 0 \\ 0 & \mathbf{K}_{p\phi} \end{bmatrix} \quad \mathbf{K}_d = \begin{bmatrix} \mathbf{K}_{dp} & 0 \\ 0 & \mathbf{K}_{d\phi} \end{bmatrix}. \quad (2.6)$$

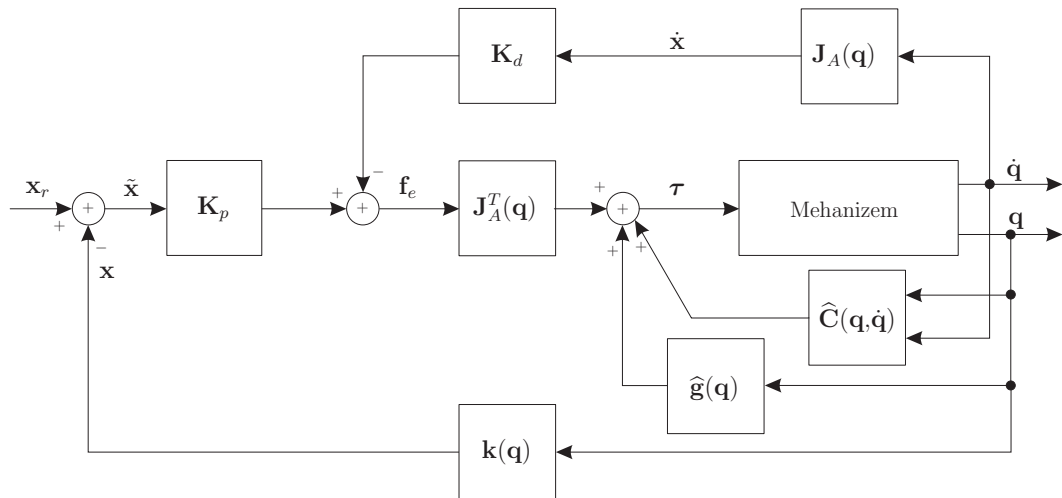
Diagonalnim elementom matrike \mathbf{K}_{pp} smo za privzeto vrednost določili 1000 N/m, medtem ko smo elementom matrike $\mathbf{K}_{p\phi}$ za privzeto vrednost določili 100 Nm/rad. Privzete vrednosti istoležnih elementov matrike \mathbf{K}_d pa smo nadaljnje določili na podlagi enačbe

$$K_{d_{ii}} = 2\sqrt{\frac{K_{p_{ii}}}{10}}. \quad (2.7)$$

Če nazadnje združimo enačbi (2.2) in (2.3), dobimo izraz

$$\ddot{\mathbf{q}} = \mathbf{B}^{-1}(\mathbf{q}) \left(\mathbf{J}_A^T(\mathbf{q})(\mathbf{K}_p \tilde{\mathbf{x}} - \mathbf{K}_d \mathbf{J}_A(\mathbf{q})\dot{\mathbf{q}}) - \mathbf{F}_v \dot{\mathbf{q}} + \widehat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \widehat{\mathbf{g}}(\mathbf{q}) - \mathbf{g}(\mathbf{q}) \right), \quad (2.8)$$

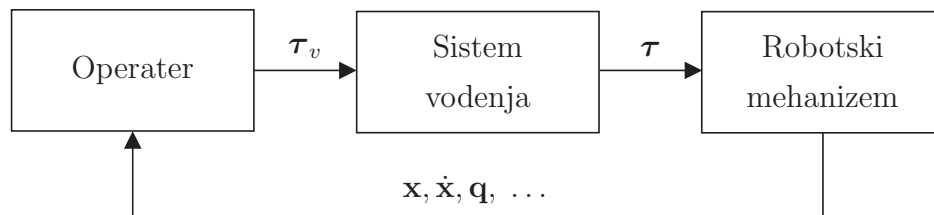
iz katerega je razvidno, da je gibanje robota, poleg parametrov PD regulatorja, pogojeno zgolj še z vztrajnostjo robota $\mathbf{B}(\mathbf{q})$ in prispevkom viskoznega dušenja \mathbf{F}_v . To seveda velja, v kolikor $\widehat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})$ in $\widehat{\mathbf{g}}(\mathbf{q})$ predstavljata popoln opis $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ in $\mathbf{g}(\mathbf{q})$.



Slika 2.2: PD regulacija lege v zunanjih koordinatah s kompenzacijo gravitacije ter Coriolisovih in centripetalnih sil.

2.2.2 Kinestetično vodenje

V primeru kinestetičnega vodenja operater nadzoruje robota tako, da prime posamezne segmente robota in jih premakne, da izvede želen gib ali doseže želeno lego. Tako nadzoruje gibanje robota neposredno, brez uporabe dodatnih upravljalnih naprav (Slika 2.3). Robot mora biti zato podajen silam in navorom τ_v , ki jih operater vsiljuje na posamezne dele robota, hkrati pa ohranjati lego, ko operater nanj ne vsiljuje sil.



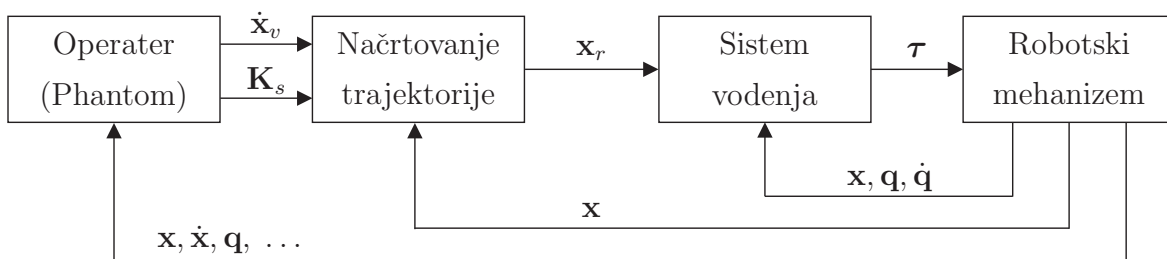
Slika 2.3: Shematski prikaz operaterjevega nadziranja gibanja robota s pristopom kinestetičnega vodenja.

Za izvedbo kinestetičnega vodenja v sklopu predstavljenega sistema vodenja (Slika 2.2), definicija referenčne lege \mathbf{x}_r tako ni relevantna, vrednost diagonalnih elementov matrik proporcionalnega \mathbf{K}_p in diferencirnega \mathbf{K}_d ojačenja pa je enaka 0. V sklopu naših študij pa smo kinestetično učenje delno omejili in operaterju onemogočili spreminjanje orientacije vrha. V teh primerih smo vrednosti elementov matrik $\mathbf{K}_{p\phi}$ in $\mathbf{K}_{d\phi}$ pustili privzete, medtem ko smo vrednosti elementov v matrikah \mathbf{K}_{pp} in \mathbf{K}_{dp}

nastavili na 0. Prav tako smo ob začetku študij definirali referenčno lego \mathbf{x}_r , ki je opisovala želeno orientacijo vrha.

2.2.3 Teleoperacija

Teleoperacija je način vodenja, pri katerem operater nadzoruje gibanje robota z uporabo vmesne naprave, navadno haptičnega vmesnika (Slika 2.4). Njena poglavitna prednost je, da se na robota prenese prostorsko skaliran gib, ki ga operater izvede na vmesniku. To omogoča, da se na robota prenesejo gibi majhnih prostorskih razsežnosti (tudi znotraj velikostnega reda milimetra), za katere ni nujno, da jih je operater zmožen izvesti brez uporabe vmesnika. Omogoča tudi minimizacijo prenosa operaterjeve tresavice na robota, zaradi česar je pristop uveljavljen tudi v kirurški robotiki [11, 12].



Slika 2.4: Shematski prikaz operatorjevega nadzora gibanja robota s pristopom teleoperacije.

Skalirni faktor med demonstriranim gibom in gibom prenesenim na izvršno napravo je lahko konstanten ali spremenljiv. V sklopu doktorske disertacije smo implementirali spremenljiv skalirni faktor, saj nam je to omogočilo demonstracijo tako daljšega giba, kjer natančnost ni bila zahtevana, kot tudi krajšega, kjer je bila zahtevana submilimetrski natančnost. Ker smo želeli, da operater vpliva na skalirni faktor tekom same demonstracije, smo za nastavljanje le tega uporabili krmilno palico z eno prostostno stopnjo, s pomočjo katere je operater lahko spreminjal skalirni faktor od minimalne do maksimalne vrednosti.

Za upravljanje z robotom Panda smo uporabili haptični vmesnik Phantom Premium 1.5 (3D Systems, Inc.), ki je nesamozaporni robotski manipulator s tremi prostostnimi stopnjami. Za generiranje navorov v sklepih uporablja tetivni pogon, zaradi česar mehanizem odlikuje nizko trenje, ki je ena izmed ključnih zahtev haptičnih vmesnikov.

V primeru teleoperacije referenčno lego sistema vodenja \mathbf{x}_r tako definiramo kot

$$\mathbf{x}_r(t) = \int \mathbf{K}_s(t) \dot{\mathbf{x}}_v(t) dt + \mathbf{x}(0), \quad (2.9)$$

kjer $\mathbf{x}_r(t)$ predstavlja želeno lego robota Panda, $\dot{\mathbf{x}}_v(t)$ hitrost gibanja robota Phantom, $\mathbf{K}_s(t)$ matriko skalirnih faktorjev in $\mathbf{x}(0)$ izmerjeno začetno lego robota Panda. Operater je pri upravljanju spreminjal zgolj pozicijo vrha robota Phantom, zaradi česar je hitrost gibanja definirana kot

$$\dot{\mathbf{x}}_v(t) = \begin{bmatrix} \dot{\mathbf{p}}_v(t) \\ 0 \end{bmatrix}. \quad (2.10)$$

Prav tako je parameter $\mathbf{K}_s(t)$ definiran kot diagonalna matrika s šestimi komponentami, kar nam je omogočalo ločeno nastavljanje skalirnih faktorjev za posamezno prostostno stopnjo gibanja v zunanjih koordinatah

$$\mathbf{K}_s(t) = \begin{bmatrix} \mathbf{K}_{sp}(t) & 0 \\ 0 & \mathbf{K}_{s\phi}(t) \end{bmatrix}. \quad (2.11)$$

V sklopu doktorske disertacije smo tako vrednosti elementov $K_{s\phi_{ii}}(t)$ nastavili na 0, medtem ko so bile vrednosti elementov $K_{sp_{ii}}(t) \in [0, 1]$ odvisne od lege krmilne palice.

Tako robot Phantom kot tudi krmilna palica sta bila priključena na industrijski računalnik xPC Target, ki je zajemal podatke s frekvenco 5 kHz in te posredoval ostalim sistemom z uporabo protokola UDP.

2.2.4 Kooperativno robotsko orodje

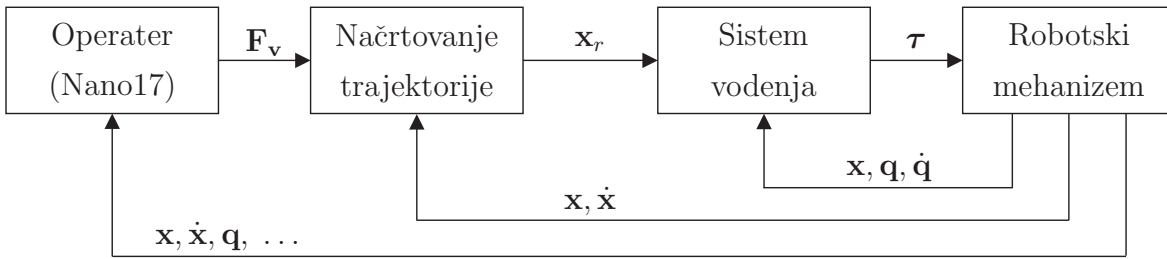
Kooperativno robotsko orodje je način vodenja, pri katerem se robotski mehanizem obnaša kot samozaporni sistem, nadzorovan z admitančno shemo vodenja. Operater namreč nadzoruje gibanje robota z generiranjem sil na senzor sile (Slika 2.5).

Hitrost gibanja vrha robota Panda smo tako določili na podlagi izmerjenih sil kot

$$\dot{\mathbf{x}}_r(t) = \int \mathbf{m}^{-1} (\mathbf{F}_v(t) - \mathbf{b}\dot{\mathbf{x}}(t)) dt, \quad (2.12)$$

kjer $\dot{\mathbf{x}}_r(t)$ predstavlja želeno hitrost gibanja robota Panda, $\dot{\mathbf{x}}(t)$ izmerjeno hitrost gibanja vrha robota in $\mathbf{F}_v(t)$ vektor izmerjenih sil in navorov generiranih na senzor sile

$$\mathbf{F}_v(t) = \begin{bmatrix} \mathbf{F}_p(t) \\ \boldsymbol{\tau}(t) \end{bmatrix}, \quad \dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{\mathbf{p}}(t) \\ \dot{\boldsymbol{\phi}}(t) \end{bmatrix}. \quad (2.13)$$



Slika 2.5: Shematski prikaz operaterjevega nadzora gibanja robota s pristopom kooperativnega robotskega orodja.

Nadalje parametra \mathbf{m} in \mathbf{b} predstavljata vztrajnost in dušenje admitančnega vodenja [13, 14]. Tako \mathbf{m} kot \mathbf{b} sta diagonalni matriki s šestimi komponentami, kar nam je omogočalo ločeno nastavljanje parametrov vztrajnosti in dušenja za posamezno prostostno stopnjo gibanja v zunanjih koordinatah

$$\mathbf{m} = \begin{bmatrix} \mathbf{m}_p & 0 \\ 0 & \mathbf{m}_\phi \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}_p & 0 \\ 0 & \mathbf{b}_\phi \end{bmatrix}. \quad (2.14)$$

S prilagajanjem vrednosti parametrov smo vplivali na dinamiko odziva med izmerjenimi silami $\mathbf{F}_v(t)$ in referenčno trajektorijo $\mathbf{x}_r(t)$, ki smo jo določili kot

$$\mathbf{x}_r(t) = \int \dot{\mathbf{x}}_r(t) dt + \mathbf{x}(0). \quad (2.15)$$

Tak pristop generiranja referenčne lege $\mathbf{x}_r(t)$ nam je omogočal odpravo tresavice ter skaliranje sil v poljubno majhen premik, kar je zaželeno za demonstracijo finih gibov.

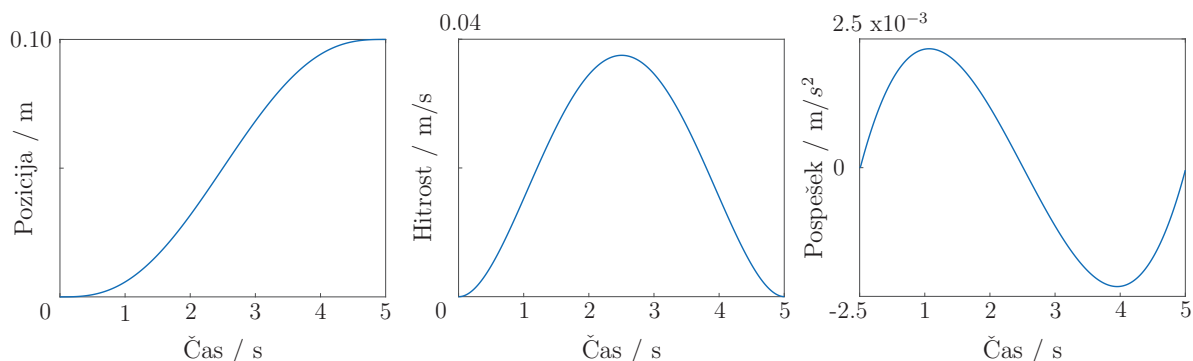
V sklopu študij, smo pri uporabi tega načina delovanja želeli spreminjati zgolj pozicijo vrha robota, medtem ko smo njegovo orientacijo želeli držati konstantno. To smo dosegli z ustreznim nastavljanjem elementov matrik \mathbf{m}_ϕ in \mathbf{b}_ϕ , pri čemer so bile vrednosti diagonalnih elementov $\mathbf{b}_{\phi_{ii}}$ enake 0, medtem ko bi morale v teoriji vrednosti diagonalnih elementov $\mathbf{m}_{\phi_{ii}}$ znašati ∞ , a to v praksi ni možno, zato smo jih nastavili na ustrezno visoko vrednost.

Za merjenje sile smo uporabili senzor Nano17 (ATI Industrial Automation, Inc.), ki ga odlikuje občutljivost primerna za demonstracijo finih gibov. Ta je bil integriran v orodje na vrhu robota Panda. Tako je operater za izvajanje gibov, podobno kot pri kinestetičnem načinu, prijel robota, le da se v tem primeru manipulator ni odzival na vsiljene sile po posameznih sklepih in segmentih, ampak zgolj na sile vsiljene na senzor sile.

2.2.5 Programsko generiranje gibov

Kinestetično učenje, teleoperacija in kooperativna robotska orodja so načini vodenja robota, pri katerih je trajektorija giba določena s strani operaterja, ki neposredno nadzoruje gibanje robota. Ti načini vodenja so smiselni, ko je za gibanje robota potreben specifičen potek trajektorije. Pogosto pa je za generiranje giba potek trajektorije nepomemben in je za aplikacijo pomembno le, da robot doseže končno lego. V tem primeru za generiranje trajektorije uporabimo interpolatorje giba. Ti, na podlagi začetne, končne in trenutne lege ter podanih zahtev glede dinamike giba, iterativno posodablajo referenčno vrednost, ki je, v primeru načrtovanja giba v zunanjih koordinatah, zelena lega vrha robota $\mathbf{x}_r(t)$.

V sklopu doktorske disertacije smo implementirali interpolator v zunanjih koordinatah, ki deluje na osnovi minimiziranja sunka in omogoča generiranje gladkega časovnega poteka pozicije, hitrosti in pospeška giba (Slika 2.6). Minimiziranje sunka



Slika 2.6: Prikaz gladkosti časovnega poteka pozicije, hitrosti in pospeška pri uporabi metode minimalnega sunka.

lahko opišemo s cenilko

$$C(\mathbf{x}) = \int_{t_0}^{t_f} \left(\frac{d^3 \mathbf{x}}{dt^3} \right)^2 dt. \quad (2.16)$$

Rešitev tega optimizacijskega problema rezultira v polinomu pete stopnje

$$\mathbf{x}_r(t) = \mathbf{x}_0 + (\mathbf{x}_f - \mathbf{x}_0)(6\tau^5 - 15\tau^4 + 10\tau^3), \quad \tau = \frac{t}{t_f - t_0}, \quad (2.17)$$

kjer \mathbf{x}_f in \mathbf{x}_0 predstavljata začetno in končno lego, t_0 in t_f začetni in končni čas giba, τ pa predstavlja normiran čas oziroma fazo [42, 43]. Hitrost gibanja $\dot{\mathbf{x}}_r$ lahko tako nadalje zapišemo kot

$$\dot{\mathbf{x}}_r(t) = (\mathbf{x}_f - \mathbf{x}_0) \frac{30\tau^4 - 60\tau^3 + 30\tau^2}{t_f - t_0}. \quad (2.18)$$

Podobno definiramo potek orientacije vrha robota, za kar moramo določiti os rotacije in potreben kot rotacije. Os rotacije φ določimo iz rotacijskih matrik začetne lege \mathbf{R}_0 in končne lege \mathbf{R}_f . Za to najprej določimo matriko pogreška orientacije \mathbf{R}_e

$$\mathbf{R}_e = \mathbf{R}_f \mathbf{R}_0^T. \quad (2.19)$$

Potreben kot rotacije nato določimo kot

$$\vartheta = \arccos \frac{\text{tr}\{\mathbf{R}_e\} - 1}{2}, \quad (2.20)$$

kjer $\text{tr}\{\mathbf{R}_e\}$ predstavlja sled matrike \mathbf{R}_e (vsota elementov diagonale). Vektor osi rotacije φ lahko nato določimo kot

$$\varphi = \frac{1}{2 \sin \vartheta} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}, \quad (2.21)$$

kjer r_{ij} predstavlja element matrike \mathbf{R}_e . S poznavanjem osi rotacije φ , potrebnega kota rotacije ϑ in uporabo metode minimalnega sunka tako sproti izračunavamo potrebno spremembo orientacije, da v času t_f vrh robota doseže želeno orientacijo.

2.3 Kalibracija kinematičnega modela

V sklopu nekaterih študij doktorske disertacije so operaterji morali generirati gibe na podlagi reference, definirane v globalnem koordinatnem sistemu. Za ustrezno primerjavo izvedenih gibov z referenco, je tako nujno, da se izračunane lege vrha robotskega manipulatorja skladajo z referenčnimi. Lego vrha robota izračunamo z uporabo kinematičnega modela, ki je podan s strani proizvajalca in omogoča izračunavanje na podlagi izmerjenih vrednosti v sklepih.

Tekom testiranja pa smo ugotovili, da ob postavitvi vrha robota v isto lego, a ob drugačnih konfiguracijah sklepov, prihaja tudi do 5 mm odstopanj pri izračunani legi vrha robota. To nakazuje na nezadosten kinematičen model, ki je bil podan s strani proizvajalca. Zaradi toleranc pri mehanski konstrukciji in sestavljanju, lahko pride do pogreškov pri podanem kinematičnem modelu. Majhna odstopanja razporejena vzdolž mehanizma pa lahko zaradi odprte kinematične verige privedejo do signifikantnih pogreškov pri izračunani legi. Ker so taka odstopanja nesprijemljiva za izvajanje finih gibov, smo izvedli kalibracijo kinematičnega modela.

Uveljavljen pristop opisa kinematičnega modela je metoda Denavit-Hartenberg (DH) [44], ki uporablja nabor strukturnih parametrov in izmerjene vrednosti v sklepih

$$\boldsymbol{\varphi} = \begin{bmatrix} \boldsymbol{\alpha} & \mathbf{a} & \mathbf{d} & \boldsymbol{\theta} & \mathbf{q} \end{bmatrix}^\top, \quad (2.22)$$

na podlagi katerih je izračunana lega vrha robota

$$\mathbf{x} = \mathbf{f}(\boldsymbol{\varphi}). \quad (2.23)$$

Parametri $\boldsymbol{\alpha}$, \mathbf{a} in \mathbf{d} temeljijo na strukturi robota, medtem ko parameter $\boldsymbol{\theta}$ predstavlja odstopanja izmerjenih pozicij sklepov \mathbf{q} , ki so posledica mehanskega zamika pozicijskih enkoderjev. Zaradi konstrukcijskih netočnosti ali merilne pristranskosti pri izmerjenih položajih sklepov, lahko trenutni \mathbf{x}_i odstopa od prave vrednosti \mathbf{x}_{ref_i} . V kolikor je prava vrednost vrha robota znana, lahko odstopanja izračunane lege vrha robota v i -ti primerjavi opredelimo kot

$$\Delta \mathbf{x}_i = \mathbf{x}_{ref_i} - \mathbf{x}_i, \quad (2.24)$$

kar lahko zapišemo tudi kot

$$\Delta \mathbf{x}_i = \frac{\partial \mathbf{f}}{\partial \boldsymbol{\alpha}} \Delta \boldsymbol{\alpha} + \frac{\partial \mathbf{f}}{\partial \mathbf{a}} \Delta \mathbf{a} + \frac{\partial \mathbf{f}}{\partial \mathbf{d}} \Delta \mathbf{d} + \frac{\partial \mathbf{f}}{\partial \boldsymbol{\theta}} \Delta \boldsymbol{\theta} + \frac{\partial \mathbf{f}}{\partial \mathbf{q}_i} \Delta \mathbf{q}_i. \quad (2.25)$$

Enačbo (2.25) lahko nadalje zapišemo s produktom matrike $\boldsymbol{\phi}_i$ in vektorja $\Delta \boldsymbol{\varphi}_i$

$$\boldsymbol{\phi}_i = \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial \boldsymbol{\alpha}} & \frac{\partial \mathbf{f}}{\partial \mathbf{a}} & \frac{\partial \mathbf{f}}{\partial \mathbf{d}} & \frac{\partial \mathbf{f}}{\partial \boldsymbol{\theta}} & \frac{\partial \mathbf{f}}{\partial \mathbf{q}_i} \end{bmatrix}, \quad (2.26)$$

$$\Delta \boldsymbol{\varphi}_i = \begin{bmatrix} \Delta \boldsymbol{\alpha} & \Delta \mathbf{a} & \Delta \mathbf{d} & \Delta \boldsymbol{\theta} & \Delta \mathbf{q}_i \end{bmatrix}^\top, \quad (2.27)$$

$$\Delta \mathbf{x}_i = \boldsymbol{\phi}_i \Delta \boldsymbol{\varphi}_i. \quad (2.28)$$

V sklopu procesa kalibracije opravimo N meritev, za vsako določimo $\Delta \mathbf{x}_i$ in nadalje definiramo

$$\Delta \mathbf{x} = \begin{bmatrix} \Delta \mathbf{x}_1 \\ \Delta \mathbf{x}_2 \\ \dots \\ \Delta \mathbf{x}_N \end{bmatrix}, \quad \boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\phi}_1 \\ \boldsymbol{\phi}_2 \\ \dots \\ \boldsymbol{\phi}_N \end{bmatrix}, \quad (2.29)$$

$$\Delta \mathbf{x} = \boldsymbol{\Phi} \Delta \boldsymbol{\varphi}, \quad (2.30)$$

kjer enačba (2.30) predstavlja, v kolikor je N dovolj velik, predeterminiran sistem enačb. Tako z uporabo levega pseudoinverza določimo

$$\Delta \boldsymbol{\varphi} = \boldsymbol{\Phi}^+ \Delta \mathbf{x} \quad (2.31)$$

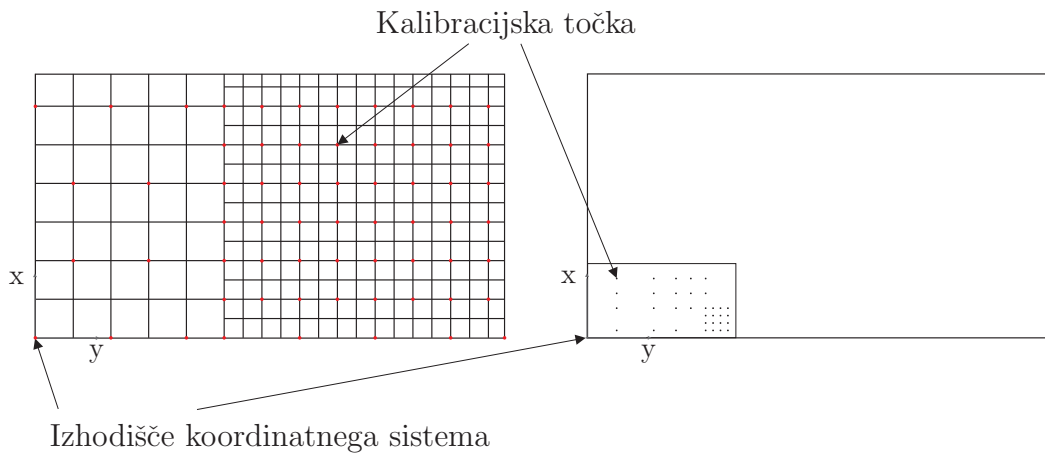
in $\Delta\varphi$ uporabimo za posodobitev obstoječih DH parametrov

$$\varphi' = \varphi + \Delta\varphi, \quad (2.32)$$

kar posledično privede do posodobitve vrednosti enačbe (2.30)

$$\Delta\mathbf{x}' = \Phi' \Delta\varphi. \quad (2.33)$$

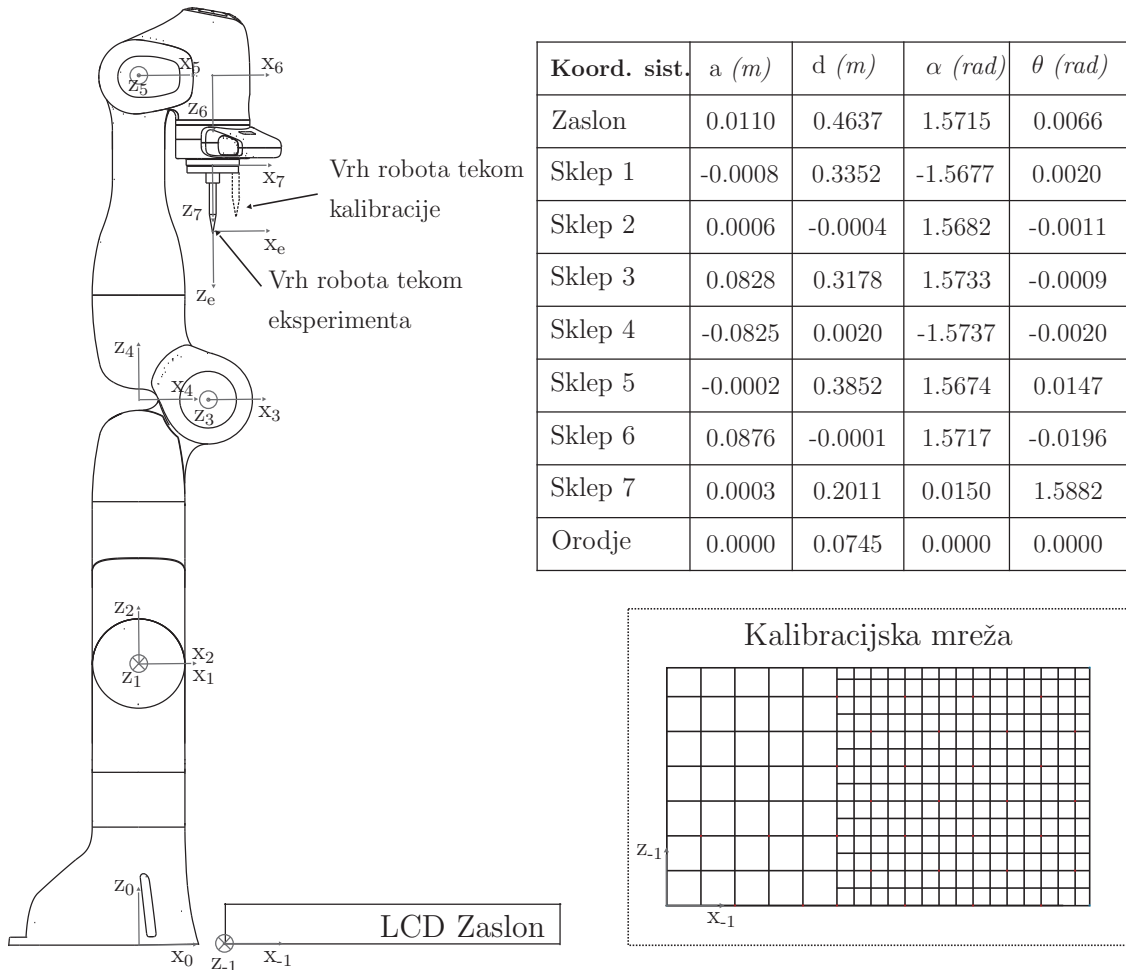
Ta postopek iskanja rešitve ($\Delta\varphi$) in posodabljanja parametrov (φ') izvajamo iterativno, dokler se srednja vrednost pogreškov $\Delta\mathbf{x}$ ne zmanjša pod želeno vrednost. Tako smo v našem primeru iterativni postopek ponavljali, dokler srednja vrednost pogreškov $\Delta\mathbf{x}$ ni bila manjša od 0,1 mm.



Slika 2.7: Prikaz različnih kalibracijskih mrež glede na študijo. Leva slika prikazuje kalibracijsko mrežo študije demonstracije finih gibov, ki je bila prikazana na LCD zaslonu, medtem ko desna slika prikazuje kalibracijsko mrežo študije kinestetičnega učenja v sklopu laboratorijske aplikacije, ki je bila definirana z izvrtinami na aluminijasti plošči.

Za referenčno merjenje lege vrha robota se v splošnem lahko uporablja sisteme za vizualno zajemanje [45]. V našem primeru smo se odločili za drugačen pristop, saj smo v delovnem prostoru definirali mrežo referenčnih točk, za katere smo poznali njihovo točno pozicijo. Te točke so bile definirane različno glede na študijo. V prvi študiji smo definirali in prikazali 63 točk na LCD zaslonu, medtem ko smo pri ostalih definirali 33 izvrtin na aluminijasti plošči (Slika 2.7). Tekom postopka kalibracije smo tako vrh robota postavili točno na pozicijo posamezne referenčne točke ter shranili vrednosti izmerjene pozicije v sklepkih \mathbf{q}_i , izračunano pozicijo vrha robota \mathbf{x}_i in pozicijo referenčne točke \mathbf{x}_{ref_i} . Shranjene vrednosti smo uporabili v zgoraj omenjenem algoritmu

za izračun novih DH parametrov (Slika 2.8). Za ustrezen opis sistema smo DH parametrom robota dodali še dodatni koordinatni sistem (\mathbf{T}_{-1}), ki je definiral transformacijo med koordinatnim sistemom mreže referenčnih točk in koordinatnim sistemom baze robota.



Slika 2.8: Prikaz mehanske strukture robota Panda, pripadajočih koordinatnih sistemov ter izračunanih vrednosti DH parametrov za primer študije finih gibov.

Proces kalibracije se je delno razlikoval tudi glede na zasnovo orodja robota. V študiji učenja finih gibov smo uporabili orodje robotskega manipulatorja, čigar vrh je ležal vzdolž osi rotacije zadnjega sklepa (Slika 2.8). Zaradi te postavitve, kalibracija sklepa 7 ni bila možna, saj je bila pozicija vrha robota nespremenjena ne glede na vrednost sklepa 7. Zato smo za namene kalibracije tega orodja prilagodili orodje tako, da je ta vseboval dodatno konico, ki je bila usmerjena v smeri osi rotacije sklepa 7, a je bila hkrati za 5 cm izmaknjena iz osi rotacije. Kalibracija je tako potekala v dveh korakih. V prvem koraku smo zajeli meritve z orodjem z izmaknjeno konico iz

osi rotacije zadnjega sklepa, kar je omogočalo natančno kalibracijo parametrov vseh segmentov razen zadnjega. Drugi set meritev pa smo izvedli z neizmaknjeno konico, kjer smo kalibracijo izvedli z uporabo DH parametrov pridobljenih iz prve kalibracije ter tokrat posodabljali zgolj parametre zadnjega segmenta (tj. orodja). Tako smo pridobili pravilne DH parametre, kar nam je omogočalo natančno poznavanje vrha robota in posledično ustrezno izvedbo študij.

Pri tem se moramo zavedati, da v sklopu procesa kalibracije prihaja do različnih vplivov, ki doprinesejo k merilni negotovosti sistema in nenazadnje negotovosti kalibriranih DH parametrov. Ti vplivi so lahko posledica prikaza slikovnih pikslov, izvrtin kalibracijske plošče in nenazadnje mehanske togosti orodja robotskega manipulatorja. Tekom izvedbe kalibracije smo lahko, s previdno izvedbo pozicioniranja robotskega manipulatorja, minimizirali zgolj doprinos slednjega.

3 Demonstriranje finih robotskih gibov

UČENJE z demonstracijo je princip učenja robotskega giba, pri katerem operater gib demonstrira, izvedena demonstracija oziroma set demonstracij pa je nato posplošen z uporabo namenskih metod. Kljub temu, da je učenje z demonstracijo uveljavljen in dobro raziskan princip, še ni bila izvedena primerjava primernosti različnih pristopov izvajanja demonstracij, ki zahtevajo sub-milimetrsko natančnost pozicioniranja. V tem poglavju se bomo osredotočili na demonstracijo finih gibov. Pri učenju z demonstracijo so pogosti pristopi prenosa giba na robota kinestetično učenje, teleoperacija in kooperativna robotska orodja (angl. cooperative robot tool - CRT). V uvodnem poglavju smo že podali pregled področja in predstavili prednosti posameznega pristopa, medtem ko je bilo njihovo podrobno delovanje predstavljeno v poglavju 2.2. Kljub številnim prednostim kinestetičnega učenja, ni jasno ali je primerno za demonstracijo finih gibov. V tej študiji smo primerjali kinestetično učenje z dvema uveljavljenima pristopoma, ki se pogosto uporabljata tudi pri demonstraciji finih gibov (teleoperacija in CRT).

Primerjavo smo izvedli na dveh različnih nalogah. Pri prvi nalogi je moral operater premakniti vrh robota od začetne do posamezne končne točke tako, da je gib končal znotraj radija 0,5 mm od končne točke. Pri tem je bila lega začetne točke konstantna, lega končne točke pa se je tekom iteracij demonstracij spreminjala med petimi predhodno definiranimi legami. Pri drugi nalogi pa je moral operater vrh robota pomikati vzdolž poti širine 0,5 mm. S prvo nalogo smo lahko določili sposobnost operaterja pri natančnem pozicioniranju vrha robota, pri čemer je vmesni potek giba irelevanten, medtem ko smo pri drugi lahko določili sposobnost natančne izvedbe celotnega giba. Pripravili smo tudi tretjo nalogo, katere namen je bila analiza primernosti omenjenih pristopov za demonstracijo finih dinamičnih gibov. Zaradi manjšega obsega študije

in vsebinske drugačnosti, pa rezultati in zaključki tretje naloge niso predstavljeni v tem poglavju, temveč v sklopu poglavja Dodatek A. Pri vseh nalogah smo vizualne reference (končna točka in pot giba) prikazali na vodoravno postavljenem LCD zaslonu. V sklopu študije smo preizkusili tudi vpliv vizualnega in prostorskega skaliranja na natančnost operaterja. Zasnovali smo funkcijo vizualnega skaliranja, da operaterju omogočimo skaliran prikaz območja pod vrhom robota in posledično izboljšamo njegovo vizualno zaznavanje in občutljivost. Funkcija prostorskega skaliranja je bila uporabljena pri pristopu teleoperacije in je omogočala spremenljiv faktor prostorskega skaliranja med gibom operaterja in robota. To je posledično vplivalo na občutljivost robota na generirano gibanje operaterja.

Pri primerjavi smo poleg objektivnih meril (pogrešek pozicioniranja, hitrost in gladkost gibanja), upoštevali tudi nekatera subjektivna merila (enostavnost uporabe, občutek natančnosti in možnosti za izboljšave), ki smo jih določili na podlagi vprašalnika, ki so ga operaterji izpolnili ob koncu študije. To nam je omogočilo boljši vpogled in širšo razpravo pri primerjavi različnih pristopov.

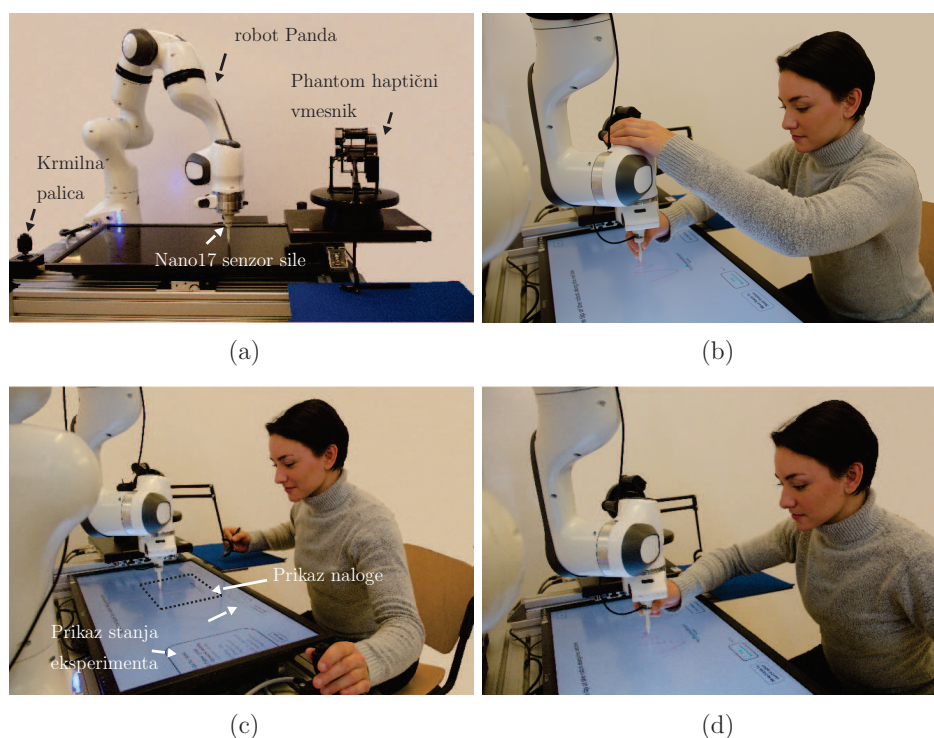
3.1 Materiali in metode

3.1.1 Postavitev študije

Za namen študije smo uporabili sistem prikazan na sliki 3.1, sestavljen iz sledečih komponent

- robotski manipulator Panda,
- haptični vmesnik Phantom Premium 1.5,
- senzor sile Nano17,
- krmilno palico z eno prostostno stopnjo,
- vodoravno postavljen LCD zaslon.

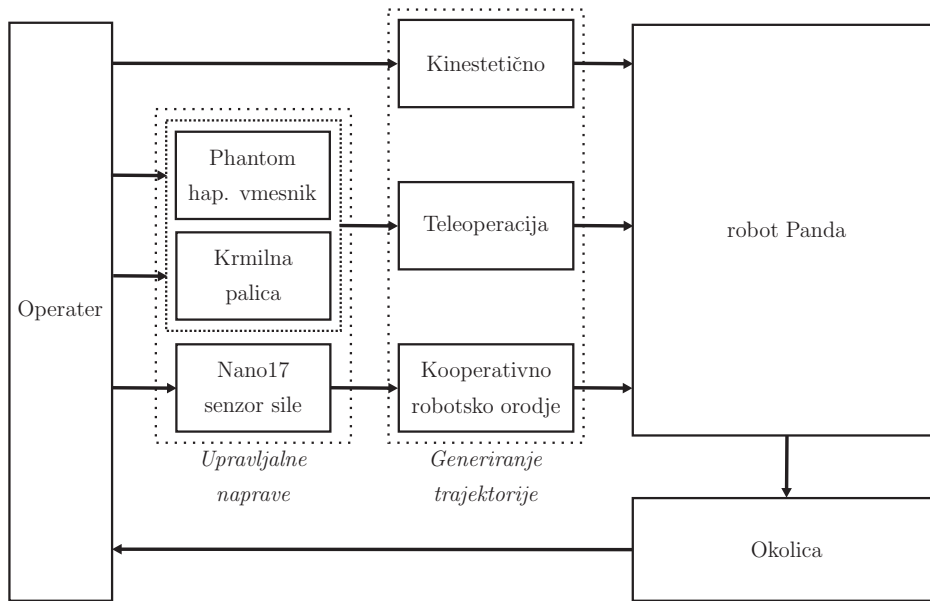
Robot Panda je bil uporabljen kot izvršna naprava, hkrati pa je pri kinestetičnem učenju služil tudi kot upravljalna naprava. Pri teleoperaciji in pristopu CRT sta kot upravljalni napravi služila vmesnik Phantom in senzor sile Nano17 (Slika 3.2). Pri teleoperaciji so uporabniki kot dodatno upravljalno napravo uporabljali tudi krmilno



Slika 3.1: Prikaz postavitve študije (a) in način uporabe naprav pri upravljanju z robotom Panda s pristopom kinestetičnega učenja (b), teleoperacije (c) in kooperativnega robotskega orodja (d).

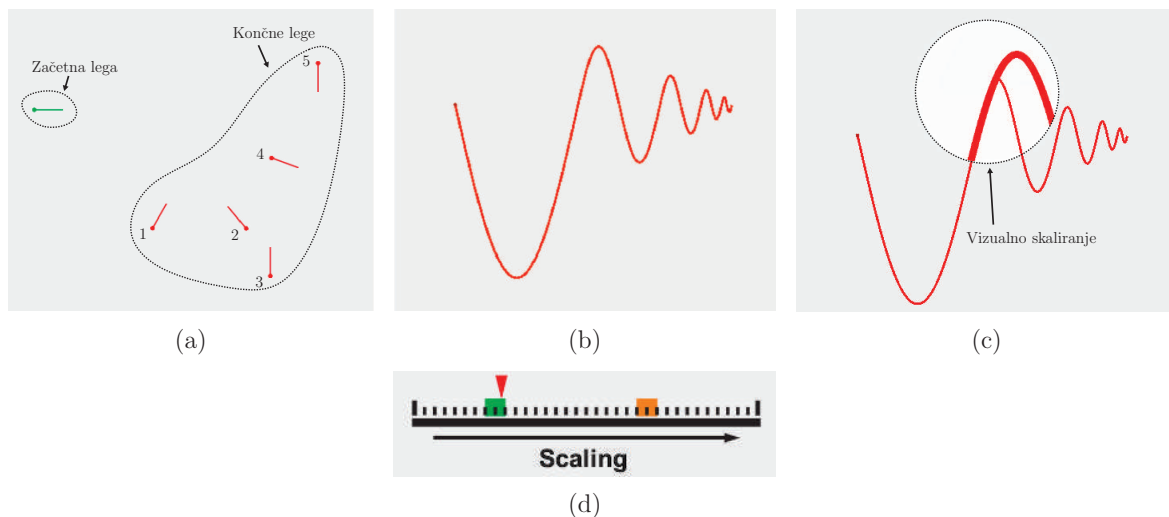
palico, s katero so nadzorovali vrednost prostorskega skaliranja med vhodnim gibom (Phantom) in generiranim gibom (Panda). Tako so operaterji v primeru kinestetičnega učenja z robotom opravljali dvoročno, pri čemer je ena roka služila generiranju večjih sil, druga pa natančnemu pozicioniranju (Slika 3.1, b). Dvoročno so operaterji upravljali tudi v primeru teleoperacije saj so z eno roko demonstrirali gib z vmesnikom Phantom, z drugo pa preko krmilne palice nadzirali vrednost prostorskega skaliranja med demonstriranim gibom in generiranim gibom robota Panda (Slika 3.1, c). V primeru pristopa CRT pa so operaterji z robotom upravljali enoročno saj je to zadostovalo za generiranje sil na senzor sile Nano17 (Slika 3.1, d).

Zasnovali smo prilagojeno orodje na vrhu robota z ozko konico, ki je operaterju omogočalo določanje lege vrha robota s sub-milimetrsko natančnostjo. V sklopu orodja je bil integriran tudi senzor sile Nano17, saj je to omogočalo optimalno izkušnjo pri pristopu CRT. Tekom študije je bila orientacija vrha robota konstantna, gibanje pa je bilo omejeno na vodoravno ravnino. To nam je omogočilo poenostavitev primerjave med posameznimi demonstracijami.



Slika 3.2: Shematski prikaz postavitve študije.

Za prikaz uporabniškega vmesnika je služil LCD zaslon, saj je glede na nalogo prikazoval lege začetnih in končnih točk ali referenčno pot (Slika 3.3). Omogočal pa je tudi implementacijo funkcije vizualnega skaliranja in prikazovanje indikatorja trenutne vrednosti prostorskega skaliranja. Funkcija vizualnega skaliranja je operaterju povečala območje pod vrhom robota s faktorjem 3. Druga pomembna funkcija uporabniškega

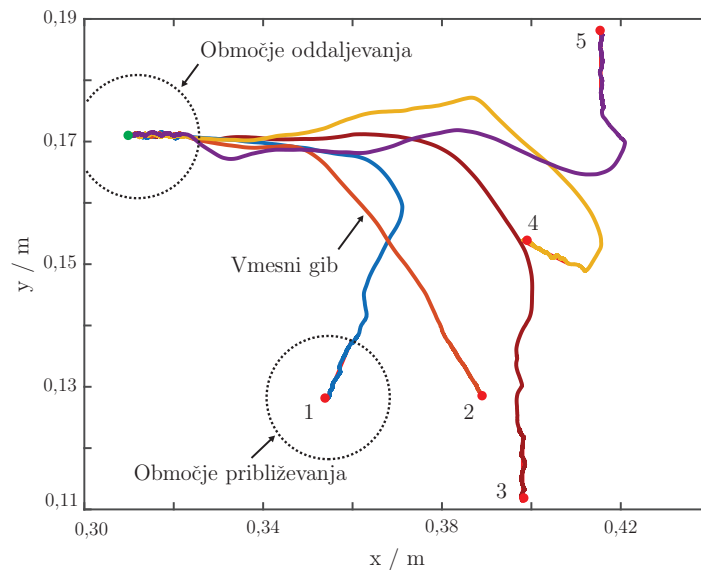


Slika 3.3: Podroben prikaz uporabniškega vmesnika - naloga premika od točke do točke (a) in sledenja poti (b), funkcija vizualnega skaliranja (c) in indikator trenutne vrednosti prostorskega skaliranja (d).

vmesnika je indikator trenutne vrednosti prostorskega skaliranja pri pristopu teleoperacije. Namen indikatorja je informiranje operaterja o trenutni vrednosti prostorskega skaliranja $K_{sp_{ii}}(t)$ (poglavje 2.2.3). Indikator vsebuje zeleno in oranžno območje. Zeleno območje je označevalo priporočeno območje vrednosti, ko je bila zahtevana večja natančnost ($K_{sp_{ii}}(t) = 0.25 \pm 0.05$), medtem ko je oranžno območje predstavljalo območje vrednosti, ko so bili potrebni večji premiki ($K_{sp_{ii}}(t) = 0.70 \pm 0.05$). Uporabniški vmesnik je bil zasnovan z uporabo igralnega pogona Unity (Unity Software, Inc.).

3.1.2 Naloge

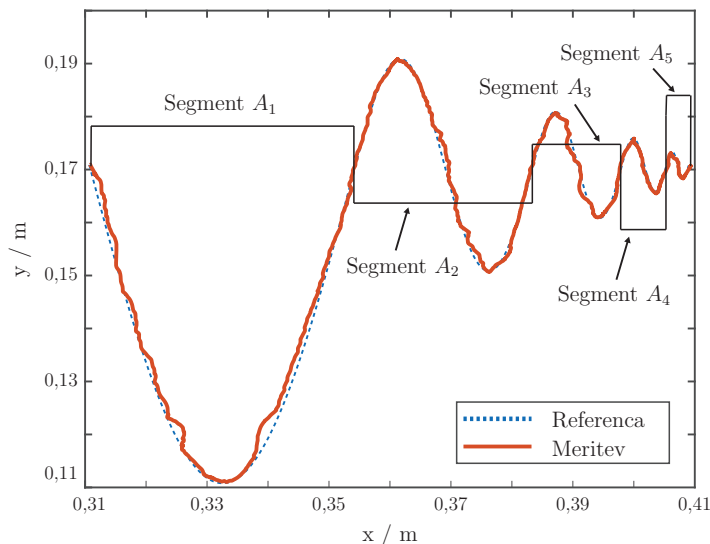
V sklopu študije smo pripravili dve različni nalogi za primerjavo pristopov demonstriranja finih gibov. Po zgledu Akgun in drugih smo zasnovali nalogo giba od točke do točke in nalogo sledenja poti [1].



Slika 3.4: Prikaz izvedenih demonstracij operaterja v posamezne končne točke (1 - 5) v sklopu naloge giba od točke do točke.

Naloga giba od točke do točke je v praksi uporabna, ko je potrebno nastaviti začetne, končne in vmesne lege, pri čemer je ustvarjena pot med njimi nepomembna. V nekaterih primerih proces zahteva natančno definicijo lege. Zato smo zasnovali nalogo, pri kateri je moral operater prestaviti vrh robota iz začetne v ciljno lego, ki je bila prikazana kot krog s premerom 1 mm (Slika 3.3, a). Iz kroga je potekala tudi črta dolga 10 mm, ki je prikazovala zeleno smer, iz katere naj se operater približuje cilju. To nam je

omogočalo lažjo primerjavo med operaterji, saj smo poleg tega, da smo določili, ali je operater dosegel želeno lego, lahko tudi analizirali način približevanja cilju. Tekom študije smo prikazali pet različnih ciljnih točk, pri čemer je imela vsaka drugačno smer črte približevanja. To nam je posledično omogočilo bolj posplošeno analizo. Primeri izvedenih demonstracij so prikazani na sliki 3.4.



Slika 3.5: Prikaz izvedene demonstracije operaterja v sklopu naloge gibanja vzdolž poti.

Naloga sledenja poti je v praksi smiselna, ko je potrebno izvesti specifičen gib. V tej nalogi smo preverili vpliv prostorske razsežnosti poti na natančnost izvedene demonstracije. Zasnovali smo referenčno pot $f(x,y)$ s postopno manjšimi prostorskimi razsežnostmi (Slika 3.3, b), vzdolž katere je operater moral čim bolj natančno premikati vrh robota. Sestavljena je iz večjih segmentov sinusnih krivulj $f(x_i,y_i)$, katerih amplituda se postopoma zmanjšuje

$$y_i(t) = A_i \sin(\omega_i t) + y_0, \quad \mathbf{A} = \begin{bmatrix} 0.06 \\ 0.02 \\ 0.01 \\ 0.005 \\ 0.0025 \end{bmatrix} m, \quad (3.1)$$

pri čemer $y_i(t)$ predstavlja časovni potek i -tega segmenta poti vzdolž osi y koordinatnega sistema, y_0 začetno vrednost krivulje, A_i pa amplitudo sinusne krivulje segmenta. Vrednost spremenljivke $x_i(t)$, ki predstavlja časovni potek i -tega segmenta poti vzdolž

osi x koordinatnega sistema, pa smo nadalje določili kot

$$x_i(t) = K(t - t_{s_i}) + x_{s_i}, \quad (3.2)$$

pri čemer K predstavlja skalirni faktor med časovno in prostorsko dimenzijo, x_{s_i} začetno x vrednost, t_{s_i} pa začetni čas posameznega segmenta. Za zagotovitev zveznosti celotne krivulje smo nadalje definirali ustrezno ω_i

$$\omega_i = \frac{A_{i-1}}{A_i} \omega_{i-1}, \quad \omega_0 = 0.01256. \quad (3.3)$$

Postopno zmanjševanje amplitude segmentov je omogočilo gladek prehod od grobih do finih gibov. Referenčna pot je predstavljala fin gib v zadnjih dveh segmentih (A_4 , A_5). Posamezne dele poti (Slika 3.5) bomo naslavljali s segmenti in označevali z A_i (tj. segment A_3 predstavlja del poti, kjer je $A_i = A_3$). Kljub temu da je bila krivulja generirana kot funkcija časa t , pa je bil časovni potek relevanten zgolj pri generiranju krivulje. Tekom izvajanja demonstracije, je bil časovni potek generirane poti irelevanten.

3.1.3 Protokol študije

Celotni protokol študije je bil pri vsakemu udeležencu sestavljen iz uvajanja, meritev in vprašalnika. V študiji je sodelovalo 31 udeležencev:

- 20 moških in 11 žensk,
- 11 strokovnjakov iz področja robotike in 20 nestrokovnjakov,
- povprečna starost posameznika je znašala $32 \pm 9,8$ let.

Vsak udeleženec je začel s približno 20 minut dolgim uvajanjem, med katerim je

- preizkusil različne pristope izvajanja demonstracij,
- se seznanil s funkcijo prostorskega in vizualnega skaliranja.

Preliminarne študije so pokazale, da se je čas usposabljanja, potrebnega za zmanjšanje učinka učenja tekom študije, pri posameznih udeležencih razlikoval glede na njihove motorične sposobnosti in predhodne izkušnje. Tako smo vsakemu udeležencu dodelili dovolj časa, da se je pri nadzoru robota počutil sposobnega. Niz kratkih testov nam je nakazal, ali so udeleženci še v postopku učenja ali so njihove sposobnosti upravljanja že

začele stagnirati in so pripravljene na začetek meritev. Tako smo uspešno minimizirali vplive kot so perspektiva pogleda na delovno okolje, nepoznavanje vmesnih naprav ter funkcije vizualnega skaliranja. Meritve so se začele z nalogo giba od točke do točke, pri kateri je vsak udeleženec:

- za vsakega od treh načinov demonstriranja (kinestetično, teleoperacija in CRT) premaknil vrh robota iz začetne točke v vsako od petih ciljnih točk,
- štirikrat brez vizualnega skaliranja in štirikrat z vizualnim skaliranjem.

Temu je sledila naloga sledenja poti, pri kateri je vsak udeleženec:

- premaknil vrh robota vzdolž referenčne poti za vsakega od pristopov demonstriranja,
- štirikrat brez vizualnega skaliranja in štirikrat z vizualnim skaliranjem.

Pri teleoperaciji smo želeli preveriti vpliv prostorskega skaliranja, zato smo protokol v drugi alineji prilagodili. Operaterji so še vedno izvedli štiri ponovitve pod vsakim od pogojev vizualnega skaliranja, vendar so tri od teh ponovitev izvedli s spremenljivim faktorjem prostorskega skaliranja, eno pa s konstantnim faktorjem prostorskega skaliranja ($K_{spi}(t) = 1$).

Na koncu je moral operater izpolniti vprašalnik za vsak pristop izvajanja demonstracij in odgovoriti na dve vprašanji odprtega tipa. Vsak udeleženec je za celoten eksperiment potreboval 2-3 ure.

3.1.4 Analiza podatkov

Pri nalogi giba od začetne do končne točke je bil cilj analizirati pogrešek pozicioniranja pri odmiku iz začetne točke in pri približevanju končni točki. V sklopu študije smo za analizo pridobili 3720 meritev.

Pogrešek pozicioniranja smo spremljali, ko je bil vrh robota znotraj radija 10 mm od začetne ali končne točke (Slika 3.4), določili pa smo ga kot najmanjšo razdaljo med vrhom robota in črto približevanja/oddaljevanja. Analizirali smo tudi hitrost in gladkost gibanja. Hitrost je bila določena na podlagi dolžine opravljene poti giba in pretečenega časa, medtem ko je bila gladkost giba določena z uporabo uveljavljene metode SAL (angl. spectral arc length) [46].

Gladkost temelji na analizi Fourierjevega spektra hitrosti gibanja vrha robota $\dot{x}(t)$. Določena je namreč z dolžino loka frekvenčnega spektra $\dot{X}(\omega)$ kot

$$\eta_{sal} = - \int_0^{\omega_b} \sqrt{\left(\frac{1}{\omega_b}\right)^2 + \left(\frac{d\hat{X}(\omega)}{d\omega}\right)^2} d\omega, \quad \hat{X}(\omega) = \frac{\dot{X}(\omega)}{\dot{X}(0)}, \quad (3.4)$$

pri čemer $\omega \in [0, \omega_b]$ predstavlja frekvenčni pas, ki je relevanten za analizo. Vrednost η_{sal} se tako z večanjem gladkosti zmanjšuje proti vrednosti 0. Hitrost in gladkost gibanja sta bili analizirani ločeno za vmesno gibanje (grobo gibanje med začetno in končno točko) in gibanje v bližini točk.

Pri nalogi sledenja poti je bil cilj določiti natančnost, s katero lahko operater sledi referenčni poti. V sklopu študije smo pridobili 744 meritev. Za ustrezno primerjavo izvedenih demonstracij z referenčno potjo je bilo potrebno zajete meritve obdelati za namen časovne poravnave. Uporabili smo uveljavljeno metodo DTW (angl. dynamic time warp) [47]. Metoda določa optimalno ujemanje med dvema setoma podatkov, pri čemer nelinearno preoblikuje njuno časovno dimenzijo. Časovna poravnava nam je omogočila primerjavo izvedenih demonstracij z referenčno potjo in izračun pogreška pozicioniranja, ki je bil določen kot evklidska razdalja med vrhom robota in referenčnim položajem. Pogreški so bili izračunani vzdolž celotne poti in združeni pri analizi posameznega segmenta (Slika 3.5). Opravljena je bila tudi analiza hitrosti gibanja in gladkosti in sicer ločeno za vsak segment poti. Hitrost je bila odvisna od dolžine poti in časa, potrebnega za izvedbo giba, medtem ko je bila gladkost ponovno določena z uporabo metode SAL.

3.1.5 Statistična analiza

Da bi ugotovili, ali obstaja statistično signifikantna razlika med različnimi pristopi izvajanja demonstracije in pri različnih pogojih, je bilo potrebno izbrati ustrezne statistične metode. Ker je ista skupina udeležencev izvedla demonstracije pod različnimi pogoji, je ustrezen parni statistični test. Zaradi pogoste kršitve sferičnosti in pojava osamelcev v naboru podatkov, smo se odločili, da je potrebno uporabiti neparametrične teste [48–50]. Za primerjavo dveh pogojev je bil uporabljen dvostranski Wilcoxonov test, Friedmanov test pa smo izvedli, ko smo hkrati primerjali več kot dva pogoja [51,52]. V kolikor je bila ničta hipoteza Friedmanovega testa zavrnjena, je bil za medsebojno primerjavo uporabljen dvostranski Wilcoxonov test. Prag signifikance α je bil določen na 0,05. Za omejitev inflacije napake tipa 1 zaradi medsebojnih primerjav, smo uporabili

Bonferronijev popravek [53].

3.1.6 Vprašalnik

Vprašalnik je bil zasnovan z namenom pridobitve dodatnega vpogleda v vsakega izmed pristopov izvajanja demonstracij ter funkcije vizualnega in prostorskega skaliranja. Udeleženci so morali za vsako nalogo oceniti enostavnost uporabe, doseženo natančnost in možnost izboljšave z uporabo sedemstopenjske Likertove lestvice.

- Enostavnost uporabe je predstavljala težavnost, s katero operater pozicionira vrh robota, pri čemer 1 predstavlja “zelo težavno”, 7 pa “najmanj težavno”.
- Dosežena natančnost je predstavljala operaterjevo mnenje o doseženi natančnosti, pri čemer je 1 predstavljal “najnižjo natančnost”, 7 pa “najvišjo natančnost”.
- Možnost izboljšave pa je predstavljalo operaterjevo mnenje o možnih nadaljnjih izboljšavah pri posameznem pristopu, v kolikor bi imel na voljo več časa za usposabljanje. Tu je 1 predstavljal “brez izboljšav”, 7 pa “veliko izboljšav”.

Poleg tega smo udeležencem postavili tudi dve odprti vprašanji:

- “Za katerega od pristopov bi se odločil, v kolikor bi moral izvesti fin gib in zakaj?”
- “Ali se vam je zdelo vizualno skaliranje koristno in zakaj?”

Odgovori na ti dve vprašanji so nam dali dodaten vpogled v vsakega od pristopov in v funkcijo vizualnega skaliranja.

3.2 Rezultati

V poglavju predstavljamo rezultate študije. Rezultati so predstavljeni s škatličnimi diagrami in opisani z uporabo mediane (Med), petindvajsetega percentila (Q1), petinsemdesetega percentila (Q3), najmanjše vrednosti (Min) in največje vrednosti (Maks). Vse primerjave smo statistično ovrednotili ter rezultate prikazali v poglavju Dodatek B.

Med analizo podatkov smo ugotovili, da se rezultati naloge gibanja od točke do točke niso bistveno razlikovali glede na različne lege končnih točk. Zato smo rezultate meritev pri enakih eksperimentalnih pogojih združili in se osredotočili na raz-

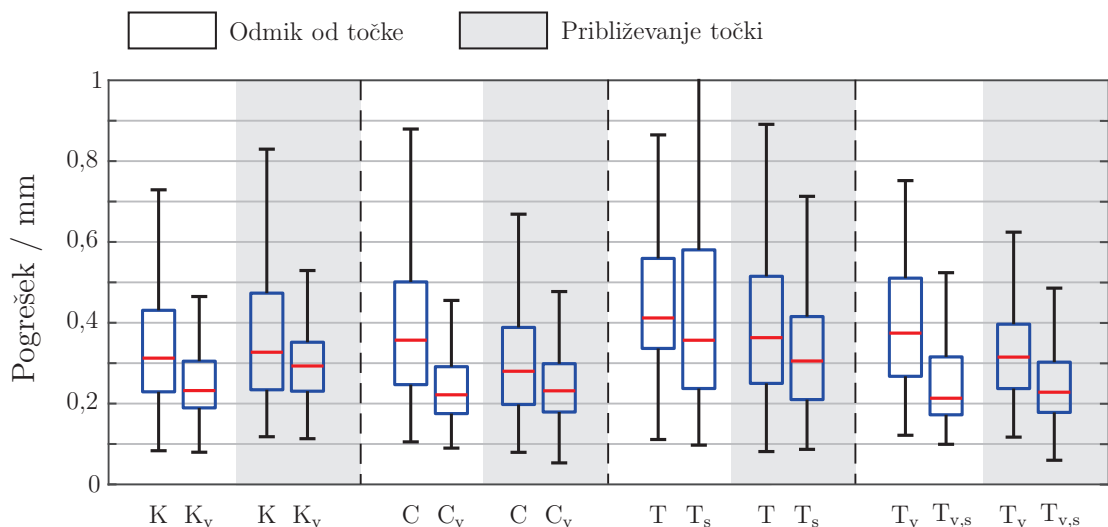
like v uspešnosti z različnimi pristopi demonstracije. Pri obeh nalogah prav tako nismo ugotovili signifikantnih razlik glede na spol (moški/ženske) ali strokovno znanje (strokovnjak/nepoznavalec). Zadosten in prilagodljiv postopek uvajanja in dejstvo, da je študija od udeležencev zahtevala dobre motorične sposobnosti namesto obsežnih izkušenj z roboti, bi lahko bila dva razloga za to, da nismo ugotovili signifikantnih razlik glede na strokovno znanje.

Prav tako smo pri nalogi sledenja poti ugotovili, da rezultati iz segmentov A_1 in A_2 niso odstopali od rezultatov, ki smo jih opazili v segmentih A_3 , A_4 in A_5 . Rezultate prvih dveh segmentov smo tako izpustili iz študije in se osredotočili na segmente, ki so zahtevali generiranje gibanja manjših prostorskih dimenzij.

3.2.1 Naloga 1: premik od točke do točke

Na sliki 3.6 smo s pomočjo škatličnih diagramov predstavili odstopanja pri pozicioniranju, ki so jih operaterji naredili med odmikom iz začetne točke (belo ozadje) in pri približevanju končni točki (sivo ozadje). Slika prikazuje posamezne pristope z in brez vpliva vizualnega ter prostorskega skaliranja.

K - kinestetično (brez vizualnega skaliranja)	T - teleoperacija (brez viz. skal., brez prost. skal.)
K_v - kinestetično (z vizualnim skaliranjem)	T_s - teleoperacija (brez viz. skal., s prost. skal.)
C - CRT (brez vizualnega skaliranja)	T_v - teleoperacija (z viz. skal., brez prost. skal.)
C_v - CRT (z vizualnim skaliranjem)	$T_{v,s}$ - teleoperacija (z viz. skal., s prost. skal.)



Slika 3.6: Prikaz pogreškov pozicioniranja pri odmiku od začetne točke in med približevanjem končni točki.

Pogrešek pozicioniranja - brez vizualnega skaliranja

Ko funkcija vizualnega skaliranja ni bila uporabljena med odmikanjem iz začetnega položaja, se je kinestetično učenje (Med = 0,33 mm; Q1 = 0,23 mm; Q3 = 0,47 mm; Min = 0,12 mm; in Maks = 1,43 mm) obneslo podobno kot CRT in teleoperacija s funkcijo prostorskega skaliranja brez statistično signifikantnih razlik. Teleoperacija brez funkcije prostorskega skaliranja se je obnesla slabše kot kinestetično učenje in CRT, ni pa bilo statistično signifikantne razlike med obema pristopoma teleoperacije. Med približevanjem ciljni točki se je CRT izkazal za najboljšega (Med = 0,28 mm; Q1 = 0,19 mm; Q3 = 0,39 mm; Min = 0,07 mm; in Maks = 0,87 mm), sledili pa so kinestetično učenje, teleoperacija z uporabo prostorskega skaliranja in teleoperacija brez uporabe prostorskega skaliranja.

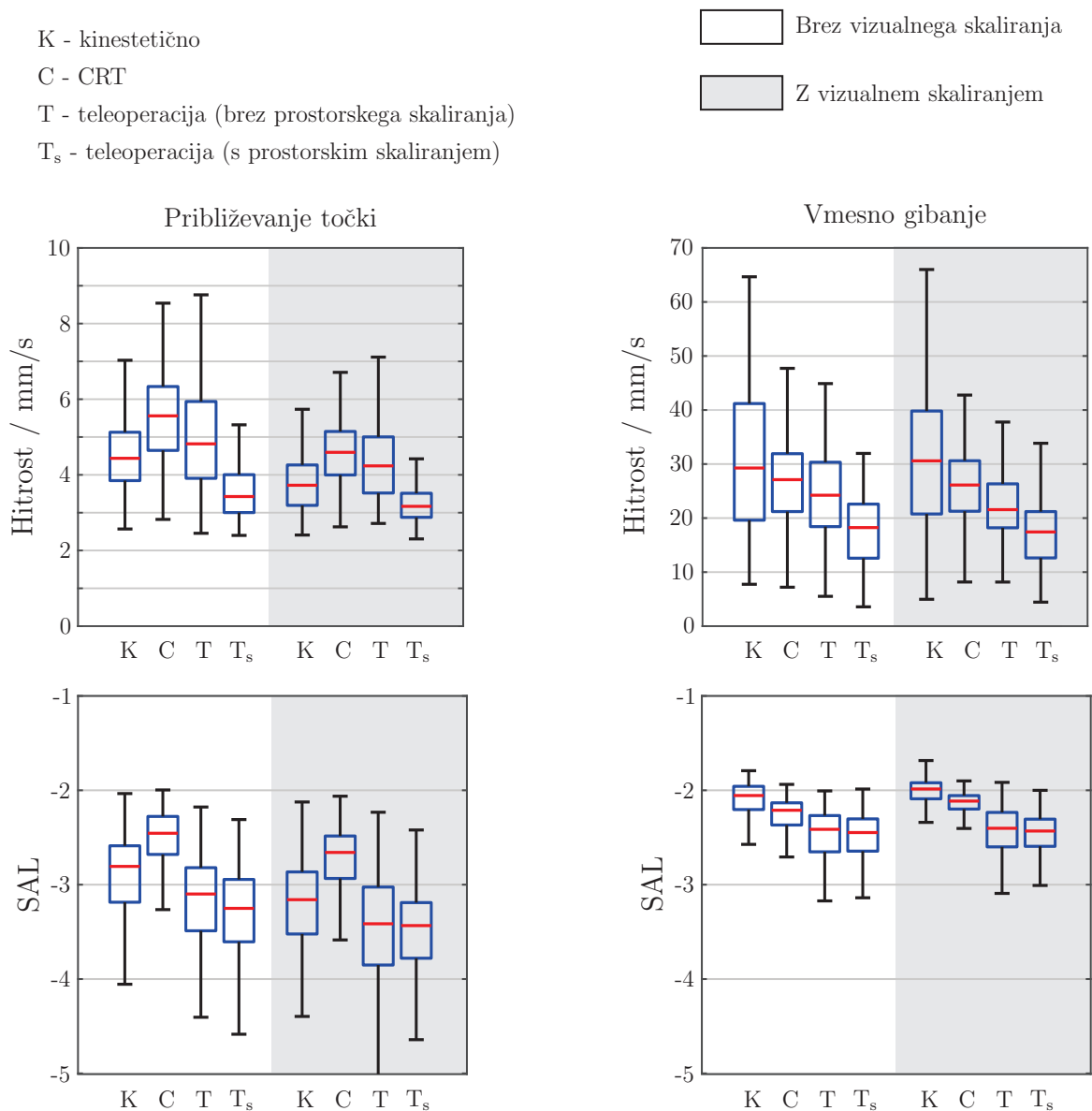
Pogrešek pozicioniranja - z vizualnim skaliranjem

Ko je bila uporabljena funkcija vizualnega skaliranja, sta se pri odmiku iz začetne točke za najboljšega izkazala CRT (Med = 0,22 mm; Q1 = 0,17 mm; Q3 = 0,29 mm; Min = 0,09 mm in Maks = 0,79 mm) in teleoperacija s prostorskim skaliranjem brez statistično signifikantne razlike v rezultatih. Temu sta sledila kinestetično učenje in teleoperacija brez funkcije prostorskega skaliranja. Pri približevanju cilju so se kinestetično učenje (Med = 0,23 mm; Q1 = 0,19 mm; Q3 = 0,31 mm; Min = 0,08 mm in Maks = 0,84 mm), CRT in teleoperacija s prostorskim skaliranjem obnesli podobno dobro, brez statistično signifikantnih razlik, medtem ko je bil pri teleoperaciji brez prostorskega skaliranja generiran večji pogrešek pozicioniranja.

Na sliki 3.7 smo za prikaz hitrosti gibanja in gladkosti giba pri posameznem pristopu demonstracije uporabili škatlični diagram. Rezultati so prikazani ločeno za vmesno gibanje in gibanje med približevanjem ciljni točki. Prikazan je tudi vpliv funkcije vizualnega in prostorskega skaliranja.

Hitrost in gladkost gibanja - brez vizualnega skaliranja

Ko med gibanjem proti ciljni točki ni bila uporabljena funkcija vizualnega skaliranja, se je za najhitrejši pristop izkazal CRT (Med = 5,6 mm/s; Q1 = 4,7 mm/s; Q3 = 6,3 mm/s; Min = 2,8 mm/s; in Maks = 9,7 mm/s), ki so mu sledili teleoperacija brez prostorskega skaliranja, kinestetično učenje in teleoperacija s prostorskim skaliranjem. Glede gladkosti gibanja je CRT (Med = -2,46; Q1 = -2,69; Q3 = -2,28; Min = -4,49; in Maks = -2,00) ustvaril tudi najbolj gladko gibanje, sledili pa so kinestetično učenje, teleoperacija brez prostorskega skaliranja in teleoperacija s prostor-



Slika 3.7: Hitrost in gladkost demonstriranih gibov med vmesnim premikom in pri približevanju končni točki.

skim skaliranjem. Med vmesnim gibanjem se je kinestetično učenje (Med = 29,3 mm/s; Q1 = 19,6 mm/s; Q3 = 41,2 mm/s; Min = 7,8 mm/s; Maks = 64,7 mm/s) izkazalo za najhitrejši pristop, sledila pa sta CRT in teleoperacija brez prostorskega skaliranja, pri čemer med njima ni bilo statistično signifikantne razlike. Pri teleoperaciji s prostorskim skaliranjem je bilo gibanje najpočasnejše. Gladkost giba je bila najboljša pri kinestetičnem učenju (Med = -2,05; Q1 = -2,20; Q3 = -1,96; Min = -3,43; Maks = -1,79), sledili pa so CRT, teleoperacija brez prostorskega skaliranja in teleoperacija s

prostorskim skaliranjem. Med obema pristopoma teleoperacije ni bilo statistično signifikantnih razlik.

Hitrost in gladkost gibanja - z vizualnim skaliranjem

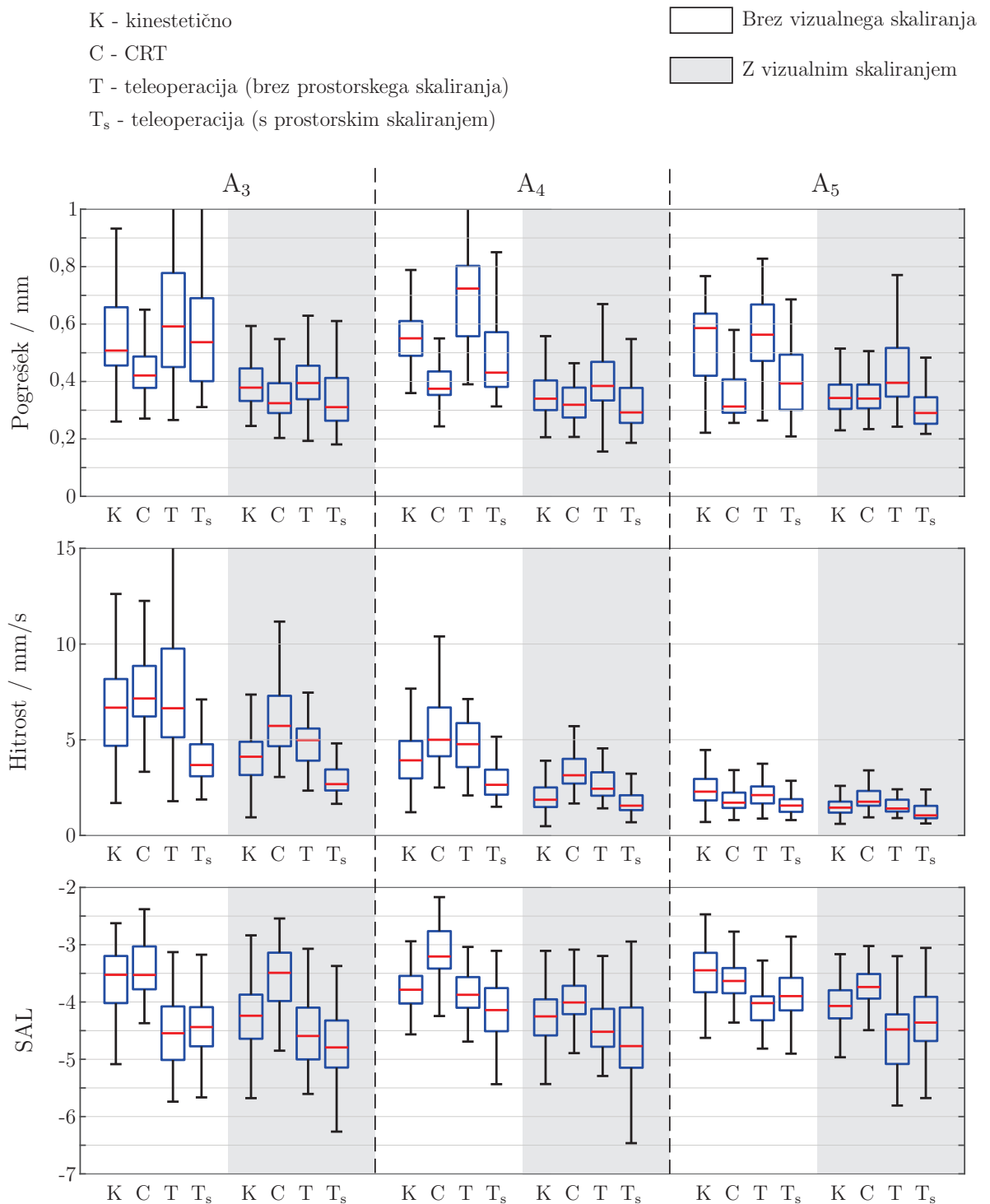
Pri uporabi funkcije vizualnega skaliranja se je pri približevanju ciljni točki za najhitrejši pristop izkazal CRT (Med = 4,6 mm/s; Q1 = 4,0 mm/s; Q3 = 5,1 mm/s; Min = 2,6 mm/s; in Maks = 7,7 mm/s), sledili pa so teleoperacija brez prostorskega skaliranja, kinestetično učenje in teleoperacija s prostorskim skaliranjem. Glede gladkosti gibanja je CRT (Med = -2,66; Q1 = -2,93; Q3 = -2,48; Min = -4,07; Maks = -2,06) generiral najbolj gladek gib, sledili pa so kinestetično učenje in teleoperacija brez prostorskega skaliranja in teleoperacija s prostorskim skaliranjem, pri čemer med slednjima dvema ni statistično signifikantne razlike. Med vmesnim gibom pa se je kinestetično učenje (Med = 30,6 mm/s; Q1 = 20,7 mm/s; Q3 = 39,8 mm/s; Min = 4,9 mm/s; in Maks = 70,2 mm/s) izkazalo za najhitrejšega, sledili pa so CRT, teleoperacija brez prostorskega skaliranja in teleoperacija s prostorskim skaliranjem. Kar zadeva gladkosti gibanja je kinestetično učenje (Med = -1,98; Q1 = -2,09; Q3 = -1,92; Min = 2,78; in Maks = 1,68) generiralo najbolj gladko gibanje, sledijo pa CRT, teleoperacija brez prostorskega skaliranja in teleoperacija s prostorskim skaliranjem, kjer med slednjima ni bilo statistično signifikantne razlike.

3.2.2 Naloga 2: sledenje poti

Na sliki 3.8 smo s pomočjo škatličnih diagramov predstavili pogoške pozicioniranja, hitrost in gladkost gibanja za vsakega izmed pristopov demonstracije. Rezultati so prikazani ločeno za vsakega od zadnjih treh segmentov poti (A_3 , A_4 in A_5). Poleg tega je prikazan tudi vpliv funkcije vizualnega in prostorskega skaliranja.

Segment A_3 - brez vizualnega skaliranja

Ko funkcija vizualnega skaliranja ni bila uporabljena, je pristop CRT (Med = 0,42 mm; Q1 = 0,38 mm; Q3 = 0,49 mm; Min = 0,27 mm; in Maks = 0,75 mm) ustvaril najmanjši pogrešek pozicioniranja v segmentu A_3 . Sledili so mu kinestetično učenje, teleoperacija s prostorskim skaliranjem in teleoperacija brez prostorskega skaliranja, vendar med kinestetičnim učenjem in teleoperacijo s prostorskim skaliranjem ni bilo statistično signifikantne razlike. Prav tako ni bilo statistično signifikantne razlike med obema pristopoma teleoperacije. Hitrost gibanja je bila največja pri CRT (Med = 7,2 mm/s; Q1 = 6,2 mm/s; Q3 = 8,9 mm/s; Min = 3,3 mm/s; Maks = 17,8 mm/s), sledili pa



Slika 3.8: Pregled pogoškov pozicioniranja, hitrosti in gladkosti gibanja v vzdolž posameznih segmentov poti.

so teleoperacija brez prostorskega skaliranja, kinestetično učenje in teleoperacija s prostorskim skaliranjem. Med teleoperacijo brez prostorskega skaliranja in kinestetičnim

učenjem ter CRT ni bilo statistično signifikantne razlike. Glede gladkosti gibanja je CRT (Med = -3,53; Q1 = -3,78; Q3 = -3,03; Min = -4,37; Maks = -2,38) generiral najbolj gladko gibanje, sledili pa so kinestetično učenje, teleoperacija brez prostorskega skaliranja in teleoperacija s prostorskim skaliranjem, pri čemer med slednjima ni bilo statistično signifikantnih razlik.

Segment A_4 - brez vizualnega skaliranja

V segmentu A_4 je CRT (Med = 0,37 mm; Q1 = 0,35 mm; Q3 = 0,44 mm; Min = 0,24 mm; Maks = 0,75 mm) ustvaril najmanjši pogrešek pozicioniranja, sledijo teleoperacija s prostorskim skaliranjem, kinestetično učenje in teleoperacija brez prostorskega skaliranja. Generirana hitrost gibanja je bila največja pri CRT (Med = 5,0 mm/s; Q1 = 4,1 mm/s; Q3 = 6,7 mm/s; Min = 2,5 mm/s; in Maks = 12,7 mm/s), sledila pa sta kinestetično učenje in teleoperacija brez prostorskega skaliranja, med katerima ni bilo statistično signifikantne razlike. Gibanje je bilo najpočasnejše pri uporabi teleoperacije s funkcijo prostorskega skaliranja. Najbolj gladko gibanje je bilo generirano s pristopom CRT (Med = -3,21; Q1 = -3,42; Q3 = -2,76; Min = -4,24; in Maks = -2,17), sledita kinestetično učenje in teleoperacija brez prostorskega skaliranja, med katerima ni bilo statistično signifikantne razlike. Teleoperacija s prostorskim skaliranjem je generirala najbolj grobo gibanje.

Segment A_5 - brez vizualnega skaliranja

V segmentu A_5 je CRT (Med = 0,31 mm; Q1 = 0,29 mm; Q3 = 0,41 mm; Min = 0,26 mm; in Maks = 0,79 mm) dosegel najmanjši pogrešek pozicioniranja, sledili pa so teleoperacija s prostorskim skaliranjem, kinestetično učenje in teleoperacija brez prostorskega skaliranja. Ustvarjena hitrost gibanja je bila največja pri kinestetičnem učenju (Med = 2,3 mm/s; Q1 = 1,8 mm/s; Q3 = 2,9 mm/s; Min = 0,7 mm/s; in Maks = 6,2 mm/s) sledili pa so teleoperacija brez prostorskega skaliranja, CRT in teleoperacija s prostorskim skaliranjem, vendar med CRT in teleoperacijo s prostorskim skaliranjem ni bilo statistično signifikantne razlike. Kinestetično učenje (Med = -3,45; Q1 = -3,83; Q3 = -3,14; Min = -5,48; in Maks = -2,47) je prav tako generiralo najbolj gladko gibanje, sledili pa so CRT, teleoperacija s prostorskim skaliranjem in teleoperacija brez prostorskega skaliranja. Med obema pristopoma teleoperacije ni bilo statistično signifikantne razlike.

Segment A_3 - z vizualnim skaliranjem

Ko je bila uporabljena funkcija vizualnega skaliranja, sta v segmentu A_3 CRT

(Med = 0,32 mm; Q1 = 0,29 mm; Q3 = 0,39 mm; Min = 0,20 mm in Maks = 0,59 mm) in teleoperacija s prostorskim skaliranjem (Med = 0,31 mm; Q1 = 0,26 mm; Q3 = 0,41 mm; Min = 0,18 mm; in Maks = 0,67 mm) dosegla manjši pogrešek pozicioniranja kot kinestetično učenje in teleoperacija brez prostorskega skaliranja. Med CRT in teleoperacijo s prostorskim skaliranjem ni bilo statistično signifikantne razlike. Generirana hitrost gibanja je bila največja pri CRT (Med = 5,7 mm/s; Q1 = 4,7 mm/s; Q3 = 7,3 mm/s; Min = 3,1 mm/s; in Maks = 15,3 mm/s) in teleoperaciji brez prostorskega skaliranja, sledili pa sta kinestetično učenje in teleoperacija s prostorskim skaliranjem. Med CRT in teleoperacijo brez prostorskega skaliranja ni bilo statistično signifikantne razlike. Kar zadeva gladkosti gibanja je CRT ustvaril najbolj gladko gibanje (Med = 3,49; Q1 = 3,98; Q3 = 3,14; Min = -5,42; in Maks = -2,54) sledili pa so kinestetično učenje, teleoperacija brez prostorskega skaliranja in teleoperacija s prostorskim skaliranjem. Vendar med kinestetičnim učenjem in teleoperacijo brez prostorskega skaliranja ni bilo statistično signifikantne razlike. Prav tako ni bilo razlike med obema pristopoma teleoperacije.

Segment A_4 - z vizualnim skaliranjem

V segmentu A_4 sta teleoperacija s prostorskim skaliranjem (Med = 0,29 mm; Q1 = 0,26 mm; Q3 = 0,38 mm; Min = 0,19 mm in Maks = 0,58 mm) in CRT dosegla najmanjši pogrešek pozicioniranja, sledila sta kinestetično učenje in teleoperacija brez prostorskega skaliranja. Med CRT in teleoperacijo s prostorskim skaliranjem ni statistično signifikantne razlike. Največjo hitrost gibanja je dosegel CRT (Med = 3,1 mm/s; Q1 = 2,7 mm/s; Q3 = 4,0 mm/s; Min = 1,7 mm/s; in Maks = 7,8 mm/s), sledili so teleoperacija brez prostorskega skaliranja, kinestetično učenje in teleoperacija s prostorskim skaliranjem. Glede gladkosti gibanja je CRT (Med = -4,01; Q1 = -4,21; Q3 = -3,72; Min = -4,89; in Maks = -3,09) ustvaril najbolj gladko gibanje, sledili so kinestetično učenje, teleoperacija brez prostorskega skaliranja in teleoperacija s prostorskim skaliranjem, vendar med kinestetičnim učenjem in teleoperacijo brez prostorskega skaliranja ni bilo statistično signifikantne razlike. Prav tako ni bilo statistično signifikantne razlike med obema pristopoma teleoperacije.

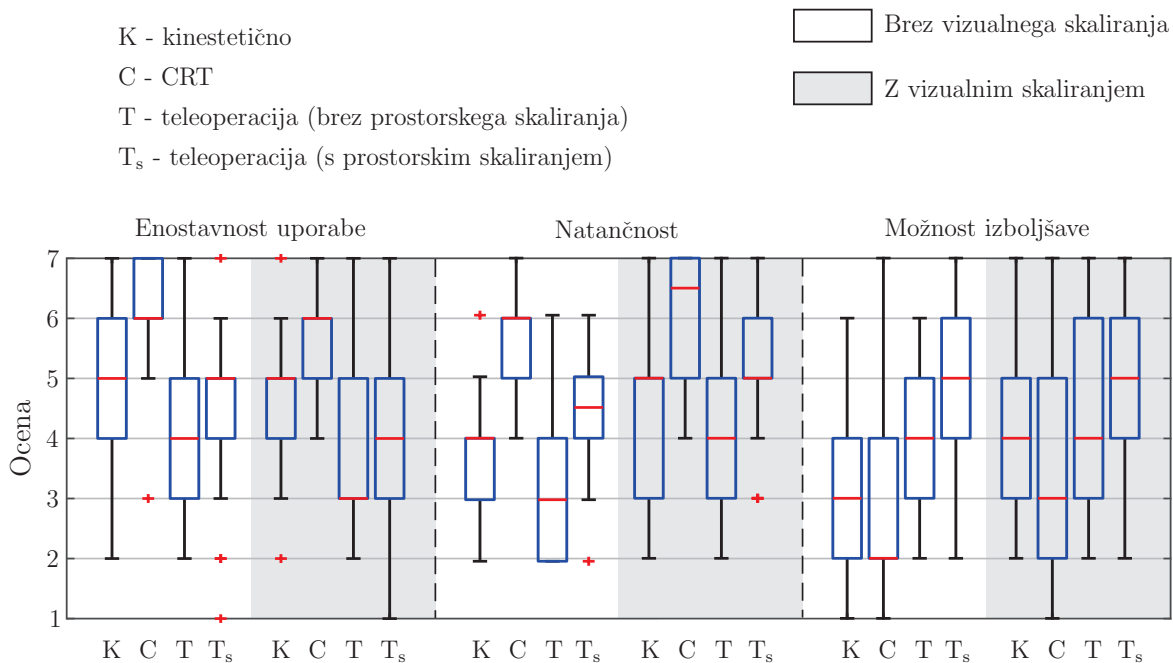
Segment A_5 - z vizualnim skaliranjem

V segmentu A_5 je teleoperacija s prostorskim skaliranjem (Med = 0,29 mm; Q1 = 0,25 mm; Q3 = 0,35 mm; Min = 0,22 mm; in Maks = 0,62 mm) dosegla najmanjši pogrešek pozicioniranja, sledili pa so CRT, kinestetično učenje in teleoperacija

brez prostorskega skaliranja. Med CRT in kinestetičnim učenjem ni bilo statistično signifikantne razlike. Največjo hitrost gibanja je dosegel CRT (Med = 1,8 mm/s; Q1 = 1,6 mm/s; Q3 = 2,3 mm/s; Min = 0,9 mm/s; in Maks = 4,0 mm/s), ki so mu sledili kinestetično učenje, teleoperacija brez prostorskega skaliranja in teleoperacija s prostorskim skaliranjem. Vendar med kinestetičnim učenjem in teleoperacijo brez prostorskega skaliranja. Prav tako ni bilo statistično signifikantne razlike med obema teleoperacijama. Kar zadeva gladkosti je CRT (Med = -3,74; Q1 = -3,94; Q3 = -3,51; Min = -4,74; in Maks = -2,72) generiral najbolj gladko gibanje, sledilo je kinestetično učenje in temu teleoperacija s prostorskim skaliranjem in teleoperacija brez prostorskega skaliranja.

3.2.3 Vprašalnik

Na sliki 3.9 smo za predstavitev rezultatov vprašalnika uporabili škatlične diagrame. Udeleženci so ocenili pristope demonstracij (z in brez vizualnega in prostorskega skaliranja) na podlagi enostavnosti uporabe, dosežene znanosti in možnosti izboljšav z uporabo sedemstopenjske Likertove lestvice.



Slika 3.9: Pregled subjektivnih ocen udeležencev o pristopih in vplivu funkcij vizualnega in prostorskega skaliranja. Rdeči + znaki predstavljajo podatkovne osamelce.

Ko vizualno skaliranje ni bilo uporabljeno, so operaterji najlažje uporabljali CRT,

sledilo je kinestetično učenje, teleoperacija s prostorskim skaliranjem in teleoperacija brez prostorskega skaliranja. Predvidevali so, da so bili najnatančnejši s pristopom CRT, sledi teleoperacija s prostorskim skaliranjem, kinestetično učenje in teleoperacija brez prostorskega skaliranja. So pa največ prostora za izboljšave videli pri teleoperaciji s prostorskim skaliranjem, sledili so teleoperacija brez prostorskega skaliranja, kinestetično učenje in CRT.

Med uporabo vizualnega skaliranja so operaterji najlažje uporabljali CRT, sledili so kinestetično učenje, teleoperacija s prostorskim skaliranjem in teleoperacija brez prostorskega skaliranja. Predvidevali so, da so najbolj natančni pri uporabi CRT, sledili so teleoperacija s prostorskim skaliranjem, kinestetično učenje in teleoperacija brez prostorskega skaliranja. Največ prostora za izboljšave pa so imeli pri pristopu teleoperacije s prostorskim skaliranjem, ki so mu sledili teleoperacija brez prostorskega skaliranja, kinestetično učenje in CRT.

Večina operaterjev je v odgovoru na prvo odprto vprašanje navedla, da bi uporabili pristop CRT, v kolikor bi morali demonstrirati fin gib. Uporabili bi ga, ker jim je omogočal, da so generirali gladko gibanje z uporabo nesignifikantnih vhodnih sil. Poleg tega je v odgovoru na drugo odprto vprašanje večina operaterjev navedla, da se jim zdi vizualno skaliranje uporabno, saj jim omogoča boljše natančnost pozicioniranja. Vendar pa je nezanemarljivo število operaterjev navedlo, da jim je funkcija vizualnega skaliranja tudi nekoliko povečala zahtevnost in psihološki stres zaradi večjega zaznavanja odstopanj pri pozicioniranju robota.

3.3 Diskusija

V tej študiji smo primerjali kinestetično učenje, teleoperacijo in CRT na dveh različnih nalogah, da bi ugotovili njihovo učinkovitost pri generiranju finih gibov. Implementirali smo funkcijo vizualnega skaliranja in njen vpliv preverili pri vsakem izmed pristopov. Preverili smo tudi vpliv prostorskega skaliranja pri pristopu teleoperacije. Različne pristope smo primerjali na podlagi pogreška pozicioniranja, hitrosti gibanja, gladkosti gibanja (SAL) in rezultatov vprašalnika. Čeprav je bil pogrešek pozicioniranja najpomembnejše merilo, na podlagi katerega smo primerjali pristope, so nam druge metrike omogočile pomemben dodaten vpogled. Hitrost in gladkost gibanja nam namreč nakazujejo stopnjo nadzora, ki so ga operaterji imeli nad robotom Panda. Manj negativne vrednosti SAL (tj. bolj gladko gibanje) in višje ravni hitrosti bi kazale na to, da so

operaterji lažje dosegali želeno natančnost pozicioniranja.

Naloga 1: premik od točke do točke

Med nalogo gibanja od točke do točke je iz porazdelitve pogreškov pozicioniranja (Slika 3.6) razvidno, da je z vsemi pristopi možno pozicionirati s submilimetrsko natančnostjo. Ko ni bilo uporabljenih nobenih dodatnih funkcij, je pristop CRT ustvaril najmanjši pogrešek pozicioniranja. Če primerjamo oba pristopa teleoperacije, je bil ustvarjen pogrešek pozicioniranja manjši, ko je bila uporabljena funkcija prostorskega skaliranja. To je posledica boljše motorične občutljivosti robotovega gibanja.

Poleg tega je uporaba vizualnega skaliranja pri vseh pristopih signifikantno zmanjšala generiran pogrešek pozicioniranja. To je lahko posledica lažje zaznave odstopanj, lahko pa je tudi posledica povečane previdnosti operaterja zaradi lažje zaznave odstopanj in posledično večje količine vloženega truda.

Generirani pogreški pri premiku iz začetne točke so bile na splošno višji od tistih med približevanjem cilju. Razlog za to bi lahko bil, da so operaterji začetno gibanje izvedli z manjšo previdnostjo, saj mu je sledilo vmesno gibanje, ki ni zahtevalo natančnosti pozicioniranja. Z uporabo vizualnega skaliranja pa se je pogrešek znatno zmanjšal, kar kaže na to, da so bili operaterji bolj pozorni na natančnost že od začetka demonstracije.

Prednost kinestetičnega učenja je mogoče opaziti pri vmesnem gibanju (Slika 3.7), saj je generirano gibanje v splošnem imelo višjo hitrost v primerjavi z drugimi pristopi. Razlog za to je, razmeroma enostavno generiranje večjih premikov. Prav tako je bilo gibanje pri kinestetičnem učenju najbolj gladko. Med približevanjem cilju pa se je za hitrejši pristop izkazal CRT. Kinestetično učenje ni bilo hitrejše zaradi potrebe po natančnosti pozicioniranja. Teleoperacija s prostorskim skaliranjem je generirala najpočasnejše gibanje, kar je posledica večje motorične občutljivosti.

Uporaba funkcije vizualnega skaliranja je zmanjšala generirano hitrost gibanja, kar je posledica večje previdnosti operaterjev zaradi zahtevane natančnosti. Vendar pa je imelo vizualno skaliranje manjši vpliv na hitrost in gladkost vmesnega gibanja, kar je posledica pomanjkanja referenčnega gibanja.

Naloga 2: sledenje poti

Razlika v generiranem pogrešku pozicioniranja je bolj očitna pri nalogi sledenja poti (Slika 3.8). V zadnjih treh segmentih (A_3 , A_4 in A_5) je CRT dosegel najmanjši pogrešek

pozicioniranja. S kinestetičnim učenjem je bil generiran submilimetrski pogrešek, a je bil občutno višji kot pri CRT. Razlika je očitnejša zaradi večje kompleksnosti naloge. Vsak segment je zahteval gibanje manjših prostorskih razsežnosti, kar je pri operaterjih predstavljalo večji izziv kot pa pri nalogi premika od točke do točke. Posledično se je hitrost gibanja od segmenta do segmenta znatno zmanjšala, da je bila dosežena ustrezna natančnost. Natančnost pozicioniranja se je izboljšala z uporabo vizualnega in prostorskega skaliranja. Brez prostorskega skaliranja se je teleoperacija odrezala slabše kot kinestetično učenje. S prostorskim skaliranjem pa se je ta pogrešek znatno zmanjšal. Povečana motorična občutljivost je omogočila natančnejši prenos giba vzdolž celotne poti. Izboljšanje teleoperacije je še bolj očitno, ko je uporabljena funkcija vizualnega skaliranja, saj je bila celo nekoliko boljša od pristopa CRT. Kinestetično učenje z vizualnim skaliranjem je prav tako doseglo podobno izboljšanje, saj je bil pogrešek pozicioniranja enak kot pri CRT. Zanimivo je, da je vizualno skaliranje prineslo malo koristi pristopu CRT. Lahko je mogoče, da so operaterji dosegli določeno raven natančnosti, ki jo je tudi z uporabo vizualnega skaliranja težko preseči.

S praktičnega vidika so rezultati, pridobljeni v tej študiji na meji možne dosežene natančnosti zaradi zunanjih dejavnikov, kot so nezanimljiva širina referenčne poti, širina konice vrha robota, velikost robotskega manipulatorja in pogreškov kinematičnega modela.

Zaključek

Na splošno je CRT dosegel najvišjo stopnjo natančnosti, pri čemer je omogočal razmeroma hitre in gladke gibe. Njegova zmogljivost je razvidna tudi iz rezultatov vprašalnika (Slika 3.9), saj je dosegel najboljšo oceno za enostavnost uporabe in sposobnost zaznane natančnosti. Bil je tudi najprimernejši pristop na podlagi prvega odprtega vprašanja, saj po mnenju operaterjev omogoča generiranje gladkih gibov brez pretiranega truda.

Noben izmed pristopov teleoperacije ni prejel visoke ocene glede enostavnosti uporabe, kar je posledica večje kompleksnosti nadzornih naprav. Opazen je bil vpliv prostorskega skaliranja, saj je teleoperacija s prostorskim skaliranjem dosegla višjo oceno zaznane natančnosti kot tista brez nje. Oba pristopa sta prejela tudi visoko oceno glede možnega nadaljnjega izboljšanja. Operaterji so menili, da bi lahko sčasoma še izboljšali natančnost pozicioniranja, kar pa ni veljalo za kinestetično učenje in CRT. Tudi to je posledica večje kompleksnosti nadzornega sistema teleoperacije.

Kinestetično učenje je zaradi nepopolne kompenzacije dinamičnega modela zah-

tevalo več truda za natančno pozicioniranje, zaradi česar je prejelo nižjo oceno za enostavnost uporabe in zaznano natančnost. Kljub temu pa so operaterji konsistentno dosegali submilimetrsko natančnost pozicioniranja, zaradi česar lahko zaključimo, da je kinestetično učenje primerno za demonstracijo finih gibov.

Uporaba vizualnega skaliranja je pri vseh pristopih pripomogla k signifikantnem zmanjšanju pozicijskih pogreškov. Prav tako je večina operaterjev v sklopu drugega odprtega vprašanja potrdila uporabnost te funkcionalnosti, saj naj bi povečano zaznavanje odstopanj pozicioniranja odtehtalo povečano zahtevnost upravljanja. Na podlagi teh ugotovitev lahko zaključimo, da lahko ta funkcionalnost služi kot uporabno orodje tekom nadaljnjih raziskav.

Med udeleženci je bila podobna zastopanost moških in žensk ter strokovnjakov in nestrokovnjakov. Skupna uspešnost ni bila odvisna od spola ali izkušenj. To pomeni, da je te rezultate mogoče doseči z operaterjem, ki nima predhodnih izkušenj z robotiko. Potrebno pa se je zavedati, da vsi operaterji potrebujejo določeno stopnjo uvajanja. Le tako se lahko namreč privadijo na izbran pristop upravljanja in se naučijo določenih dobrih praks, ki pripomorejo k natančnosti izvedenih demonstracij. V sklopu predstavlja študije je bil primer dobre prakse opazovati vrh robota pod čim manjšim kotom (tj. kot med normalo ravnine prikaza in smerjo pogleda) za namene minimiziranja vpliva perspektive. Operaterji so pogosto ob začetku opazovali vrh pod relativno visokim kotom in posledično generirali visoke pogreške pozicioniranja glede na referenčni prikaz. Vrh robota je bil namreč oddaljen približno 2 mm vzdolž osi z od ravnine s prikazano referenco. V kolikor je operater opazoval vrh robota pod kotom 45° se je vrh preslikal na ravnino s pogreškom vrednosti 2 mm. Z zmanjšanjem kota na 20° pa se ta vrednost zmanjša na 0,35 mm. Vpliv perspektive je tako nezanemarljiv in prisoten tekom demonstracije vsakega neizkušenega operaterja. Z zadostnim procesom učenja pa se operater privadi in nauči prepoznavati in posledično minimizirati njen vpliv.

Kljub obetavnim rezultatom demonstracije finih gibov pa se je potrebno zavedati, da so imeli operaterji pri izvedbi gibov določene olajševalne okoliščine. Zahtevane gibe so namreč opravljali v ravnini, hkrati pa jim tudi ni bilo potrebno demonstrirati orientacije vrha robota. Za nadaljne študije bi bilo tako potrebno razviti obogateno okolje, ki omogoča prikaz pozicije in orientacije gibov izven ravnine, hkrati pa bi bilo potrebno za namene demonstracije orientacije, v primeru pristopov CRT in teleoperacija, sistem upravljanja dodatno nadgraditi.

4 Analitični zapis finih robotskih gibov

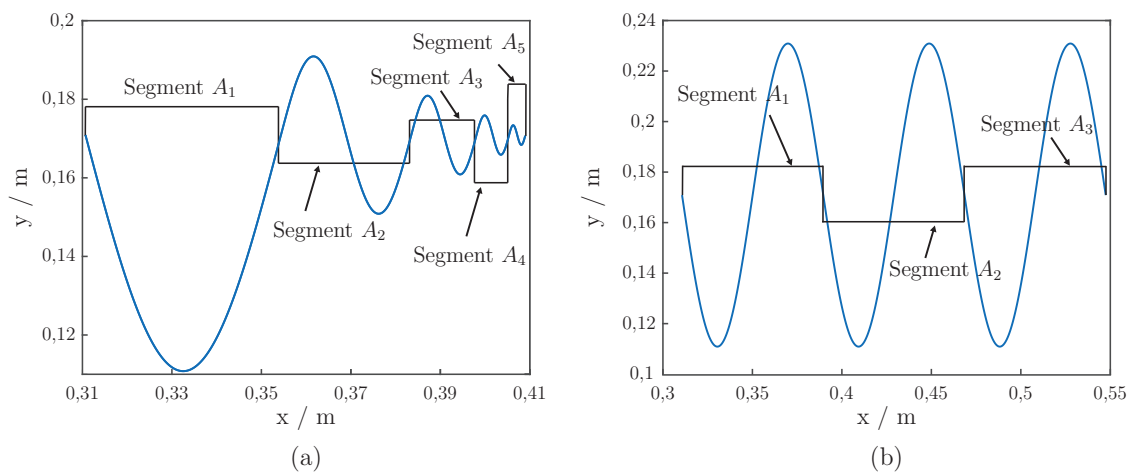
Pri učenju z demonstracijo je poleg načina izvedbe demonstracije pomembna tudi izbira metod za zapis izvedenih demonstracij, kar omogoča nadaljnje posploševanje iz množice demonstracij in prilagajanje naučenih trajektorij. V grobem so metode lahko namenjene visokonivojskemu izločanju konteksta ali nizkonivojskemu zapisu trajektorije. Za zapis finih gibov so zanimive slednje. Metodi dinamični generatorji gibov (angl. dynamic movement primitives - DMP) in verjetnostni generatorji gibov (angl. probabilistic movement primitives - ProMP) temeljita na parametričnem zapisu trajektorije in sta primerni za opise trajektorij kot funkcije ene vhodne spremenljivke (npr. čas) [17, 54]. Metoda mešani Gaussovi modeli (angl. Gaussian mixture models - GMM) omogoča zapis trajektorij kot funkcije več vhodnih spremenljivk [18, 19]. Čeprav je tak zapis uporaben v določenih aplikacijah, kompleksnost modela otežuje naknadno prilagajanje naučenega. Razvijajo se tudi nove metode, kot so jedrni generatorji gibov (angl. kernelized movement primitives - KMP), ki uporabljajo statistične lastnosti ProMP in GMM metod, a omogočajo tudi prilagajanje vmesnih točk [6].

Za namene analize primernosti posameznih generatorjev gibov pri učenju finih gibov smo v sklopu študije primerjali uveljavljeni metodi DMP in GMM. Ocenjevali smo njuno sposobnost zapisa finih gibov znotraj demonstracij, ki vsebujejo tudi grobe gibe in preverjali vpliv prilagajanja parametrov metod na zapis finih gibov. Predlagali smo tudi nadgradnjo metode GMM, ki izboljša zapis finih gibov na podlagi frekvenčne analize demonstriranih trajektorij.

4.1 Izbira in simulacija trajektorij za analizo metod

Za ustrezno primerjavo metod smo potrebovali dobro definirane trajektorije. V sklopu študije smo kot učne množice uporabili krivulje, ki niso vsebovale informacij o zahtevanem časovnem poteku in bi se potemtakem na njih morali navezovati z izrazom pot. V splošnem pa so metode učenja namenjene zapisu in generiranju trajektorij, pri katerih pa je relevanten tudi časovni potek. Zato smo se odločili, da se v sklopu te študije, za potrebe jasnosti, izognemu izmenični uporabi izrazov pot in trajektorija in vseskozi uporabljamo izraz trajektorija. Pri tem pa se jasno zavedamo, da učne množice niso podajale informacije o zahtevanem časovnem poteku.

Učno množico trajektorij smo generirali programsko, saj demonstriranje trajektorij vnese neželena odstopanja, ki bi nam oteževala splošno analizo metod zapisa. Za referenco smo izbrali trajektorijo iz študije demonstriranja finih gibov (Slika 4.1, a). Izbrana trajektorija namreč vsebuje tako večje premike, ki predstavljajo grobe gibe (segmenti A_1 , A_2 , A_3), kot tudi manjše premike, ki predstavljajo fine gibe (segmenti A_4 , A_5). Dodatno smo metode želeli analizirati tudi na trajektoriji, ki vsebuje zgolj grobe gibe. Tako smo definirali trajektorijo (Slika 4.1, b), ki je sestavljena iz treh enakih segmentov A_1 izvorne trajektorije (Slika 4.1, a).



Slika 4.1: Prikaz trajektorije s finimi gibi (a) in trajektorije brez finih gibov (b).

Metoda DMP za zapis potrebuje zgolj eno učno trajektorijo, medtem ko jih metoda GMM potrebuje več. Tako smo za analizo metode GMM programsko generirali nabor 10-ih trajektorij, pri čemer smo izvornemu poteku dodali še nesignifikanten šum. Šum smo dodali za ustrezno delovanje metode, saj smo tako preprečili nastanek ničelne kovariančne matrike, ki bi bila posledica nabora enakih trajektorij (enačba 4.4).

V nadaljevanju se bomo na te trajektorije navezovali kot na demonstracije, kljub temu da so trajektorije generirane programsko in niso demonstrirane s strani operaterja.

4.2 Metoda dinamičnih generatorjev gibov

DMP je parametrična metoda, ki demonstracijo zapiše kot sistem vzmet-dušilnik z dodano nelinearno funkcijo, ki določa obliko trajektorije [17]. Potek trajektorije je tako opisan z nelinearno diferencialno enačbo 2. reda

$$\tau \ddot{\mathbf{x}}_{\mathbf{r}}(t) = \alpha_p(\beta_p(\mathbf{g} - \mathbf{x}_{\mathbf{r}}(t)) - \tau \dot{\mathbf{x}}_{\mathbf{r}}(t)) + \mathbf{f}(s), \quad (4.1)$$

kjer τ predstavlja časovno skaliranje, $\mathbf{x}_{\mathbf{r}}(t)$, $\dot{\mathbf{x}}_{\mathbf{r}}(t)$ in $\ddot{\mathbf{x}}_{\mathbf{r}}(t)$ predstavljajo izračunano lego, hitrost in pospešek funkcije DMP, \mathbf{g} predstavlja točkovni atraktor sistema vzmet-dušilnik, parametra α_p in β_p pa določata dinamiko odziva tega sistema.

Nelinearna funkcija $\mathbf{f}(s)$ predstavlja sistem vsiljenih sil in določa obliko trajektorije. Posamezna komponenta nelinearne funkcije $\mathbf{f}(s)$ (npr. $f_x(s)$ za os x v zunanjih koordinatah), pa je določena kot

$$f_x(s) = \frac{\sum_{i=1}^N \omega_i \psi_i(s)}{\sum_{i=1}^N \psi_i(s)} s, \quad \psi_i(s) = e^{-h_i(s-c_i)^2} \quad (4.2)$$

Ta aproksimira potek posamezne prostostne komponente demonstracije z linearno kombinacijo N -tih Gaussovih baznih funkcij $\psi_i(s)$ in pripadajočih uteži ω_i , pri čemer so uteži naučene z uporabo linearne regresije. Potek baznih funkcij je določen s parametroma h_i in c_i , saj h_i določa širino posamezne bazne funkcije, c_i pa njeno pozicijo vzdolž spremenljivke faznega poteka s .

Fazno spremenljivko s uporabimo, da se izognemo eksplicitni časovni odvisnosti funkcije $\mathbf{f}(s)$. Spremenljivka je definirana prek kanoničnega sistema

$$\tau \dot{s} = -\alpha_s s \quad (4.3)$$

in iz začetne vrednosti $s(0) = 1$ monotono konvergira proti 0. Posledično tudi funkcija $\mathbf{f}(s)$ z zmanjšanjem vrednosti s izzveni.

To omogoča, da metoda DMP iz vsake začetne lege $\mathbf{x}_{\mathbf{r}}(0)$ generira gladko trajektorijo, katere obliko definira $\mathbf{f}(s)$ in se ob končanem vplivu člena vsiljenih sil zaključi v \mathbf{g} . DMP lahko zapiše poljubne informacije o gibanju (npr. pozicija, orientacija), vendar

pa je bila v sklopu naše študije orientacija konstantna, tako da je bil zadosten zapis pozicije.

Glavne prednosti uporabe metode DMP so hiter proces učenja, saj je potrebna zgolj ena demonstracija, in možnost prilagajanja začetne in končne pozicije generirane trajektorije. Glavna slabost je ta, da metoda ni verjetnostna, kar pomeni, da ni podane informacije o raztrosu demonstracij, ki je lahko uporabna informacija pri načrtovanju sistemov vodenja (npr. prilagajanje togosti robotskega manipulatorja glede na rezultirajočo varianco). Prav tako je potrebno za zapis DMP-ja na podlagi večih demonstracij uporabiti pristope strojnega učenja.

Na kvaliteto rezultirajoče trajektorije lahko vplivamo s spreminjanjem števila baznih funkcij N . Manjše število baznih funkcij povzroči slabši zapis demonstracije, a hkrati preprečuje zapis manjših neželenih potekov, ki so lahko posledica tresavice ali nepopolnega nadzora nad gibanjem robota in so lahko tretirane kot šum. Višje število baznih funkcij po drugi strani predstavlja podrobnejši opis dane demonstracije, kar je uporabno predvsem takrat, ko demonstracija vsebuje tudi fine gibe.

4.3 Metoda mešanih Gaussovih modelov

Metoda mešanih Gaussovih modelov je parametrična metoda, ki temelji na verjetnostnem opisu podatkov, pri čemer so podatki zapisani z N Gaussovimi funkcijami (t.i. komponentami)

$$\mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \frac{1}{\sqrt{(2\pi)^N |\boldsymbol{\Sigma}_i|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)}, \quad (4.4)$$

pri čemer $\boldsymbol{\mu}_i$ predstavlja srednjo vrednost i -te komponente, $\boldsymbol{\Sigma}_i$ pa njeno kovariančno matriko [18, 19]. Trajektorija je tako zapisana z vsoto 2D Gaussovih funkcij $\mathcal{N}(x | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ in pripadajočih uteži ϕ_i

$$p(\mathbf{x}) = \sum_{i=1}^N \phi_i \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \quad \sum_{i=1}^N \phi_i = 1, \quad (4.5)$$

pri čemer se vrednosti ϕ_i , $\boldsymbol{\mu}_i$ in $\boldsymbol{\Sigma}_i$ določijo tekom procesa učenja.

Proces učenja temelji na iterativni metodi oceni-maksimiziraj (angl. expectation-maximization - EM), kjer parametrom modela (ϕ_i , $\boldsymbol{\mu}_i$ in $\boldsymbol{\Sigma}_i$) določimo začetne vrednosti, nato pa se te vrednosti iterativno prilagajajo tako, da model čim boljše opiše dane podatke. Opis danih podatkov ocenimo z metodo največjega verjetja (angl. maximum

likelihood estimation - MLE). Na podlagi naučenega modela nato generiramo rezultirajočo trajektorijo z uporabo Gaussove mešane regresije (angl. Gaussian mixture regression - GMR).

Na kvaliteto opisa danih podatkov ima velik vpliv izbira števila komponent N . V splošnem manjše število komponent pomeni, da posamezna komponenta opiše večji del podatkov, kar oteži zapis podrobnosti. Po drugi strani, pa preveliko število komponent močno poveča kompleksnost modela in podaljša čas učenja. Število potrebnih komponent je tako potrebno prilagoditi potrebam študije. Proces določanja si lahko olajšamo z uporabo metod, kot sta Bayesov informacijski kriterij (angl. Bayesian information criterion - BIC) in Akaikov informacijski kriterij (angl. Akaike information criterion - AIC), ki služita kot cenilki. Podata namreč informacijo o kvaliteti opisanih podatkov ob uporabi izbranega števila komponent [55].

4.4 Nadgradnja metode GMM s prilagajanjem gostote komponent na podlagi frekvenčne analize

Pri inicializaciji parametrov modela GMM so v splošnem komponente porazdeljene enakomerno vzdolž neodvisne spremenljivke podatkov (npr. normiran čas t). To je v redu, saj se porazdelitev prilagodi tekom izvedbe metode oceni-maksimiziraj, ki podatkom prilagaja lego posamezne komponente. Ker pa vsaka komponenta opiše veliko število podatkovnih točk, se lahko podrobnosti izgubijo, kar je problem, ko te podrobnosti predstavljajo relevantno informacijo (npr. fini gibi v sklopu splošne trajektorije). Eden od načinov reševanja te težave je večanje števila komponent K . Z večanjem števila komponent, posamezna komponenta opiše manjše število točk in podrobnosti tako lažje pridejo do izraza. Vendar pa povečanje števila komponent znatno poveča kompleksnost modela in potreben čas učenja. Zato smo v sklopu študije razvili metodo, ki namesto spreminjanja števila komponent prilagodi začetno porazdelitev komponent tako, da je njihova gostota večja na območjih z relevantnimi informacijami in manjša drugje. V našem primeru nabora trajektorij s finimi gibi, relevantno informacijo predstavlja potek v sklopu segmentov A_4 in A_5 (Slika 4.1, a).

Za doseg tega cilja, smo vzdolž trajektorije določili dominantne frekvence $f_d(t)$, saj območja z višjimi frekvencami predstavljajo fino gibanje in s tem relevantno informacijo. Za namene lažje frekvenčne analize, smo zapis trajektorije spremenili v polarni

zapis. Trajektorija je namesto spremenljivk $x(t)$ in $y(t)$, tako funkcija kota $\varphi(t)$. Za odstranitev irelevantnega šuma, smo novo definirano trajektorijo $\varphi(t)$ obdelali z nizkopropustim filtrom (mejna frekvenca 10 Hz), na obdelani trajektoriji, pa nato izvedli frekvenčno analizo z uporabo Welchove metode. Ta za vsak časovni korak t_i izračuna frekvenčni spekter $\phi_i(\omega)$, kar rezultira v spektrogramu $\mathbf{S} = [\phi_1(\omega), \phi_2(\omega), \dots, \phi_N(\omega)]$, kjer N predstavlja število izračunanih frekvenčnih spektrov (Slika 4.2, a) [56]. $\mathbf{S}(t, \omega)$ nadalje prilagodimo tako, da amplitude spektrov posameznih diskretnih frekvenc ω_j normiramo vzdolž časovne osi

$$\widehat{\mathbf{S}}(t, \omega_i) = \frac{\mathbf{S}(t, \omega_i)}{\max\{\mathbf{S}(t, \omega_i)\}}. \quad (4.6)$$

To nam izboljša zaznavo višjih frekvenc, katerih amplituda je v izvornem spektrogramu $\mathbf{S}(t, \omega)$ zanemarljiva v primerjavi z nizkimi frekvencami. Posledično je v normiranem spektrogramu $\widehat{\mathbf{S}}(t, \omega)$ olajšana zaznava prehoda signala iz nižjih k višjim frekvencam in obratno.

Nadalje spektrogram $\widehat{\mathbf{S}}$ z uporabo časovnih oken in frekvenčnih pasov dodatno diskretiziramo v več časovno-frekvenčnih polj $\mathbf{s}_{m,n}$

$$\widehat{\mathbf{S}} = \begin{bmatrix} \mathbf{s}_{1,1} & \dots & \mathbf{s}_{1,n} & \dots & \mathbf{s}_{1,\Omega} \\ \vdots & \ddots & \vdots & & \vdots \\ \mathbf{s}_{m,1} & \dots & \mathbf{s}_{m,n} & \dots & \mathbf{s}_{m,\Omega} \\ \vdots & & \vdots & \ddots & \vdots \\ \mathbf{s}_{T,1} & \dots & \mathbf{s}_{T,n} & \dots & \mathbf{s}_{T,\Omega} \end{bmatrix}, \quad (4.7)$$

$$\mathbf{s}_{m,n} = \begin{bmatrix} \widehat{\mathbf{S}}(\Delta T(m-1), \Delta\Omega(n-1)) & \dots \\ \vdots & \widehat{\mathbf{S}}(\Delta Tm, \Delta\Omega n) \end{bmatrix}, \quad (4.8)$$

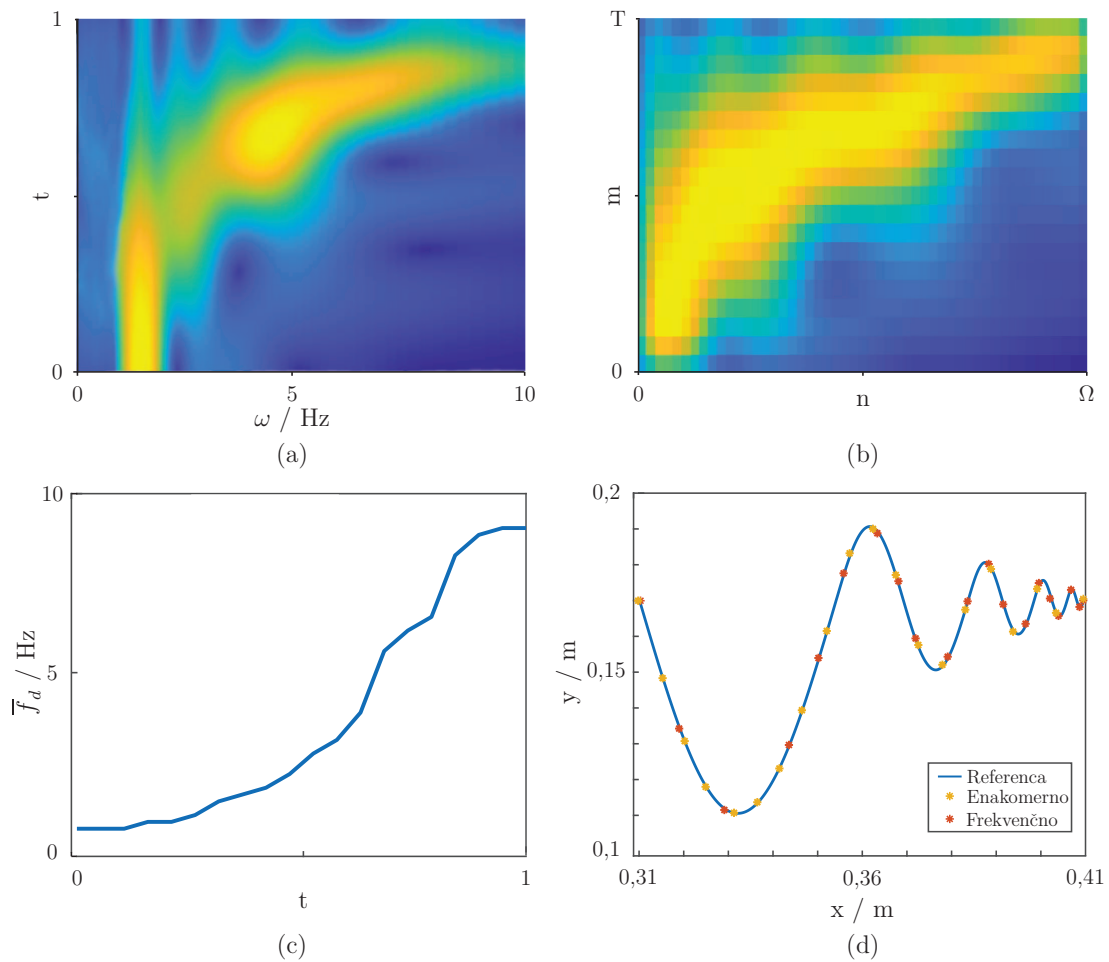
pri čemer T predstavlja število časovnih oken in Ω število frekvenčnih pasov, medtem ko ΔT predstavlja dolžino časovnega okna in $\Delta\Omega$ obseg frekvenčnega pasu. Vsakemu časovno-frekvenčnemu polju $\mathbf{s}_{m,n}$ določimo srednjo vrednost

$$\mu_{m,n} = \bar{\mathbf{s}}_{m,n}, \quad (4.9)$$

na podlagi katere lahko za vsako časovno okno m , določimo časovno-frekvenčno polje $\mathbf{s}_{m,n}$ z največjo izračunano srednjo vrednostjo $\mu_{m,n}$. To polje namreč predstavlja dominantni frekvenčni pas $f_d(m)$ v izbranem časovnem oknu. Potek dominantne frekvence trajektorije vzdolž časovnih oken

$$f_d(m) = [f_{d1}, \dots, f_{dT}] \quad (4.10)$$

nato interpoliramo, da dobimo potek $f_d(t)$ z zadostno časovno ločljivostjo, ki je potrebna za ustrezen izračun izhodiščnih leg komponent vzdolž časovne osi. Na podlagi $f_d(t)$ namreč določamo porazdelitev komponent mešanega Gaussovega modela. Ker pa metoda GMM za zapis uporabi več demonstracij, sta frekvenčna analiza in izračun $f_d(t)$ potrebna za vsako trajektorijo posebj. Na podlagi vseh izračunanih potekov dominantnih frekvenc določimo $\bar{f}_d(t)$, ki predstavlja srednjo vrednost vseh potekov.



Slika 4.2: Prikaz frekvenčne analize izbrane trajektorije z uporabo Welchove transformacije (a), normiranega spektrograma $\hat{S}(t, \omega)$ (b), potek srednje vrednosti dominantnih frekvenc $\bar{f}_d(t)$ (c) in porazdelitev komponent mešanega Gaussovega modela vzdolž trajektorije (d).

V našem primeru višja frekvenca predstavlja relevantno informacijo, zato bi morala biti gostota komponent na teh območjih višja. Ustrezno porazdelitev komponent

dosežemo tako, da izračunamo ploščino pod funkcijo $\bar{f}_d(t)$

$$A = \sum_{t=1}^M \bar{f}_d(t) \Delta t \quad (4.11)$$

in izračunamo ploščinski korak

$$\Delta A = \frac{A}{K}, \quad (4.12)$$

pri čemer K predstavlja izbrano število komponent, M pa število vseh časovnih korakov. Komponente nato razporedimo vzdolž časovne osi t tako, da je izračunana površina pod funkcijo $\bar{f}_d(t)$ med dvema zaporednima komponentama enaka ΔA . Tako bo gostota komponent na območjih z višjo vrednostjo $\bar{f}_d(t)$ večja. Na ta način dosežemo, da je lega komponent prilagojena poteku podatkov še pred začetkom postopka oceni-maksimiziraj.

4.5 Rezultati

V tem poglavju bomo predstavili delovanje metod za zapis dveh različnih trajektorij. Posamezno metodo smo uporabili za zapis trajektorije s finimi gibi (Slika 4.1, a), ter za zapis trajektorije brez finih gibov (Slika 4.1, b). Pri tem smo analizirali vpliv parametrov na kvaliteto zapisa, pri čemer smo kvaliteto zapisa določali na podlagi izračunanih pogreškov $\tilde{p}(t)$ med rezultirajočo trajektorijo $\mathbf{p}(t)$ in referenčno trajektorijo $\mathbf{p}_r(t)$

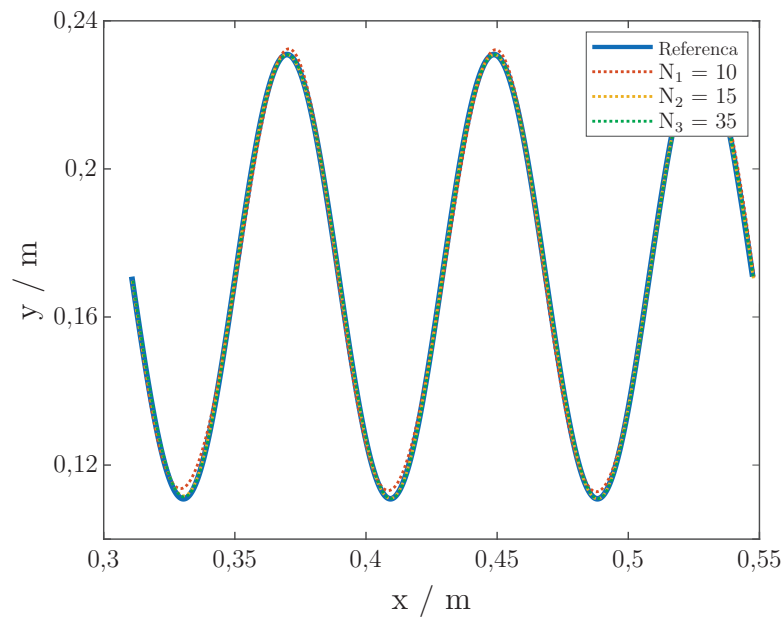
$$\tilde{p}(t) = \|\mathbf{p}(t) - \mathbf{p}_r(t)\|. \quad (4.13)$$

Za lažjo analizo, smo izračunana odstopanja združili po segmentih prikazanih na sliki 4.1. Rezultate smo tudi statistično ovrednostili z uporabo metod predstavljenih v poglavju 3.1.5. Rezultati statistične analize so prikazani v sklopu grafov s škatličnimi diagrami, pri čemer * predstavlja statistično signifikantno spremembo.

4.5.1 Dinamični generatorji gibov

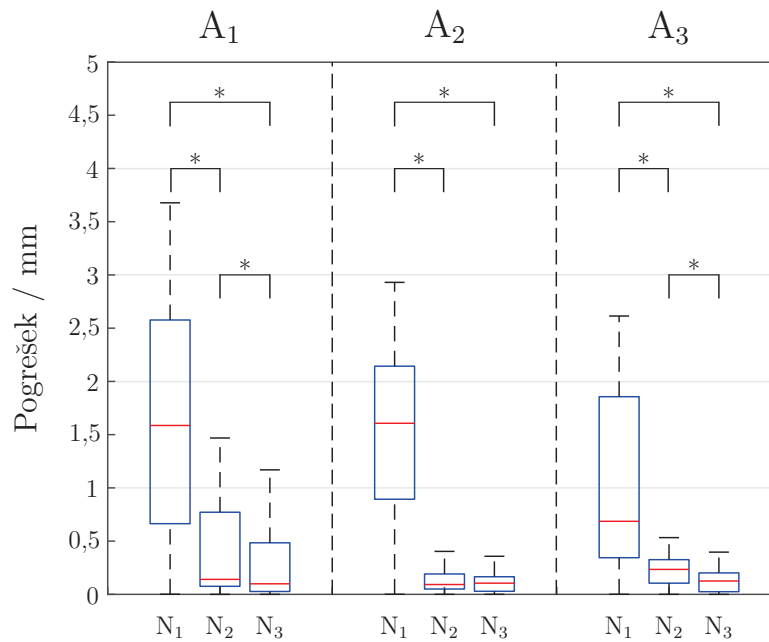
V primeru metode DMP referenčno trajektorijo zapišemo v oblike nelinearne diferencialne enačbe drugega reda pri čemer lahko na zapis vplivamo z izbiro števila baznih funkcij N . Tako smo v sklopu študije preizkusili delovanje metode, pri treh različnih številih baznih funkcij. Začetne vrednosti N smo določili na podlagi rezultatov zapisa trajektorije brez finih gibov, pridobljeno znanje pa nato uporabili za zapis trajektorije s finimi gibi.

Zapis trajektorije brez finih gibov. V sklopu študije bomo prikazali delovanje metode pri zapisu trajektorije brez finih gibov z uporabo $N_1 = 10$, $N_2 = 15$ in $N_3 = 35$ baznih funkcij. Na sliki 4.3 so tako prikazani poteki rezultirajočih trajektorij ob uporabi posameznega števila baznih funkcij. Razvidno je, da poteki rezultirajočih trajektorij sledijo poteku referenčne trajektorije, pri čemer so odstopanja v primeru N_2 in N_3 neopazna. V primeru N_1 pa je odstopanje od referenčne trajektorije opazno na območjih spremembe smeri trajektorije.



Slika 4.3: Prikaz zapisov trajektorije brez finih gibov pri uporabi metode DMP z N baznimi funkcijami.

Podobno je razvidno tudi iz prikaza izračunanih pogreškov, ki smo jih na sliki 4.4 prikazali z uporabo škatličnih diagramov. Razvidno je, da sta vzdolž celotne trajektorije, tako mediana kot tudi maksimalna vrednost pogreškov, pri N_1 občutno večji kot pri N_2 in N_3 . Manjša vrednost pogreškov je razumljiva, saj z večanjem števila baznih funkcij zmanjšamo število podatkovnih točk, ki so zapisane s posamezno bazno funkcijo, zaradi česar se model bolje prilagodi danim podatkom. Tako je bilo očitno število baznih funkcij N_1 premajhno za ustrezen zapis trajektorije. To sklepamo, ker smo že z majhnim povečanjem števila baznih funkcij N_2 bistveno zmanjšali generirane pogreške. S še nadaljnjim povečevanjem števila baznih funkcij pa ni nujno, da se bo vrednost pogreškov nadalje zmanjševala. Tako je na primer pri N_3 , število baznih funkcij mnogo večje kot v primeru N_2 , a je povprečna vrednost median pogreškov vzdolž vseh segmentov nižja za zgolj 0,05 mm. Tako lahko zaključimo, da kljub temu, da se

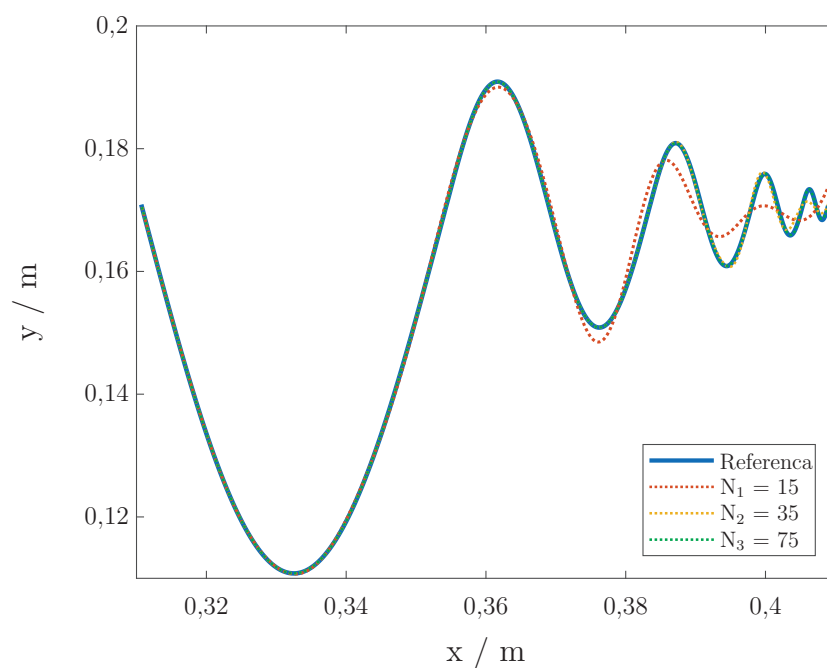


Slika 4.4: Prikaz generiranih pogreškov v primeru trajektorije brez finih gibov vzdolž posameznih segmentov (A_1 , A_2 , A_3) pri uporabi metode DMP z N baznimi funkcijami.

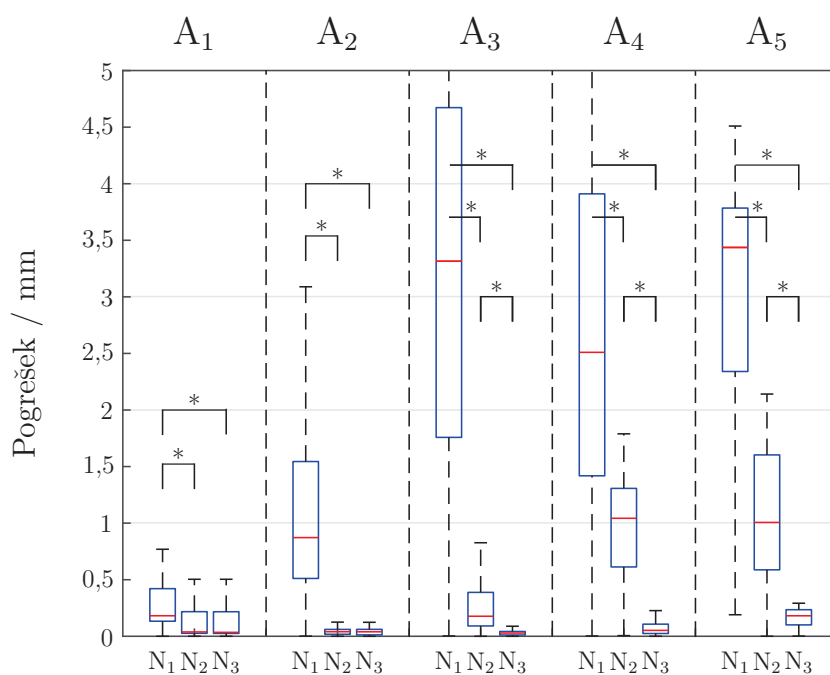
z nadaljnjim povečevanjem baznih funkcij vrednost generiranih pogreškov zmanjšuje, izbrano trajektorijo brez finih gibov zadostno opišemo že z N_2 številom baznih funkcij.

Zapis trajektorije s finimi gibi. Na podlagi ugotovitev pri zapisu trajektorij brez finih gibov, smo za zapis trajektorij s finimi gibi določili izhodiščno število baznih funkcij N_1 , ter posodobili še N_2 in N_3 ($N_1 = 15$, $N_2 = 35$ in $N_3 = 75$). Rezultirajoče trajektorije so prikazane na sliki 4.5. Razvidno je, da število baznih funkcij N_1 , ki je zadostovalo za zapis trajektorije brez finih gibov, v primeru te trajektorije ne zadostuje za zadosten zapis celotne trajektorije. Kljub temu, da je zapis segmenta A_1 ustrezen, se namreč odstopanja tekom segmentov začnejo povečevati do te mere, da je zapis segmentov A_4 in A_5 neprimeren. Zapis se izboljša s povečevanjem baznih funkcij, pri čemer v primeru N_2 še vedno prihaja do opaznega odstopanja v segmentu A_5 . To odstopanje je v primeru uporabe N_3 baznih funkcij neopazno, kar nakazuje, da je referenčna trajektorija ustrezno opisana.

To je razvidno tudi iz slike 4.6, ki prikazuje pogreške vzdolž posameznega segmenta. V primeru N_1 so tako pogreški v segmentih A_2 - A_5 občutno večji od pogojev N_2 in N_3 . V primeru N_2 so pogreški v segmentih A_1 in A_2 znotraj 1 mm in podobna N_3 , a se povečajo v segmentih A_4 - A_5 . V primeru N_3 pa so pogreški sprejemljivi in znotraj 1 mm tekom vseh segmentov.



Slika 4.5: Prikaz zapisov trajektorije s finimi gibi pri uporabi metode DMP z N baznimi funkcijami.

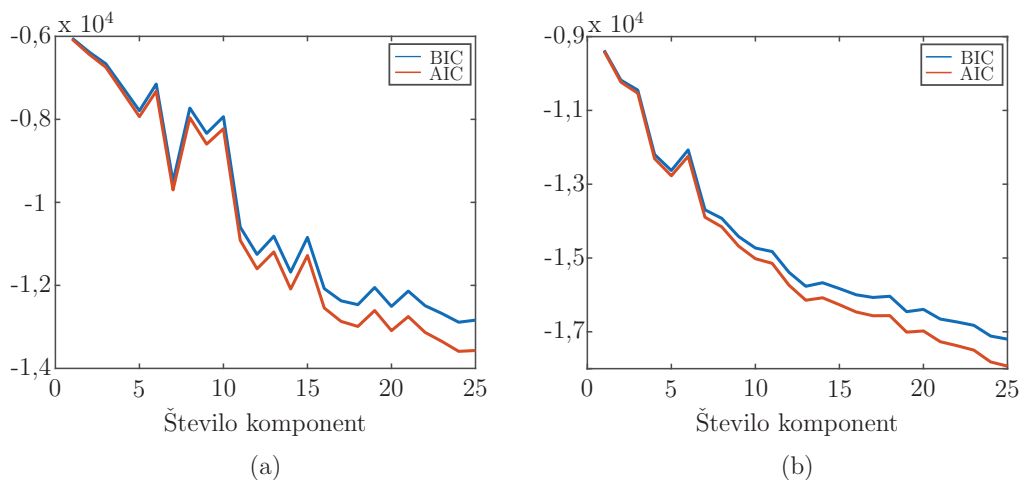


Slika 4.6: Prikaz generiranih pogreškov v primeru trajektorije s finimi gibi vzdolž posameznih segmentov ($A_1 - A_5$) pri uporabi metode DMP z N baznimi funkcijami.

V primeru metode DMP je tako mogoče zapisati fine gibe z zadostnim povečevanjem števila baznih funkcij, saj je metoda iz stališča računske kompleksnosti ugodna in omogoča hitro izračunavanje modela. Se pa moramo pri tem zavedati, da se s povečevanjem izboljšuje tudi opis neželenih potekov, ki so lahko smatrani kot šum.

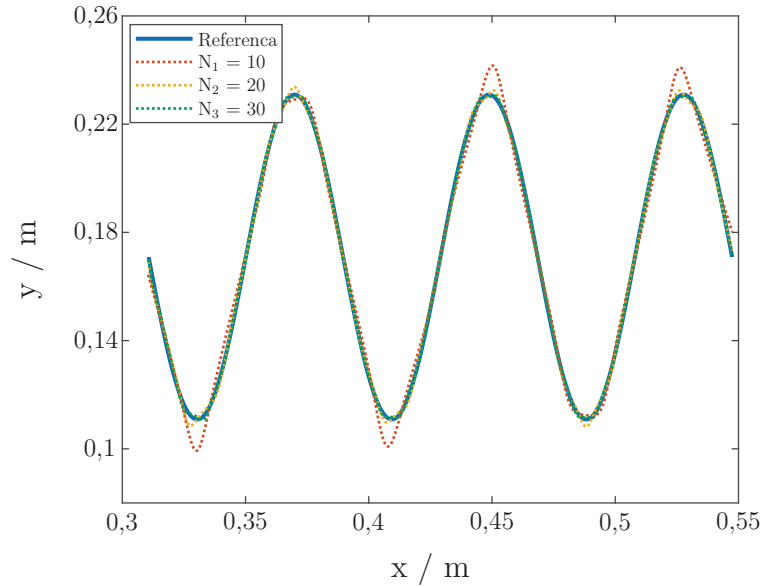
4.5.2 Mešani Gaussovi modeli

V primeru metode GMM, referenčno trajektorijo zapišemo v obliki verjetnostnega opisa podatkov z uporabo N komponent, pri čemer število komponent vpliva na kvaliteto zapisa. Tako smo v sklopu študije preizkusili delovanje metode pri različnih številih komponent. Za okvirno oceno primerne števila izbranih komponent smo uporabili metodi BIC in AIC. Ti metodi podata informacijo o kvaliteti opisa podatkov, pri uporabi izbranega števila komponent. Obe metodi smo uporabili tako na naboru trajektorij brez finih gibov, kot tudi na naboru trajektorij s finimi gibi. Potek vrednosti BIC in AIC glede na število uporabljenih komponent prikazujemo na sliki 4.7. Manjša vrednost funkcije predstavlja bolj opisan model, tako da lahko na podlagi obeh grafov sklepamo, da je model bolj opisan s povečevanjem števila komponent, kar je pričakovano. Tako za trajektorijo brez finih gibov (Slika 4.7, a), kot za trajektorijo s finimi gibi (4.7, b), velja, da je zmanjševanje bolj intenzivno pri manjšem številu komponent, z njihovim večanjem, pa se potek metod BIC in AIC umirja. To nakazuje, da v kolikor je komponent 10 ali manj, vsaka dodatna opazno pripomore k opisu podatkov, zaradi česar za opis podatkov ni smiselno uporabiti manj kot 10 komponent.

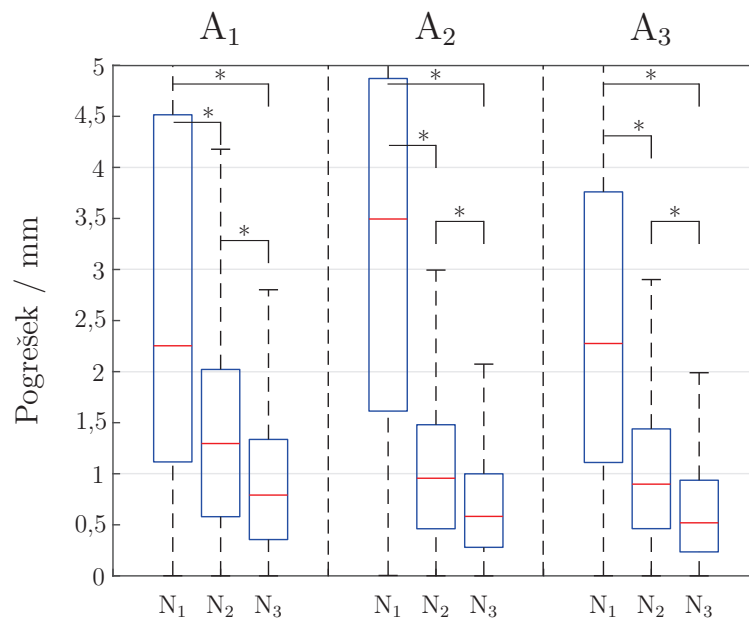


Slika 4.7: Prikaz poteka vrednosti BIC in AIC glede na število komponent pri trajektoriji brez finih gibov (a) in trajektoriji s finimi gibi (b).

Zapis trajektorije brez finih gibov. V sklopu študije smo se tako odločili prikazati delovanje metode GMM pri zapisu nabora trajektorij brez finih gibov z uporabo $N_1 = 10$, $N_2 = 20$ in $N_3 = 30$ komponent. Na sliki 4.8 so tako prikazani poteki rezultirajočih trajektorij ob uporabi posameznega števila komponent. Razvidno je,



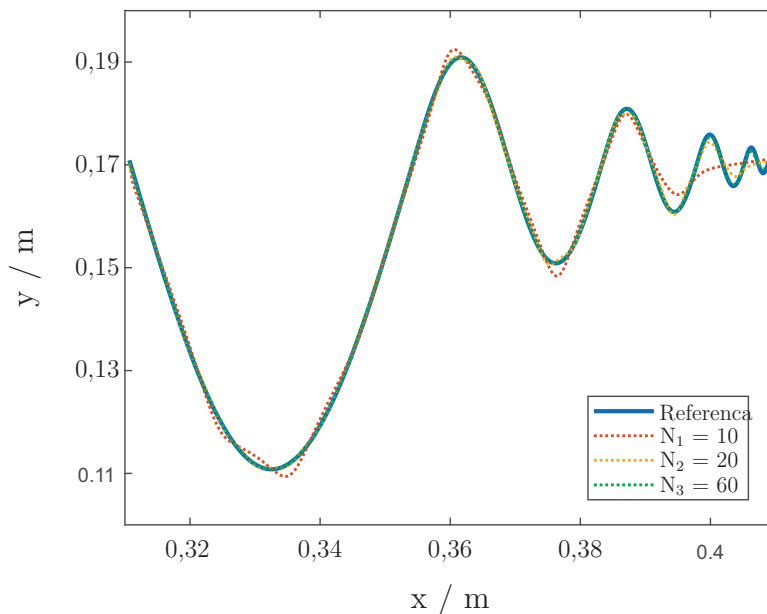
Slika 4.8: Prikaz zapisov trajektorije brez finih gibov pri uporabi metode GMM z N komponentami.



Slika 4.9: Prikaz generiranih pogreškov v primeru trajektorije brez finih gibov vzdolž posameznih segmentov (A_1 , A_2 , A_3) pri uporabi metode GMM z N komponentami.

da poteki rezultirajočih trajektorij sledijo poteku referenčne trajektorije, pri čemer so odstopanja v primeru N_1 očitna, v primeru N_2 manj opazna in v primeru N_3 neopazna, do odstopanj pa prihaja na območjih spremembe smeri trajektorije. Podobno je razvidno tudi iz prikaza izračunanih pogreškov, ki smo jih na sliki 4.9 prikazali z uporabo škatličnih diagramov. Razvidno je, da se vzdolž celotne trajektorije, tako mediana kot tudi maksimalna vrednost pogreškov zmanjšujeta s povečevanjem števila komponent. Zmanjšana vrednost pogreškov je razumljiva, saj z večanjem števila komponent zmanjšamo število podatkovnih točk, ki so opisane s posamezno komponento, zaradi česar se model bolje prilagodi danim podatkom.

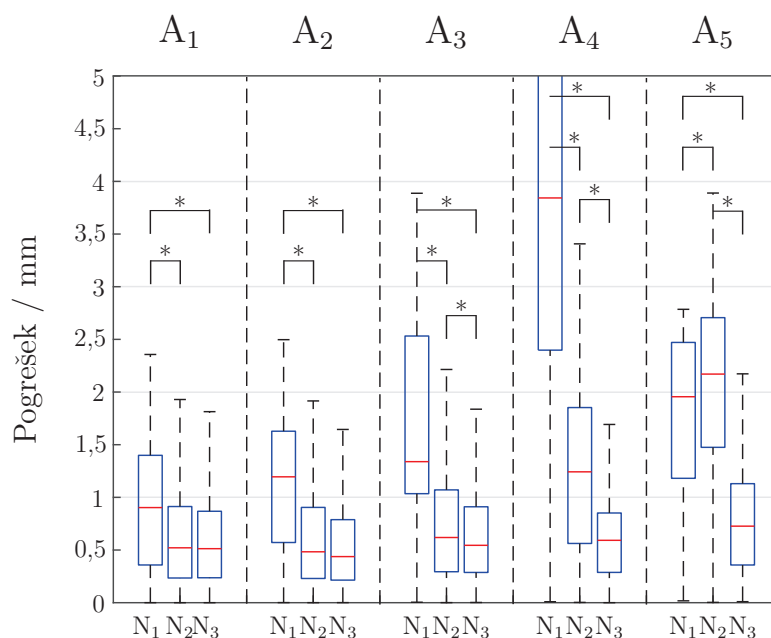
Zapis trajektorije s finimi gibi. Na podlagi teh rezultatov smo se nabor trajektorij s finimi gibi odločili zapisati z $N_1 = 10$, $N_2 = 20$ in $N_3 = 60$ komponentami. Iz slike 4.10 je razvidno, da, čeprav rezultirajoča trajektorija na podlagi modela z N_1 komponentami zadovoljivo opiše potek v sklopu segmentov A_1 in A_2 , se opis, s prehodom v fine gibe ($A_3 - A_5$) močno poslabša. Potek rezultirajočih trajektorij je tako pri zapisu z N_1 komponentami, za razliko od trajektorije brez finih gibov, neprimeren. Opis izboljšamo z večanjem števila komponent, a je, tudi ob uporabi N_3 komponent, opis segmenta A_5 nepopoln. To je razvidno iz prikaza izračunanih pogreškov, ki smo



Slika 4.10: Prikaz zapisov trajektorije s finimi gibi pri uporabi metode GMM z N komponentami.

jih na sliki 4.11 prikazali z uporabo škatličnih diagramov. Razvidno je, da se v primerih N_1 in N_2 raztros in mediana pogreškov močno povečata v sklopu segmentov A_4 in

A_5 , kar potrjuje slab zapis finih gibov. S številom komponent N_3 je raztros pogreškov vzdolž segmentov konstanten, kar nakazuje boljši zapis, a sta tako mediana kot tudi maksimalna vrednost, kljub visokemu številu komponent, relativno visoki. To nas na-



Slika 4.11: Prikaz generiranih pogreškov v primeru trajektorije s finimi gibi vzdolž posameznih segmentov ($A_1 - A_5$) pri uporabi metode GMM z N komponentami.

pelje na možnost dodatnega povečevanja števila komponent, a se z večanjem števila komponent veča tudi računaska kompleksnost in posledično podaljšuje čas učenja. Model je v primeru N_1 komponent tako potreboval za učenje 31 sekund, v primeru N_2 komponent 155 sekund, v primeru N_3 komponent pa 571 sekund (procesor Intel(R) Core(TM) i5-3570K 3.40 GHz in 8 GB pomnilnika).

Ker dolgi časi učenja niso želeni, se poraja vprašanje ali je možno ustrezno zapisati fine gibe brez povečevanja števila komponent, ki bi povzročili daljše čase učenja. Tako smo v sklopu študije razvili metodo, ki prilagodi začetno porazdelitev komponent vzdolž trajektorije tako, da je gostota komponent višja na območjih finih gibov.

4.5.3 Nadgradnja metode GMM s prilagajanjem gostote komponent na podlagi frekvenčne analize

V primeru metode GMM referenčno trajektorijo zapišemo v obliki verjetnostnega opisa podatkov z uporabo N komponent. Ker pa je za ustrezen opis finih gibov potrebno

uporabiti veliko število komponent, se računaska kompleksnost močno poveča, kar naredi metodo GMM manj primerno za zapis finih gibov.

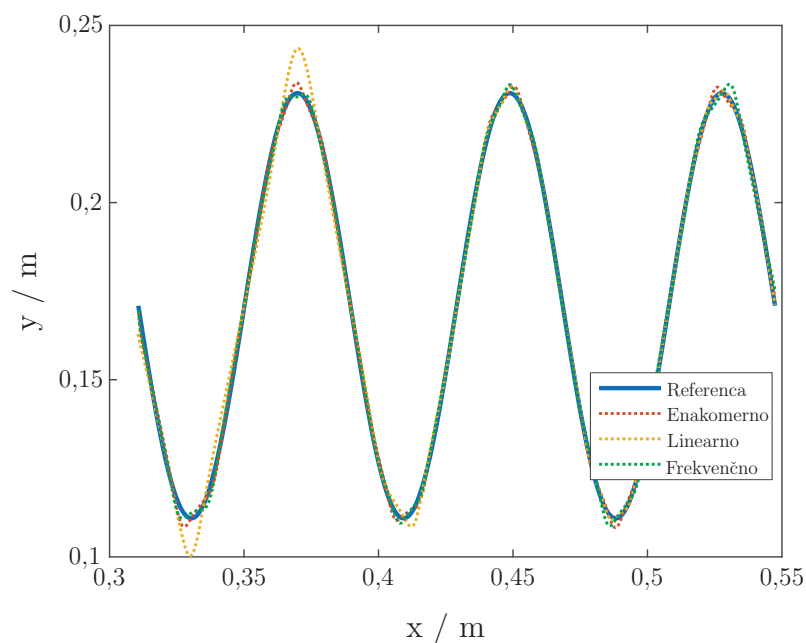
Temu problemu pa se lahko izognemo, če namesto obsežnega povečevanja števila komponent, te le drugače razporedimo vzdolž neodvisne spremenljivke podatkov (npr. normiran čas). Navadno je začetna razporeditev komponent enakomerna, kar pa za opis finih gibov ni optimalno. S predlagano metodo določimo začetno postavitev komponent na podlagi frekvenčne analize podatkov. Da pa smo preverili delovanje metode, smo primerjali zapis z metodo GMM pri enakomerni razporeditvi komponent, pri razporeditvi komponent na podlagi frekvenčne analize ter pri linearno naraščajoči porazdelitvi komponent. Linearno naraščajočo porazdelitev komponent smo izbrali, ker predstavlja smiselno porazdelitev komponent za opis trajektorije s finimi gibi (Slika 4.1, a) in nam tako služi za oceno delovanja predstavljene metode.

Pri primerjavi metod smo uporabili za zapis trajektorij $N = 20$ komponent.

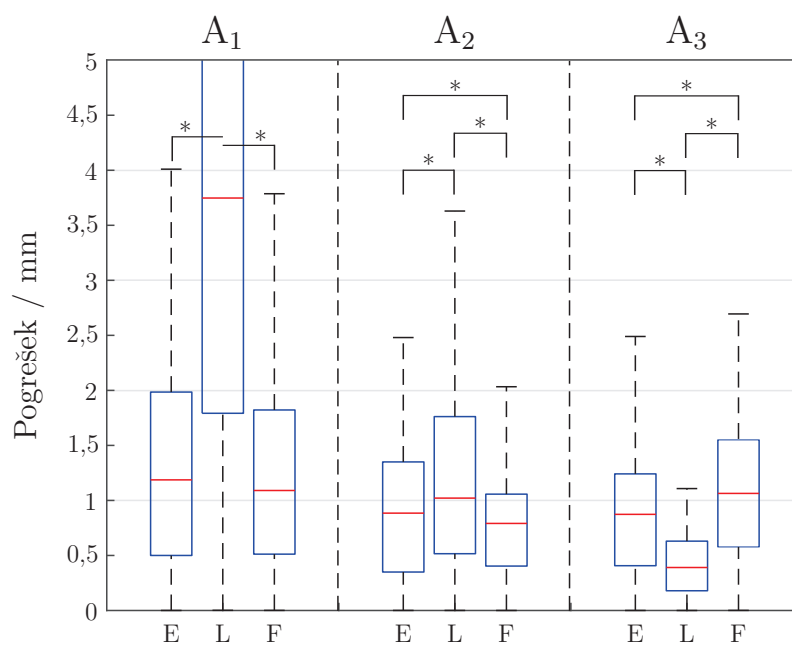
Zapis trajektorije brez finih gibov. V primeru trajektorij brez finih gibov je razvidno, da poteki rezultirajočih trajektorij sledijo poteku referenčne trajektorije (Slika 4.12). Potek je, v primeru porazdelitve komponent z uporabo predstavljene metode, podoben poteku v primeru enakomerne porazdelitve komponent. To je pričakovano saj je dominantna frekvenca vzdolž trajektorije brez finih gibov konstantna, kar rezultira v enakomerni porazdelitvi komponent. V primeru linearno naraščajoče porazdelitve komponent pa so razvidna velika odstopanja vzdolž segmenta A_1 , ki postajajo vedno manjša in manj opazna vzdolž segmentov A_2 in A_3 . To je posledica večje gostote komponent vzdolž segmentov A_2 in A_3 na račun manjše gostote komponent vzdolž segmenta A_1 .

Dodaten vpogled v vpliv porazdelitev komponent na odstopanja od reference nam podajo škatlični diagrami na sliki 4.13. Razvidno je, da sta vzdolž celotne trajektorije, tako mediana kot tudi maksimalna vrednost pogreškov podobni v primeru enakomerne porazdelitve in porazdelitve z uporabo predstavljene metode. To je zeleno, saj je v primeru trajektorije brez finih gibov, enakomerna porazdelitev komponent primerna. Dodatno je razvidno, da so v primeru linearno naraščajoče porazdelitve komponent, pogreški v segmentu A_1 občutno višji od pogreškov z uporabo ostalih dveh metod. Razlika med pristopi je manjša v sklopu segmenta A_2 , pri segmentu A_3 , pa je pogrešek celo manjši kot v primeru ostalih dveh pristopov porazdelitev. To je pričakovano, saj je gostota komponent, v primeru linearno naraščajoče porazdelitve občutno večja v

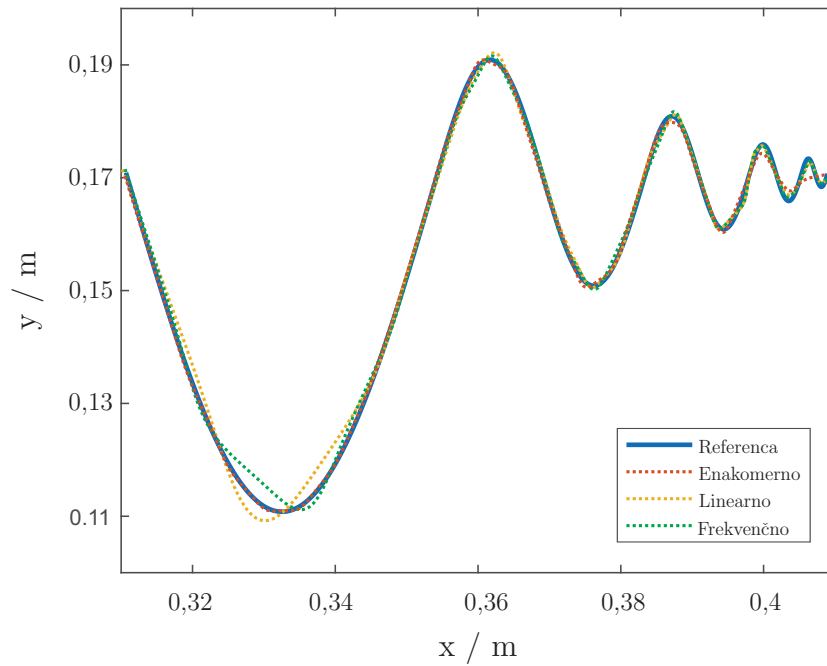
sklopu segmenta A_3 .



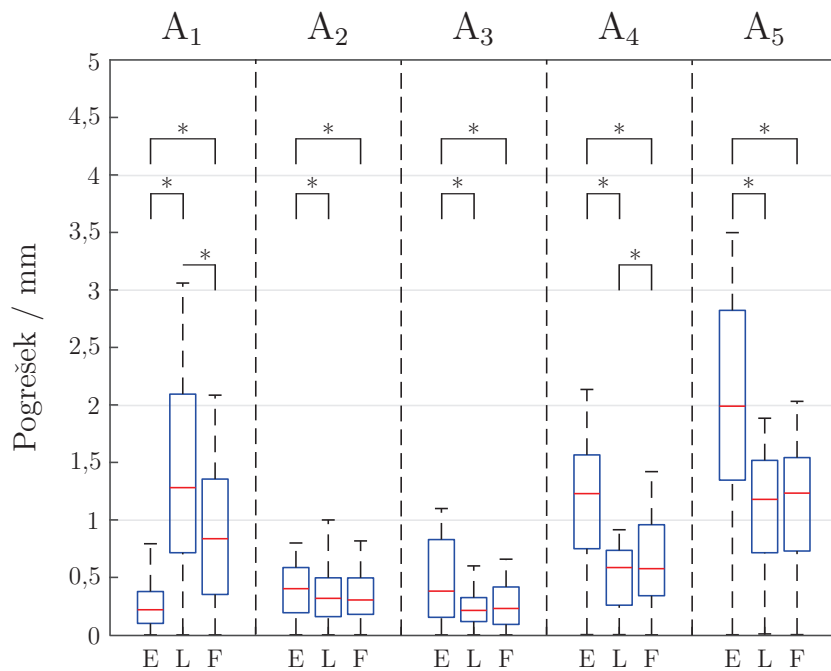
Slika 4.12: Prikaz zapisov trajektorije brez finih gibov pri uporabi metode GMM z $N = 20$ komponentami ob različnih pristopih razporeditve komponent vzdolž podatkov.



Slika 4.13: Prikaz generiranih pogreškov v primeru trajektorije brez finih gibov vzdolž posameznih segmentov (A_1 , A_2 , A_3) pri uporabi metode GMM z enakomerno (E), linearno naraščajočo (L) ali frekvenčno (F) razporeditvijo komponent.



Slika 4.14: Prikaz zapisov trajektorije brez finih gibov pri uporabi metode GMM z N komponentami.



Slika 4.15: Prikaz generiranih pogoškov v primeru trajektorije s finimi gibi vzdolž posameznih segmentov ($A_1 - A_5$) pri uporabi metode GMM z enakomerno (E), linearno naraščajočo (L) ali frekvenčno (F) razporeditvijo komponent.

Zapis trajektorije s finimi gibi. V primeru trajektorij s finimi gibi so iz potekov rezultirajočih trajektorij manj razvidne razlike med posameznimi pristopi porazdelitve komponent (Slika 4.14). Razvidno je, da v sklopu segmenta A_1 prihaja do opaznih odstopanj v primeru linearno naraščajoče porazdelitve komponent in porazdelitve na podlagi predstavljene metode. V sklopu segmentov A_4 in A_5 pa prihaja do opaznih odstopanj v primeru enakomerne porazdelitve. To je pričakovano, saj se frekvenčni potek trajektorije spreminja, porazdelitev komponent se temu ustrezno prilagodi, izračunana odstopanja pa so tako bolj podobna pristopu linearno naraščajoče kot pa enakomerne porazdelitve.

Podobnost med odstopanji pri uporabi metode z frekvenčno razporeditvijo komponent in metodo z linearno naraščajočo gostoto komponent dodatno prikazuje slika 4.15. Razvidno je, da je mediana in maksimalna vrednost pogreškov v sklopu segmenta A_1 manjša pri uporabi enakomerne porazdelitve komponent, a se ta razlika občutno zmanjša v sklopu segmentov A_2 in A_3 . V sklopu segmentov A_4 in A_5 pa je razvidno, da so pogreški večji pri uporabi enakomerne porazdelitve komponent, kar je posledica premajhne gostote komponent. Dodatno je razvidno, da so pogreški v primeru porazdelitve komponent z uporabo predstavljene metode podobni kot v primeru linearno naraščajoče porazdelitve komponent.

Tako je kvaliteta zapisa trajektorije v primeru predstavljene metode na območju finega giba primerljiva s kvaliteto zapisa z enakomerno porazdelitvijo komponent v primeru uporabe $N_3 = 60$ komponent (Slika 4.11). Potreben čas izračunavanja je v primeru $N_2 = 20$ znašal 155 s, medtem ko je v primeru N_3 znašal 571 s. Uporaba predstavljene metode podaljša čas izračunavanja za manj kot sekundo. Tako smo za zapis z N_2 komponentami potrebovali približno 156 s, kar je v primerjavi z uporabo enakomerne porazdelitve N_3 komponent hitrejše za 415 s oziroma za 73 %.

4.6 Diskusija

V sklopu študije smo analizirali primernost metod DMP in GMM za zapis finih gibov. Kljub temu da sta metodi po principu delovanja med seboj različni, za obe velja, da je kvaliteta zapisa odvisna od števila izbranih parametrov zapisa (tj. baznih funkcij v primeru metode DMP in komponent v primeru metode GMM). O absolutnih vrednostih optimalnega števila parametrov je težko govoriti, saj je ta vrednost odvisna od trajektorije in danih zahtev opisa. Za opisovanje finih gibov si namreč želimo opisa

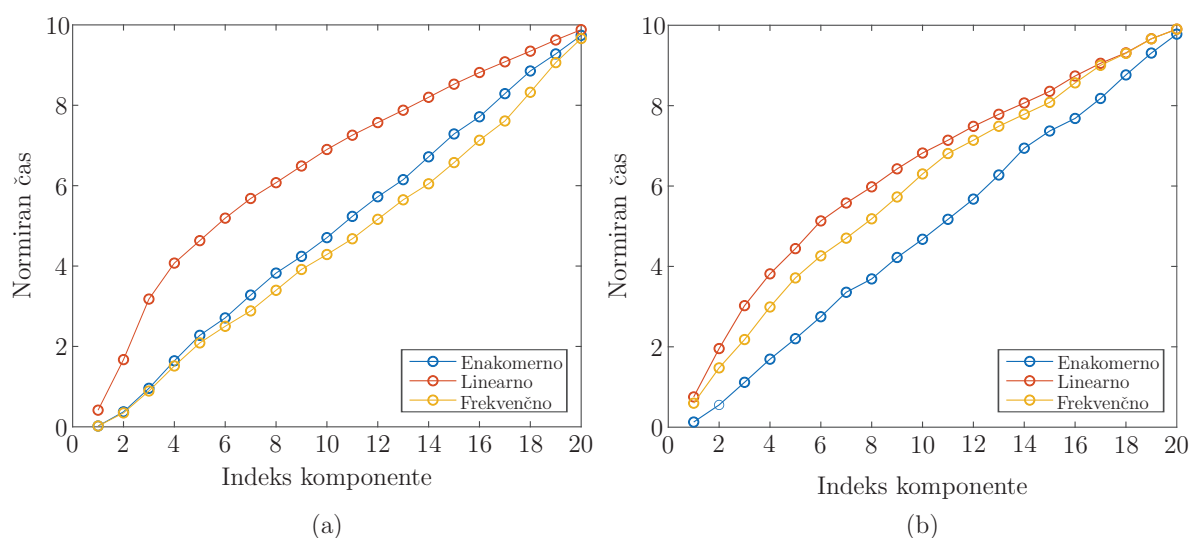
finega dela trajektorije z majhnimi pozicijskimi odstopanji, zaradi česar so zahteve po številu izbranih parametrov drugačne kot če želimo opisati zgolj nek splošen potek.

Tako v primeru metode DMP kot GMM smo generirane pogreške od referenčnih trajektorij zmanjšali s povečevanjem števila parametrov. S povečevanjem števila parametrov se je opis izboljševal, a je bilo za ustrezen opis trajektorije s finimi gibi potrebno uporabiti mnogo več parametrov kot v primeru trajektorije brez finih gibov. V primeru metode DMP je izračunavanje računsko nezahtevno in dodatno povečevanje baznih funkcij ne vpliva na čas potreben za zapis trajektorije. Tako lahko v primeru metode DMP za opis finih gibov smiselno povečevanje števila baznih funkcij, a se moramo ob tem zavedati, da s tem poleg boljšega opisa finih delov trajektorije izboljšamo tudi opis potencialno neželenih potekov znotraj trajektorije, ki so lahko posledica nepopolne demonstracije (npr. vpliv tresavice). Nasprotno velja za metodo GMM, kjer se računski zahtevnost izrazito povečuje z večanjem števila komponent. Podobno kot pri metodi DMP se opis z večanjem števila komponent izboljšuje, a je smiselnost povečevanja zaradi povečane računski kompleksnosti bolj vprašljiva kot v primeru metode DMP.

Tako smo v sklopu študije predlagali tudi novo metodo, s katero smo želeli izboljšati opis finih gibov, brez občutnega povečevanja računski kompleksnosti. Predstavljena je bila metoda, ki prilagodi porazdelitev izbranega števila komponent na podlagi frekvenčne analize. Metoda identificira območja finega giba in tam poveča gostoto komponent. Tako v primeru trajektorije brez finih gibov metoda ohrani porazdelitev komponent podobno enakomerni, medtem ko v primeru trajektorije s finimi gibi, metoda porazdelitev prilagodi. To delovanje je razvidno tudi iz slike 4.16.

V primeru referenčne trajektorije brez finih gibov je razvidno, da je porazdelitev komponent podobna enakomerni porazdelitvi, kar je ustrezno, saj trajektorija ne vsebuje območij s finimi gibi in posledično spreminjanje gostote komponent ni potrebno. Nasprotno je v primeru referenčne trajektorije s finimi gibi porazdelitev komponent dosti bolj podobna linearno naraščajoči porazdelitvi, kar je ustrezno, saj trajektorija vsebuje območje finega giba v sklopu segmentov A_4 in A_5 . Gostota komponent se poveča na račun območij trajektorije, ki ne predstavljajo finega giba. To pomeni, da je tam kvaliteta opisa slabša, kar pa nas ne moti, saj je kvaliteta opisa tega dela manj pomembna.

Na podlagi predstavljenih rezultatov lahko zaključimo, da je predstavljena nadgradnja metode GMM primerna za zapis trajektorij s finimi gibi. Z uporabo predstavljene



Slika 4.16: Prikaz porazdelitve komponent vzdolž podatkov glede na izbran pristop razporeditve komponent. Prikaz je ločen na porazdelitev komponent v primeru trajektorije brez finih gibov (a) in porazdelitev komponent v primeru trajektorije s finimi gibi (b).

metode je namreč opis finih gibov mnogo boljši kot v primeru metode z enakomerno porazdelitvijo komponent, hkrati pa dosega primerljive opise potekov v primeru trajektorij brez finih gibov. Z uporabo te metode lahko izboljšamo kvaliteto zapisa finih gibov, brez dodatnega povečevanja števila komponent.

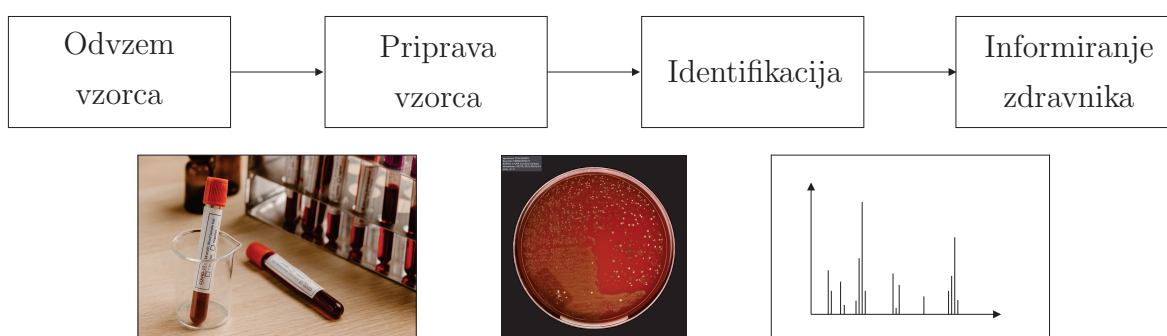
Predstavljeni rezultati nakazujejo želeno delovanje metode, v prihodnje pa bi študijo lahko dopolnili z analizo računske zahtevnosti metode in z analizo kvalitete zapisa na naboru zajetih demonstracij.

5 Študija kinestetičnega učenja finih robotskih gibov z uporabo učnega vmesnika v sklopu laboratorijske aplikacije

RAZVOJ na področju sodelujočih robotov predstavlja nove priložnosti za avtomatizacijo laboratorijev. Sodelujoči roboti so zasnovani za varno uporabo v bližini operaterjev, saj njihovi vgrajeni senzorji sile zaznajo morebitne trke in omogočajo kinestetično učenje gibov. Tako je možno te robote vključiti v laboratorijsko okolje, saj pristopi učenja z demonstracijo omogočajo, da lahko operaterji naučijo robota opravil brez potrebe po znanju programiranja. Vendar pa v laboratoriju veliko nalog zahteva fine gibe, pri katerih je zahtevana submilimetrska natančnost. V že predstavljenih študijah smo pokazali, da je mogoče demonstrirati fine gibe v ravnini z uporabo kinestetičnega učenja (poglavje 3), demonstrirane gibe pa zapisati z uporabo ustreznih metod (poglavje 4). Ugotovili smo tudi, da lahko natančnost demonstracije dodatno izboljšamo z uporabo funkcije vizualnega skaliranja. V sklopu te študije želimo nadalje preveriti primernost uporabe kinestetičnega učenja in sodelujočih robotov v sklopu laboratorijske aplikacije. Kot primer smo si izbrali proces odzema in nanosa kolonije bakterij z namenom njihove identifikacije z uporabo masne spektrometrije (angl. mass spectrometry - MS).

Klinični mikrobiološki laboratoriji imajo ključno vlogo pri preprečevanju, diagnosticiranju in zdravljenju nalezljivih bolezni, ki jih povzročajo mikroorganizmi (npr. bakterije, glive). Za namen identifikacije organizma mikrobiologi pogosto gojijo klinične vzorce na trdih gojiščih v petrijevkah ter jih nato odvzamejo in pripravijo za izbrano metodo identifikacije. V zadnjih dveh desetletjih je masna spektrometrija postala po-

gosto uporabljena metoda identifikacije bakterij in gliv, saj je natančna, hitrejša in manj delovno potratna v primerjavi z drugimi metodami [20]. Metoda ionizacije v nosilcu z lasersko desorpcijo (angl. matrix-assisted laser desorption/ionization - MALDI) z uporabo časa preleta (angl. time-of-flight - TOF) se je zaradi visoke učinkovitosti delovnega postopka (tj. kratekega časa obdelave in zmanjšanih stroškov na identifikacijo) uveljavila kot ena izmed glavnih metod MS [41, 57]. Med drugim se namreč kolonije mikroorganizmov neposredno nanese na tarčno ploščo ter prekrije z matrico brez potrebe po dodatni manipulaciji vzorcev.



Slika 5.1: Prikaz poteka identifikacije organizmov z uporabo MALDI-TOF postopka v sklopu klinične mikrobiologije.

Odvzem in nanos kolonij se pogosto izvaja ročno z uporabo inokulacijskih zank, zobotrebcev ali pipetirnih konic [58, 59]. Ker pa se potencialne napake pokažejo šele veliko pozneje, so za pravilen odvzem ponavadi potrebni izkušeni tehniki. Da bi se zmanjšalo breme tehnikov, so bili razviti različni avtomatizirani sistemi priprave in identifikacije vzorcev [60]. Ti so sposobni obdelave večjega števila vzorcev in zmanjšujejo stopnjo napak [61, 62]. Čeprav je bilo v preteklosti predlaganih več primerov uporabe robotov za odvzem kolonij, se je večina namesto nanosa na tarčno ploščo osredotočila na odlaganje vzorcev v vdolbinice mikrotitrne plošče, kar ni primerno za MALDI-TOF proces [63–66]. Nekateri avtomatizirani sistemi so dragi, zavzamejo veliko prostora in zagotavljajo visoko hitrost obdelave vzorcev, ki je večina laboratorijev morda ne potrebuje [60]. Celo namizni modeli, ki so cenejši in manjši, imajo to slabost, da so zasnovani le za specifično nalogo (npr. odvzem in nanos kolonij) [62]. Sodelujoči roboti po drugi strani omogočajo prilagodljivost različnim nalogam, saj je potrebno zgolj zamenjati orodje v vrhu robota, demonstrirati novo nalogo in robot bo v teoriji zmožen opravljati to nalogo. Zaradi vsestranskosti so tako sodelujoči roboti zanimivi za uporabo v kliničnih mikrobioloških laboratorijih.

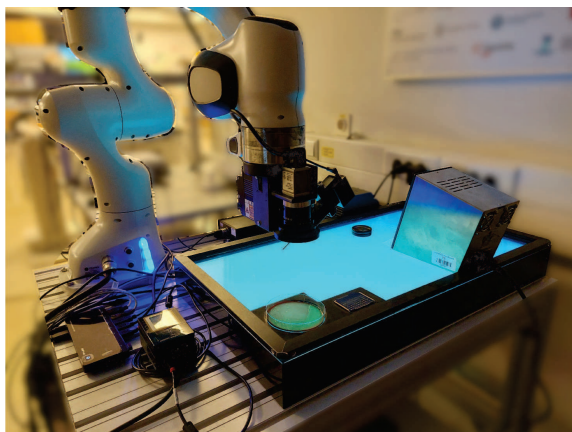
Da pa preverimo ali je implementacija sodelujočih robotov izvedljiva tudi v praksi, smo v sklopu te študije robota postavili v proces identifikacije kolonij bakterij (Slika 5.1). Tako smo razvili specializirano orodje, pričvrščeno na vrh robotskega manipulatorja, ki je omogočal natančno detekcijo kolonij, njihov odvzem in nanos. Razvit je bil tudi t.i. učni vmesnik, ki je omogočal operaterju učenje zelenih gibov. Združeval je sistem obogatene resničnosti, ki je omogočal vizualno skaliran prikaz delovnega območja, kot tudi sistem zapisa demonstriranih trajektorij, ki je temeljil na metodi DMP.

5.1 Eksperimentalni sistem

Eksperimentalni sistem sestavljajo robotski manipulator Panda (Franka Emika, GmbH), masni spektrometer MALDI Biotyper (Bruker, Inc.), toplotni sterilizator SteriMax smart (WLD-TEC, GmbH) in tarčne plošče MALDI MBT Biotarget 96 (Bruker, Inc.). Specializirano orodje, ki je pritrjeno na vrh robotskega manipulatorja sestavljajo kamera RGB a2A3840-45ucPRO (Basler, GmbH), linijski laserski merilnik ZG2-WDS8T (Omron Industrial Automation, Inc.), senzor sile Nano17 (ATI Industrial Automation, Inc.), 360° servomotor s povratno zanko (Parallax, Inc.) in igla iz nerjavečega jekla (Slika 5.2, b). Za namene obogatene prikaza delovnega okolja smo uporabili očala za navidezno resničnost Oculus Quest (Meta, Inc.).

Robot Panda v primeru kinestetičnega učenja služi tako kot upravljalna kot tudi izvršna naprava. Z uporabo RGB kamere in linijskega laserskega merilnika smo zaznali kolonijo bakterij in določili njen položaj s submilimetrsko natančnostjo. Z RGB kamero smo pridobili sliko petrijevke (Slika 5.3, a), na podlagi katere je operater lahko določil približni položaj izbrane bakterijske kolonije. Območje okoli izbranega položaja je bilo nato skenirano z linijskim laserskim merilnikom, ki je operaterju omogočil podroben 3D model območja vključno z bakterijsko kolonijo (Slika 5.3, b). Velikost skeniranega območja je bila $8\text{ mm} \times 4\text{ mm}$. Ločljivost linijskega merilnika znaša vzdolž horizontalne osi x $13\text{ }\mu\text{m}$ ter vzdolž vertikalne osi z $1\text{ }\mu\text{m}$. Ločljivost ustvarjenega 3D modela vzdolž horizontalne osi y , določena z gibanjem robota, je znašala $40\text{ }\mu\text{m}$. Ustvarjen 3D model je operaterju zagotovil natančen položaj kolonije. Kolonije so bile odvzete in nanešene z iglo iz nerjavečega jekla in premera 1 mm (Slika 5.3, c). Špico igle smo oblikovali z brušenjem, tako da je bil končni premer $0,5\text{ mm}$, špica pa je bila dolga $1,5\text{ mm}$. Da smo odvzete kolonije analizirali z MALDI Biotyper masnim spektrometerom, smo jih morali nanesti na tarčno ploščo MBT Biotarget 96 (Slika 5.3, d). Zaradi togosti tarčne plošče

so bile kontaktne sile relativno velike, kar je onemogočalo primeren nanos. Zato smo uporabili senzor sile, na podlagi katerega smo prilagodili trajektorijo nanosa vzdolž vertikalne osi z . Za ustrezen nanos je bilo potrebno iglo tudi vrteti. To smo dosegli z uporabo 360° servomotorja s povratno informacijo o legi. Po vsakem nanosu je bila igla sterilizirana s 96 % denaturiranim etilnim alkoholom in toplotnim sterilizatorjem SteriMax smart.



(a)



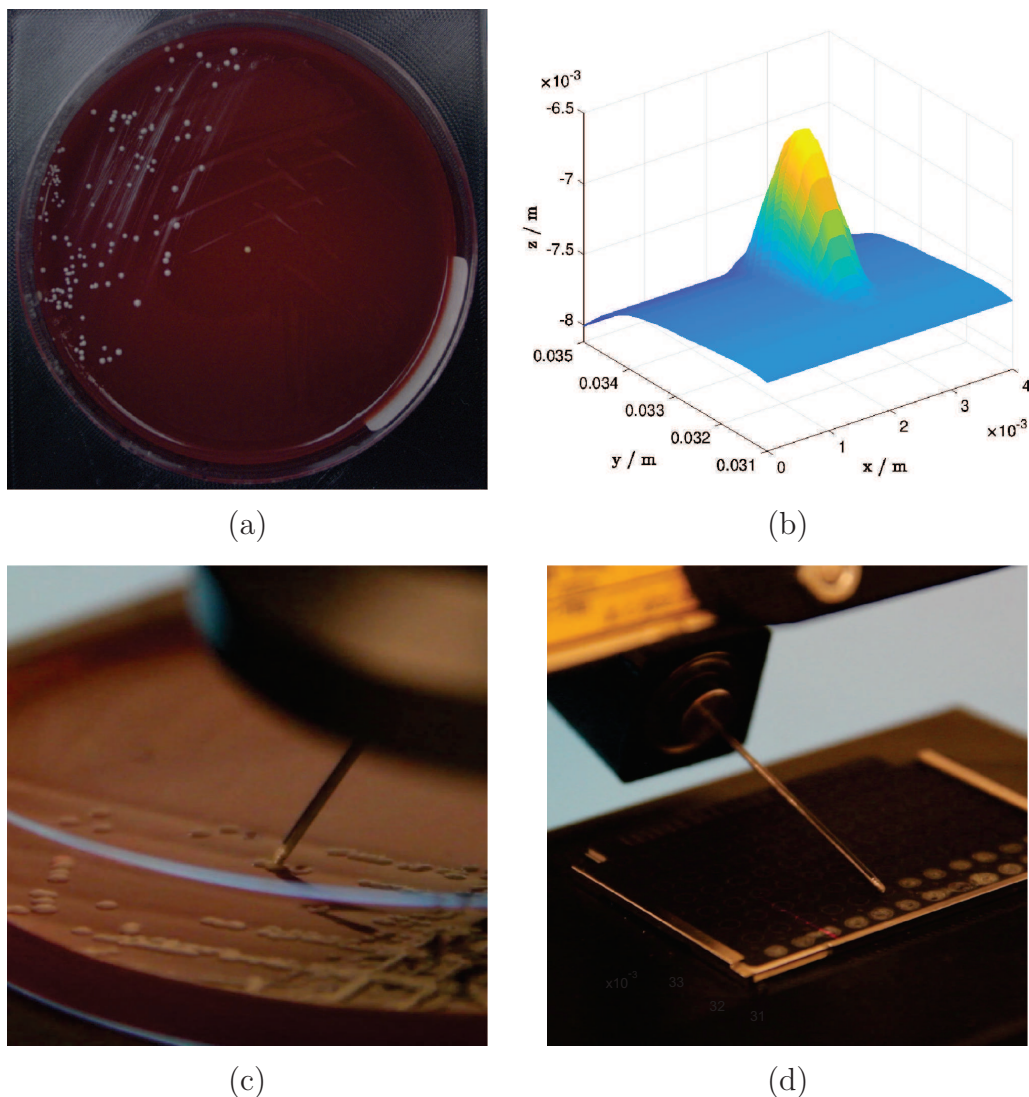
(b)

Slika 5.2: Prikaz eksperimentalnega sistema (a) in orodja na vrhu manipulatorja (b).

Visokonivojsko odločanje je bilo realizirano s programom MATLAB Simulink 2019b (The MathWorks, Inc.). V tem okolju so se združevale informacije iz posameznih senzorjev in robotskega manipulatorja, poleg tega pa je bil uporabljen za definiranje poteka eksperimenta in shranjevanje izmerjenih podatkov iz robota Panda (položaj in orientacija vrha robota, položaj v sklepih, hitrost sklepov, itd.), senzorja sile Nano17 (vektorji sil in navora), kamere RGB (pridobljena slika) in linijskega laserskega merilnika (3D model). Uporabljen je bil tudi za zapis trajektorij z metodo DMP. Za razvoj okolja navidezne resničnosti smo uporabili igralni pogon Unity (Unity Technologies, Inc.).

5.2 Učni vmesnik

Učenje gibov z demonstracijo je težavno, saj je poleg izvedbe demonstracije potrebno generirane trajektorije tudi posplošiti ter prilagajati glede na zahteve aplikacije. Z implementacijo učnega vmesnika želimo operaterja razbremeniti nepotrebnih in zahtevnih opravil z namenom, da je uporaba robota v laboratoriju enostavna do te mere, da operater zgolj izvede demonstracijo, za vse ostalo pa poskrbi vmesnik.



Slika 5.3: Prikaz poteka eksperimenta - slika petrijevke zajete z RGB kamero (a), 3D model izbrane bakterijske kolonije generiran z uporabo linijskega laserskega merilnika (b), prikaz procesa odvzema kolonije (c) in prikaz procesa nanosa kolonije (d).

V našem primeru učni vmesnik sestavljata dva ključna sistema - sistem obogatene resničnosti kot pomoč med izvajanjem demonstracij in sistem zapisa demonstracij z uporabo metode DMP. Tako lahko na učni vmesnik gledamo kot na neke vrste digitalni dvojček, ki omogoča kreiranje virtualnega okolja, ki povzema trenutno stanje realnega procesa in se ga lahko prikazuje z uporabo obogatene resničnosti. Poleg kreiranja virtualnega okolja pa omogoča tudi nadaljnje prilagajanje procesa znotraj virtualnega okolja, saj lahko z uporabo metode DMP prilagodi potek demonstriranega giba, kar se posledično odraža tudi na realnem procesu [67, 68].

5.2.1 Obogatena resničnost kot pomoč pri izvajanju demonstracij

V primeru učenja gibov za namene odvzema kolonij bakterij iz petrijevke operater želi gib demonstrirati tako, da bo kolonija uspešno odvzeta. Kvaliteto demonstracije mu otežuje visoka zahtevana natančnost in majhna dimenzija kolonij, ki otežuje operaterju zaznavo pravilnosti pozicije vrha robota in posledično uspešnosti izvedene demonstracije. Izpostavitve ključnih predmetov, sprememba zornega kota in povečan prikaz delovnega okolja bi lahko izboljšali kvaliteto izvedene demonstracije. Naštete funkcionalnosti lahko operaterju omogočimo z uporabo obogatene resničnosti. Obogatena resničnost (angl. augmented reality - AR) je princip, pri katerem so predmeti, ki se nahajajo v realnem svetu dopolnjeni z računalniško ustvarjenimi zaznavnimi informacijami, ki so lahko podane preko različnih modalitet (npr. vidnih, slušnih). Opredeljujejo jo tri osnovne značilnosti, ki so združevanje realnega in virtualnega sveta, interakcija v realnem času ter natančna 3D postavitve virtualnih in realnih predmetov [69]. Dodana senzorična informacija pa je lahko konstruktiva, pri čemer informacija dopolni realno okolje, ali destruktivna, pri čemer informacija zakrije realno okolje [70]. Tako obogatena resničnost vpliva na trenutno zaznavanje realnega okolja, medtem ko navidezna resničnost popolnoma nadomesti operaterjevo realno okolje s simuliranim [71].

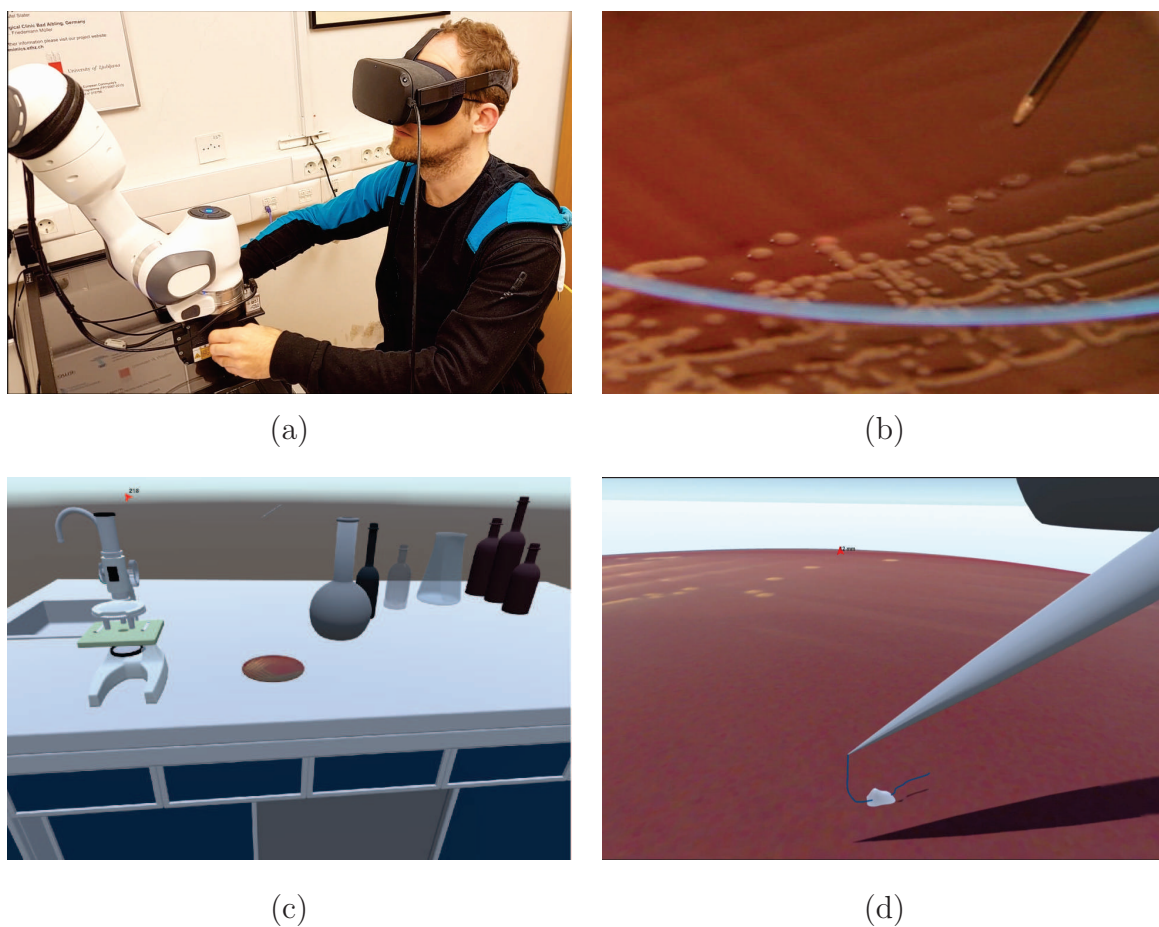
Obogateno okolje smo razvili v okolju Unity in ga sestavlja več predmetov pri čemer sta 3D model referenčne kolonije in igla robotskega manipulatorja pomembna za funkcionalnost aplikacije, ostali pa poskrbijo za ustrezno umeščenost v prostor kar operaterju izboljša občutek za razsežnost prostora (Slika 5.4, d). Poglavitna dodana vrednost obogatene okolja je skaliran prikaz, ki omogoča operaterju boljšo zaznavo odstopanj od zelene lege, kar je bilo ugotovljeno že v sklopu prve študije (poglavje 3). V sklopu okolja obogatene resničnosti smo povečavo spreminjali s premikanjem kamere vzdolž osi med začetno lego kamere in tarčno kolonijo. Tako smo lego kamere posodobili vsak časovni trenutek t

$$\mathbf{p}(t + 1) = \mathbf{p}(t) + \Delta\mathbf{p}(t), \quad (5.1)$$

kjer $\mathbf{p}(t + 1)$ predstavlja novo lego kamere, $\mathbf{p}(t)$ predstavlja trenutno lego kamere in $\Delta\mathbf{p}(t)$ predstavlja diferencialni premik. Tega definiramo kot

$$\Delta\mathbf{p}(t) = K(d) \frac{\mathbf{p}_b - \mathbf{p}_d}{N}, \quad (5.2)$$

pri čemer predstavljata \mathbf{p}_b in \mathbf{p}_d skrajni legi kamere (blizu, daleč), N predstavlja število diferencialnih delov, celoten ulomek pa posledično diferencialni premik. Definiramo



Slika 5.4: Prikaz operaterja pri demonstriranju giba z uporabo obogatene resničnosti (a), bakterijskih kolonij na podlagi katerih je bil generiran 3D model kolonije (b) ter pogled operaterja na obogateno delovno okolje ob oddaljenosti od tarčne kolonije (c) in pogled na obogateno delovno okolje v bližini kolonije (d).

tudi nelinearni skalirni faktor $K(d)$, ki je funkcija razdalje d med trenutno lego igle in tarčno kolonijo

$$K(d) = \pm \frac{d^2}{d_{max}^2 d_{thr}}, \quad (5.3)$$

kjer d_{max} predstavlja normirni člen, d_{thr} pa vpliva na dinamiko nelinearne funkcije. Nelinearnost skalirnega faktorja nam omogoča, da lahko operater izvaja tako večje premike kot tudi fine gibe, saj se izognemo premajhni občutljivosti povečave med izvajanjem večjih premikov in preveliki občutljivosti povečave med izvajanjem finih gibov.

Predznak v enačbi (5.3), ki vpliva na to ali se bo prikaz večal ali manjšal, pa je pogojen s prisotnostjo 3D modela vrha robota v vidnem polju operaterja. To določimo

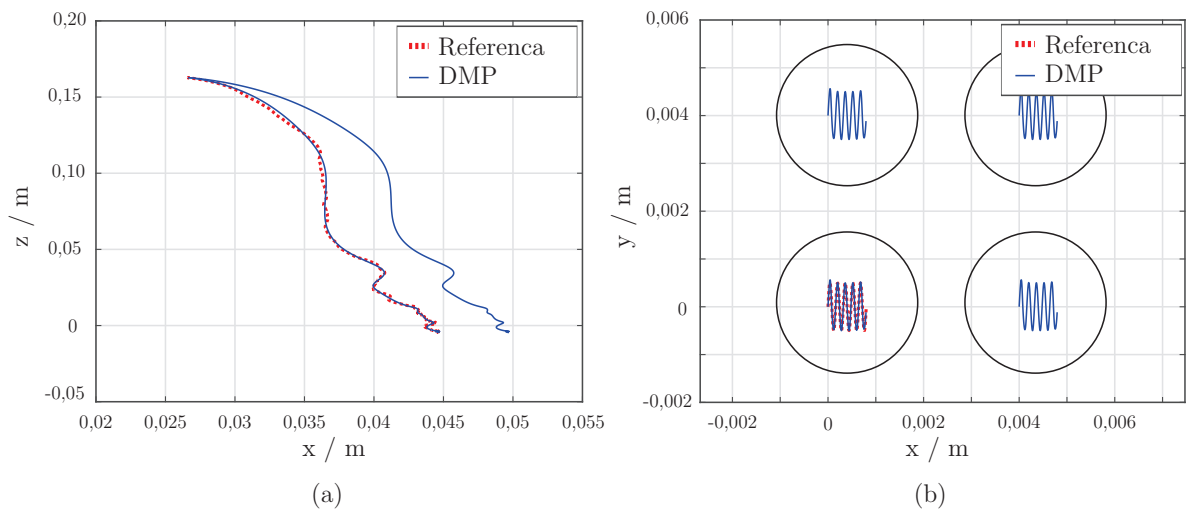
na podlagi kota Ω , ki predstavlja prostorski kot med osjo gibanja kamere in osjo med trenutno pozicijo kamere in trenutno pozicijo modela vrha robota. V kolikor je prostorski kot manjši od 20° , je predznak pozitiven in kamera se približuje, če pa je kot večji od 40° , je predznak negativen in kamera se oddaljuje. V kolikor je kot med 20° in 40° velja $\Delta\mathbf{p}(t) = 0$.

Na podlagi predstavljenega je potreben dodaten razmislek o primernosti uporabe termina obogatena resničnost. Zaradi uporabe očal za navidezno resničnost, ki so namenjene prikazu virtualnega okolja in operaterju popolnoma zakrijejo pogled na delovno okolico bi namreč lahko trdili, da uporabljamo sistem za navidezno resničnost (Slika 5.4). A se moramo zavedati, da je celotno zaznavanje operaterja, v primeru navidezne resničnosti, ustvarjeno na podlagi računalniško ustvarjenih informacij, pri čemer operaterji ne zaznavajo resničnega sveta okoli sebe [72]. To ne velja za predstavljen sistem, saj so operaterji, kljub temu da je bilo vizualno zaznavanje odvisno od računalniško ustvarjenih informacij, vseskozi zaznavali resnični svet z uporabo taktilne in zvočne zaznave. Na podlagi predstavljenih argumentov smo mnenja, da je uporaba termina obogatena resničnost, v sklopu te doktorske disertacije, primerna.

5.2.2 Učenje na podlagi demonstracij

Drugi pomemben sistem učnega vmesnika so metode za zapis izvedenih demonstracij. Razvitih je bilo več metod za namene zapisa demonstracij [6, 17, 18, 54]. Metoda DMP zapiše trajektorijo v obliki nelinearne diferencialne enačbe drugega reda, pri čemer je za zapis potrebna zgolj ena demonstracija [17]. V sklopu druge študije (poglavje 4) smo ugotovili, da je za zapis primerna tako metoda DMP, kot tudi metoda GMM s frekvenčno porazdelitvijo komponent. Za zapis gibov v sklopu aplikacije identifikacije kolonij bakterij pa smo se odločili za uporabo metode DMP. Za uporabo metode DMP smo se odločili, ker zahteva le eno demonstracijo, kar je zaželeno v sklopu laboratorijskega okolja, ko je potrebno gib naučiti v čim krajšem času. Poleg tega pa metoda omogoča prilagajanje začetne in končne pozicije trajektorije, pri čemer se ohranja gladkost trajektorije. V izbrani aplikaciji je to uporabno za prilagajanje končne pozicije pri procesu odvzema kolonije ter za prilagajanje začetne in končne pozicije pri procesu nanosa (Slika 5.5).

Generiranje trajektorije smo za namene aplikacije dodatno nadgradili, saj smo se želeli izogniti prevelikim kontaktnim silam, ki nastanejo med procesom nanosa ob stiku



Slika 5.5: Prikaz zapisa referenčnih trajektorij procesa odvzema (a) in nanosa (b) z uporabo metode DMP.

s togo tarčno ploščo. Sile omejimo s kompenzacijo trajektorije vzdolž osi z na podlagi izmerjenih sil iz sensorja sile, vgrajenega v orodje na vrhu manipulatorja. Trajektorija je kompenzirana na podlagi enačbe

$$\mathbf{p}(t) = \mathbf{p}_r(t) + \mathbf{K} \int \mathbf{f}(t) dt, \quad \mathbf{K} = \begin{bmatrix} 0 \\ 0 \\ K_Z \end{bmatrix}, \quad (5.4)$$

pri čemer $\mathbf{p}(t)$ predstavlja trenutno pozicijo vrha robota, $\mathbf{p}_r(t)$ referenčno trajektorijo na podlagi DMP-ja, $\mathbf{f}(t)$ pa predstavlja sile izmerjene s sensorjem Nano17. \mathbf{K} določa skalirni faktor med izmerjenimi silami in želenim premikom.

Uporaba metode DMP in prilagajanje trajektorije na podlagi izmerjenih sil sta omogočila enostavno učenje ter zadovoljivo generiranje gibov odvzema in nanosa bakterijskih kolonij, kar potrjuje ustrezno izbiro metod učenja za potrebe učnega vmesnika.

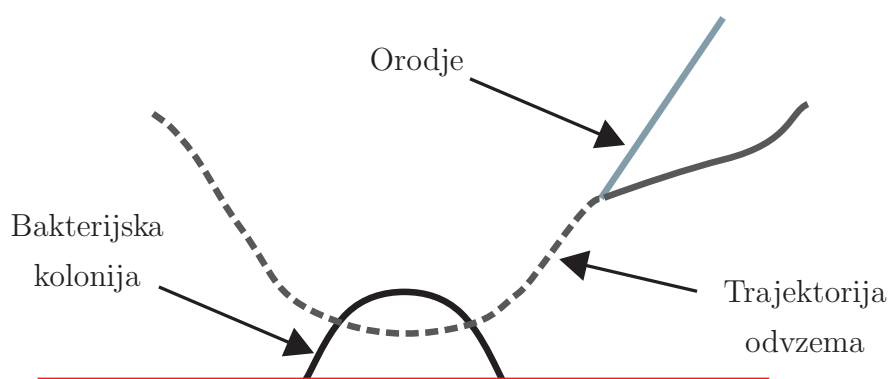
5.3 Identifikacija bakterijskih kolonij v procesu MALDI-TOF

Za prikaz primernosti kinestetičnega učenja finih gibov z uporabo učnega vmesnika v sklopu laboratorijske aplikacije smo izbrali proces identifikacije bakterijskih kolonij s procesom MALDI-TOF masne spektrometrije. Za uspešno identifikacijo je poleg ustrezno naučenih gibov potrebno tudi ustrezno zanavanje kolonij ter prilagajanje naučenih

gibov novim zahtevam. Aplikacija je sestavljena iz večih pomembnih segmentov, zato bomo rezultate predstavili za vsak segment posebj.

5.3.1 Eksperimentalni protokol

V sklopu eksperimenta smo opravili proces identifikacije 56-ih kolonij dveh različnih bakterijskih vrst (gramnegativni *Acinetobacter baumannii* in grampozitivni *Staphylococcus epidermidis*), vzgojenih na krvnem agarju. Na začetku eksperimenta je operater demonstriral zelen gib odvzema kolonije (Slika 5.6), medtem ko je bil gib nanosa ustvarjen programsko. Oba giba sta bila nato zapisana z uporabo metode DMP (Slika 5.5). Operater je nato moral pred vsakim odvzemom iz zajete RGB slike petrijevke (Slika 5.1, a) izbrati kolonijo, ki jo želi identificirati. Ko je bila kolonija izbrana, je sistem generiral 3D model kolonije in njene okolice (Slika 5.1, b) in operater je lahko določil mesto odvzema s submilimetrsko natančnostjo. Proces določanja mesta odvzema bi v splošnem lahko bil avtomatiziran, a smo ga za namene te študije prepustili operaterju. Operater je tudi določil mesto nanosa na tarčni plošči, nato pa je robotski manipulator odvil in nanese izbrano kolonijo (Slika 5.1, c in d). Tekom nanosa se je igla vrha robota vrtela okoli svoje osi, kar je omogočalo optimalen nanos. Za vsak odvzem je bil izveden en nanos. Na koncu so bile nanešene kolonije prekrte z matriksom, po kratkem sušenju pa je bila tarčna plošča vstavljena v masni spektrometer MALDI Biotyper, ki je opravil identifikacijo posameznih nanosov tarčne plošče. Vsaki identifikaciji je bila dodeljena tudi ustrezna ocena, ki je nakazovala zanesljivost identifikacije.



Slika 5.6: Ravninski prikaz želenega poteka trajektorije vrha robota (orodja) pri odvzemu kolonije bakterij.

5.3.2 Rezultati

Rezultati so predstavljeni ločeno za postopek odvzema, nanosa in identifikacije. Medtem ko rezultati identifikacije nakazujejo splošno učinkovitost sistema, pa nam postopka odvzema in nanosa omogočata dodaten vpogled v delovanje ključnih korakov za uspešnost celotnega sistema.

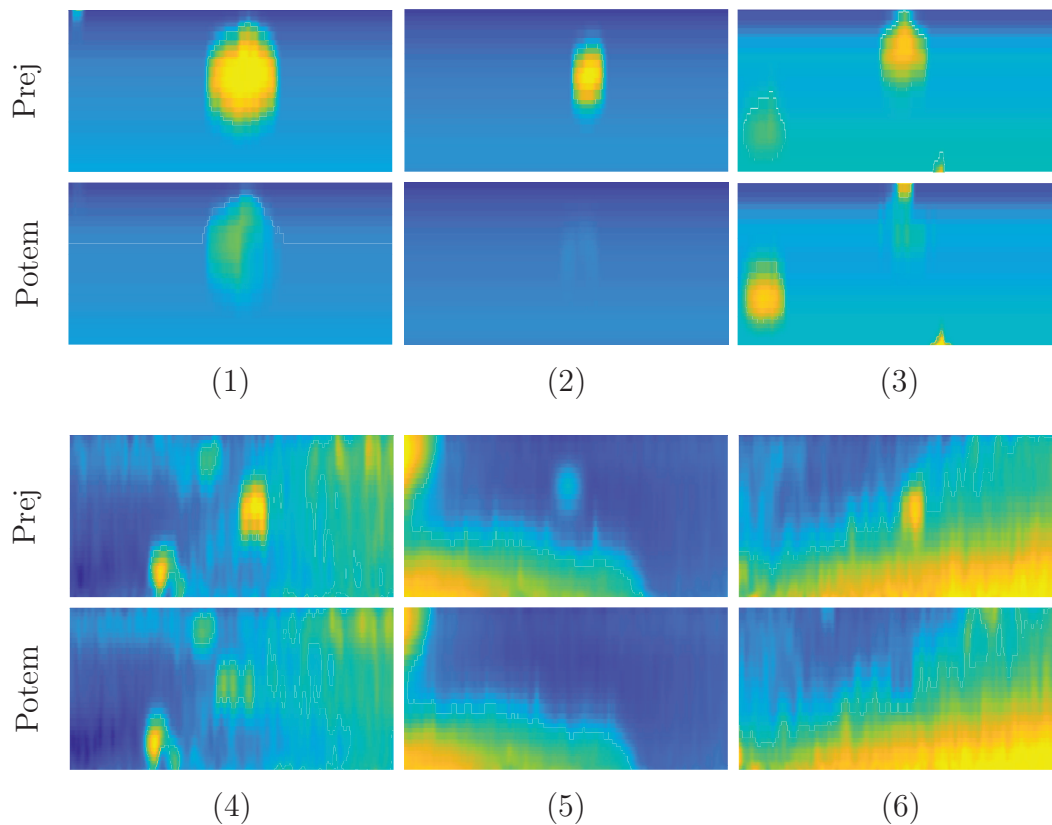
5.3.2.1 Odvzem kolonij

Za oceno postopka odvzema smo primerjali več 3D modelov kolonij pred in po postopku odvzema. Te modele prikazujemo kot *ravninske toplotne prikaze* (Slika 5.7). Medtem ko vsak piksel predstavlja različno koordinato na x - y ravnini 3D modela z dimenzijami $8\text{ mm} \times 4\text{ mm}$, njegova barva predstavlja vrednost koordinate z . Barva se spreminja od modre proti rumeni glede na vrednost, pri čemer temno modra ponazarja najnižje vrednosti z , svetlo rumena pa najvišje vrednosti. Zaradi večje preglednosti slike 5.7 smo izpustili barvne legende, ki bi navajale točne vrednosti z . Menimo namreč, da absolutne vrednosti z niso tako pomembne kot relativna sprememba njihovih vrednosti. Za boljšo predstavo lahko navedemo, da je bila daleč največja kolonija, ki je bila tudi mnogo večja od ostalih (Slika 5.7, 1), dimenzij $1,7\text{ mm} \times 2,1\text{ mm} \times 0,5\text{ mm}$.

Tabela 5.1: Ocenjena sprememba volumna kolonije pri procesu odvzema.

ID	1	2	3	4	5	6
<i>Začetni volumen</i> [mm ³]	1.44	0.78	0.46	0.11	0.06	0.02
<i>Končni volumen</i> [mm ³]	0.41	-0.04	0.04	0.05	0.007	-0.02
<i>Relativna sprememba</i> [%]	-72	-105	-91	-54	-99	-193

Za boljšo oceno postopka odvzema smo izračunali tudi spremembo v prostornini kolonij. Za to smo določili območje, pod katerim je bilo potrebno izračunati prostornino, in spremembo izračunali za 6 različnih kolonij, ki so dobri predstavniki splošne populacije. Na sliki 5.7 in v preglednici 5.1 smo jih razvrstili glede na začetno prostornino.

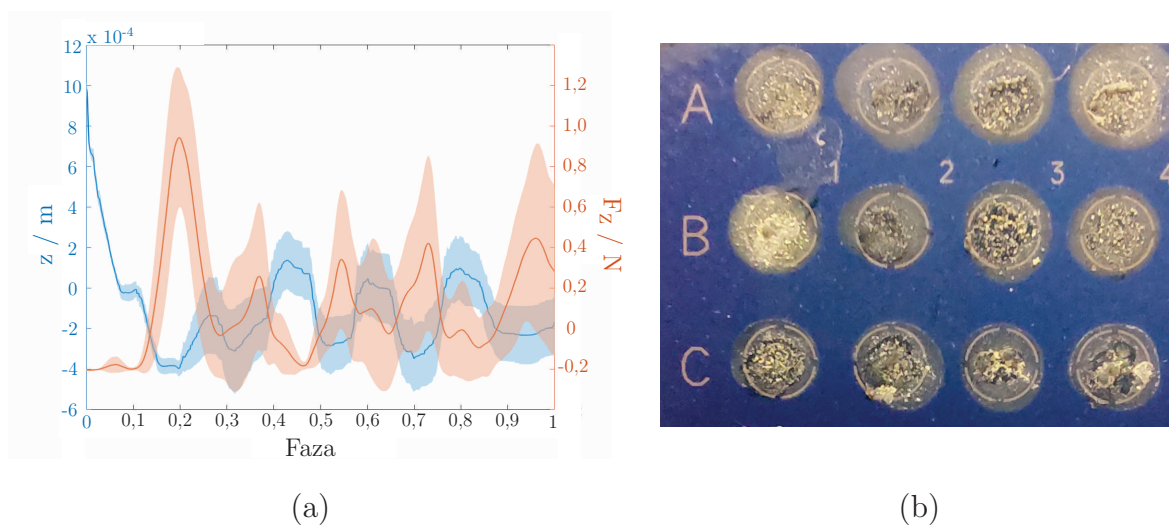


Slika 5.7: *Ravninski toplotni prikaz* 3D modelov kolonij pred in po odvzemu. Kolonije so razvrščene od (1) do (6) glede na začetni volumen.

5.3.2.2 Nanos kolonij

Kakovost nanosa je močno odvisna od kontaktne sile med iglo na vrhu robota in tarčno ploščo. Za omejitev sile je bila vzdolž osi z izvedena kompenzacija pozicije. Na sliki 5.8 je izmerjena kontaktna sila prikazana skupaj z vrednostjo koordinate z igle. Obe vrednosti sta prikazani s kombinacijo srednje vrednosti in standardne deviacije, ki sta bili določeni na podlagi 56-ih nanosov. Povprečna sila redko preseže vrednost 1 N, kar se večinoma zgodi ob začetnem stiku. Po kompenzaciji začetnega stika (faza trajektorije večja od 0,3) znaša povprečna sila $0,29 \pm 0,35$ N. Posledična kompenzacija pozicije znaša približno 0,5 mm.

Končni rezultat nanosa je prikazan na desni fotografiji slike 5.8. Fotografija je bila posneta pred postopkom identifikacije, a po nanosu matriksa.



Slika 5.8: Prikaz kontaktne sile med vrhom robota in tarčno ploščo in posledična kompenzacija pozicije vzdolž osi z (a) in fotografija tarčne plošče pred postopkom identifikacije (b).

5.3.2.3 Identifikacija kolonij

Odvzete in nanešene kolonije smo identificirali z uporabo masnega spektrometra MALDI Biotyper. Vsakemu nanosu je bila ob identifikaciji vrste dodeljena tudi ustrezna ocena, ki je označevala stopnjo zaupanja v identifikacijo. Medtem ko se identifikacije z oceno višjo od 2,0 štejejo za zelo zanesljive, se identifikacija šteje za neveljavno, če je ocena nižja od 1,7 [62].

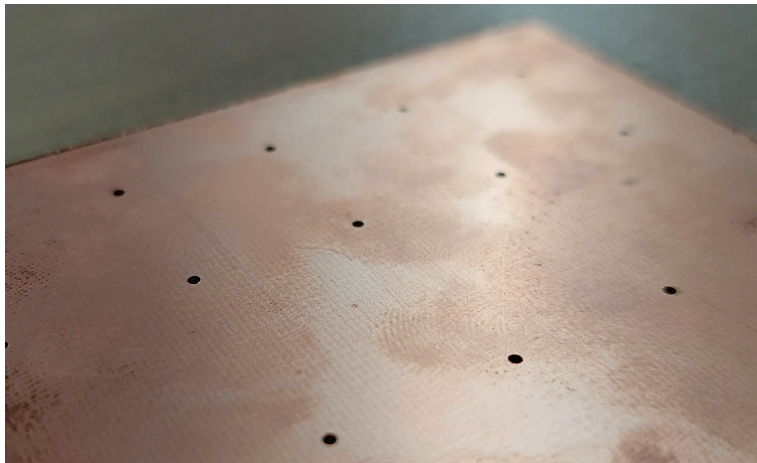
V sklopu našega eksperimenta smo identificirali 56 kolonij. Od teh je bilo uspešno identificiranih 31 kolonij vrste *A. baumannii* in 19 vrste *S. epidermidis*, 6 pa jih je bilo neveljavnih zaradi prenizkih ocen (5 *A. baumannii* in 1 *S. epidermidis*).

Tabela 5.2: Rezultati identifikacije kolonij.

	Acinetobacter baumannii	Staphylococcus epidermidis
Št. vzorcev	36	20
Št. vzorcev z oceno pod 1.70	3	1
Št. vzorcev brez identifikacije	2	0
Št. neveljavnih vzorcev	5	1
Neuspešna identifikacija [%]	13.9	5
Povprečna ocena brez neveljavnih identifikacij	2.05 ± 0.21	2.06 ± 0.13
Povprečna ocena z neveljavnimi identifikacijami	1.94 ± 0.54	2.04 ± 0.16

5.4 Validacija učnega vmesnika

Ločeno od procesa identifikacije bakterijskih kolonij smo validirali tudi predstavljen sistem obogatene resničnosti. Operaterji so morali demonstrirati zahtevan gib z uporabo kinestetičnega učenja, pri čemer so morali izvesti premik do tarčne kolonije, ki se je nahajala na znani poziciji. Ključna zahteva pri izvajanju demonstracije je bila zagotavljanje natančnosti pozicije pri gibu skozi kolonijo. Zaradi nekonsistentnih oblik kolonij in potrebe po pripravi le-teh smo se pri eksperimentu odločili za poenostavitev in kolonijo predstavili kot točo s premerom 1 mm. Te smo rezkali v bakreno ploščo, kar nam je omogočilo natančno poznavanje njihovih leg, njihova globina pa je zaradi senc omogočila dober barvni kontrast za ocenitev x in y pozicije točke (Slika 5.9).



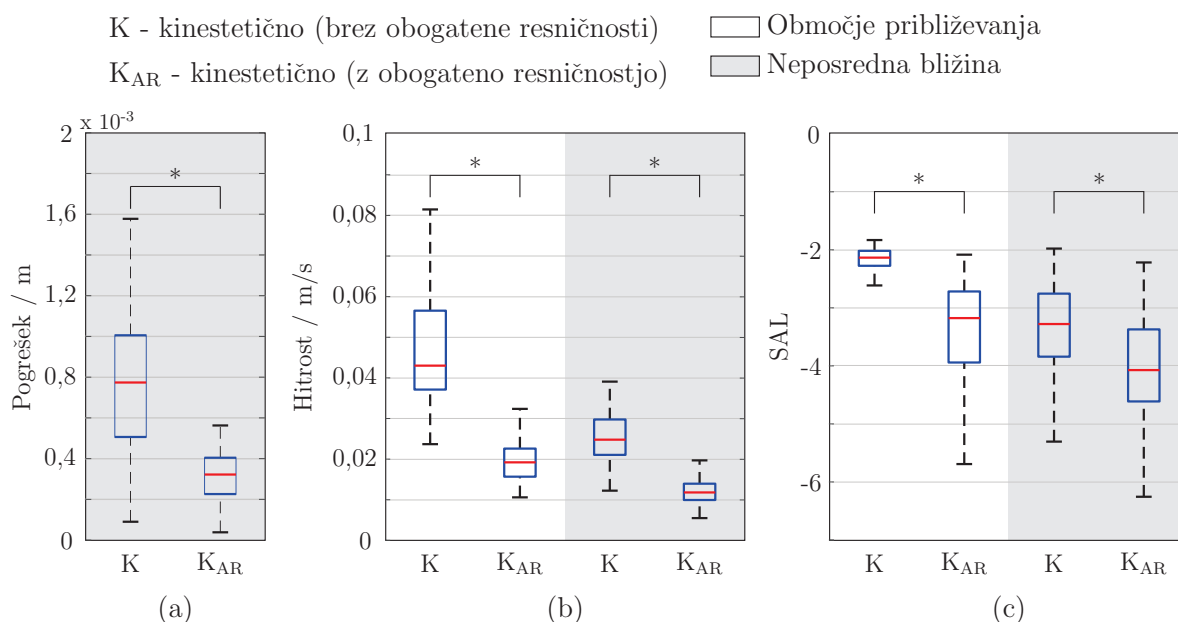
Slika 5.9: Prikaz plošče z referenčnimi točkami.

Validacijo smo opravili na 20-ih operaterjih, ki so morali izvesti gib odvzema kolonije bakterije. Zahtevan gib so morali izvesti na 10-ih različnih referenčnih točkah brez uporabe obogatene resničnosti, nato pa so proces ponovili še z uporabo obogatene resničnosti. Od operaterjev je bila zahtevana natančnost pozicioniranja pri prehodu čez območje kolonije, pri čemer sta bili hitrost in gladkost giba sekundarnega pomena. Na podlagi meritev smo pripravili analizo vpliva obogatene resničnosti na pogrešek od referenčne pozicije, hitrost in gladkost izvedenih gibov (Slika 5.10), pri čemer je bil pogrešek definiran kot

$$\tilde{p} = \min \|\mathbf{p}(t) - \mathbf{p}_r\|, \quad (5.5)$$

kjer $\mathbf{p}(t)$ predstavlja demonstrirano trajektorijo, \mathbf{p}_r pa referenčno točko.

Na sliki 5.10 smo s pomočjo škatličnih diagramov prikazali izračunane pogreške



Slika 5.10: Prikaz vpliva obogatene resničnosti na natančnost pozicioniranja (a), hitrost (b) in gladkost (c) gibanja kinestetičnega učenja. Prikazani so tudi rezultati statistične primerjave, pri čemer * predstavlja statistično signifikantno razliko.

pozicioniranja med kinestetičnim učenjem giba odvzema kolonije. Slika prikazuje vrednosti pogreškov doseženih z in brez uporabe obogatene resničnosti. Poleg tega z uporabo škatličnih diagramov prikazujemo tudi hitrost in gladkost generiranega giba med približevanjem izbrani koloniji in ob neposredni bližini le-te. Smatrali smo, da je operater v neposredni bližini kolonije, ko je vrh robota premaknil znotraj radija 10 mm od izbrane kolonije. Tako grafe hitrosti kot tudi gladkosti gibanja delimo glede na to, ali so operaterji uporabljali obogateno resničnost ali ne.

V primeru vpliva obogatene resničnosti na generirane pogreške je razvidno, da so z uporabo obogatene resničnosti (Med = 0,32 mm; Q1 = 0,23 mm; Q3 = 0,40 mm; Min = 0,04 mm; Maks = 0,56 mm) izračunani pogreški signifikantno manjši kot, če ta ni uporabljena (Med = 0,77 mm; Q1 = 0,51 mm; Q3 = 1,01 mm; Min = 0,10 mm; Maks = 1,58 mm).

Tako z kot brez uporabe obogatene resničnosti se hitrost gibanja zmanjša, ko se vrh robota nahaja v neposredni bližini kolonije. Med približevanjem izbrani koloniji je bila hitrost gibanja brez uporabe obogatene resničnosti (Med = 43,0 mm/s; Q1 = 37,1 mm/s; Q3 = 56,5 mm/s; Min = 23,7 mm/s; Maks = 81,5 mm/s) višja kot z uporabo (Med = 19,2 mm/s; Q1 = 15,7 mm/s; Q3 = 22,6 mm/s; Min = 10,6 mm/s;

Maks = 32,4 mm/s). Prav tako je bila v neposredni bližini kolonije hitrost gibanja brez uporabe obogatene resničnosti (Med = 24,8 mm/s; Q1 = 21,1 mm/s; Q3 = 29,8 mm/s; Min = 12,3 mm/s; Maks = 39,1 mm/s) višja kot z uporabo (Med = 11,8 mm/s; Q1 = 9,9 mm/s; Q3 = 13,9 mm/s; Min = 5,5 mm/s; Maks = 19,7 mm/s).

Podobno velja za gladkost gibanja. Tako z kot brez uporabe obogatene resničnosti se gladkost gibanja zmanjša, ko se vrh robota nahaja v neposredni bližini kolonije. Med približevanjem izbrani koloniji je bila gladkost gibanja brez uporabe obogatene resničnosti (Med = -2,13; Q1 = -2,28; Q3 = -2,02; Min = -2,61; Maks = -1,83) boljša kot z uporabo (Med = -3,18; Q1 = -3,94; Q3 = -2,72; Min = -5,69; Maks = -2,08). Prav tako je bila v neposredni bližini kolonije gladkost gibanja brez uporabe obogatene resničnosti (Med = -3,28; Q1 = -3,84; Q3 = -2,75; Min = -5,30; Maks = -1,98) boljša kot z uporabo (Med = -4,07; Q1 = -4,61; Q3 = -3,37; Min = -6,26; Maks = -2,22).

Iz predstavljenih rezultatov je tako razvidno, da uporaba obogatene resničnosti pripomore k demonstraciji finih gibov, saj se generirani pogreški pozicioniranja signifikantno zmanjšajo. Z uporabo se zmanjša tudi hitrost in poslabša gladkost demonstracije, a ker je glavni cilj natančna izvedba demonstracije menimo, da uporaba obogatene resničnosti pozitivno pripomore k demonstraciji finih gibov.

5.5 Diskusija

V sklopu študije smo želeli preveriti, ali je kinestetično učenje sodelujočega robota primerno za procese, ki zahtevajo submilimetrsko natančnost. Razvili smo učni vmesnik, ki je združeval sistem obogatene resničnosti in metode učenja gibov na podlagi demonstracij. Ugotovili smo, da uporaba obogatene resničnosti pripomore k natančnosti izvedenih demonstracij, saj se je pogrešek pozicioniranja signifikantno zmanjšal, ko so operaterji uporabljali obogaten prikaz delovnega območja. Kljub temu, da se tako hitrost gibanja zmanjša in gladkost gibanja poslabša, uporaba obogatene resničnosti pozitivno pripomore k demonstraciji finih gibov, saj je pogrešek pozicioniranja v tem primeru ključnega pomena.

Primernost kinestetičnega učenja smo dodatno preizkusili na laboratorijskem procesu identifikacije bakterijskih kolonij z uporabo masne spektrometrije, ki zahteva učenje finih gibov. Proces je bil sestavljen iz postopka odvzema, nanosa in identifikacije kolonij, pri čemer so bile posamezne bakterijske kolonije odvzete iz petrijevke,

nanešene na tarčno ploščo in identificirane z masnim spektrometrom. Tako je eksperiment sestavljalo več neodvisnih problemov. Ne le, da je bilo potrebno ugotoviti, ali je mogoče nalogo dovolj natančno prikazati in posplošiti, temveč tudi, ali bo razpoložljiva senzorika in zasnova orodja na vrhu robota omogočala ustrezno izvedbo nalog. Splošno uspešnost je mogoče določiti prek števila uspešnih identifikacij, vendar pa nam rezultati procesov odvzema in nanosa dajo dodaten vpogled.

Postopek odvzema smo podrobno analizirali na šestih različnih kolonijah. Vidimo lahko, da v vseh primerih kolonija skoraj popolnoma izgine (Slika 5.7). Le v primeru največje kolonije je njena oblika še vedno vidna, a opazno zmanjšana. Razlog se skriva v tem, da so bile začetne dimenzije te kolonije tako velike, da je ni bilo mogoče v celoti pobrati samo z eno iteracijo. Poleg tega je razvidno, da po odvzemu v večini primerov območje, na katerem se je nahajala kolonija, ni temnejše kot njegova neposredna okolica. To pomeni, da je bil odvzem opravljen natančno, brez nepotrebne odvzema plasti gojišča pod kolonijo. Vse to dodatno potrjujejo tudi izračunane vrednosti prikazane v preglednici 5.1. V vseh primerih so se prostornine bakterijskih kolonij znatno zmanjšale, pri čemer je le v nekaterih primerih prišlo tudi do odvzema gojišča, a v neznatnih količinah. To velja tudi za kolonijo z najmanjšim volumnom, katere dimenzije so bile tako majhne, da je bilo fizično nemogoče odvzeti kolonijo bakterij brez delnega odvzema gojišča. Absolutne spremembe v izračunanih volumnih lahko pojasnimo s strukturo bakterijskih kolonij, saj se te zaradi visoke viskoznosti pogosto držijo skupaj. Na splošno rezultati postopka odvzema kažejo, da je mogoče demonstrirati fin gib in ga uspešno zapisati z uporabo metode DMP.

Glede postopka nanašanja lahko iz slike 5.8, (a) razberemo, da kompenzacija sile deluje ustrezno. Tekom celotnega postopka se pozicija vrha robota aktivno spreminja. Manjše sile so omogočile kvalitetnejši nanos in preprečile poškodbo opreme. Med začetnim stikom kontaktna sila presega 1 N, vendar se kmalu zniža tako, da povprečna sila po začetnem stiku znaša približno 0,3 N. Kljub temu, da je proces nanosa relativno ponovljiv, pa je razvidno, da so vzorci nanešeni različno (Slika 5.8, b). Večina jih je enakomerno porazdeljenih po celotnem tarčnem območju, nekateri pa so skoncentrirani na manjših območjih. Zgoščena območja niso zaželeni, saj lahko povzročijo slabšo identifikacijo. Na žalost pa po analizi slike nismo mogli določiti vizualne značilke, na podlagi katere bi lahko predvideli kvaliteto nanosa. Območji A1 in A2 sta, na primer, dosegli odlična rezultata identifikacije in sta bila vzorca tudi enakomerno nanešena, medtem ko področje A4 ni bilo identificirano, kljub temu, da na pogled nanos izgleda

podobno. Po drugi strani sta vzorca C3 in C4 prav tako dosegla odlične rezultate, kljub temu, da njun nanos ni tako enakomeren kot pri A1 in A2. To nakazuje, da na identifikacije ne vpliva zgolj kvaliteta nanosa. Ker je bil postopek nanašanja konsistenten (trajektorija giba, sile), lahko razlike v kakovosti nanosa delno pripišemo fizični strukturi kolonij in njihovi legi na igli. Vpliv lege na igli smo minimizirali tako, da smo med nanašanjem iglo rotirali okoli njene osi. Zaradi nekonsistentnosti rezultatov identifikacije smo se odločili tudi za programsko definirano trajektorijo giba. Tako smo se znebili še dodatnega faktorja, ki bi lahko vplival na kvaliteto identifikacije. Kljub temu, da je bila trajektorija nanosa definirana programsko, se je DMP še vedno izkazala kot primerna metoda za zapis giba, saj je omogočala prilagoditev giba različnim začetnim in končnim pozicijam.

Kljub temu, da v našem primeru skoraj 11 % vzorcev ni bilo mogoče identificirati, so ti rezultati primerljivi z uspešnostjo nanosa laborantov brez dodatne obdelave nanešenega vzorca predstavljene v sklopu študije Chudejove in ostalih [62]. V omenjeni študiji so neuspešnost identifikacij dodatno zmanjšali z obdelavo nanešenega vzorca z uporabo 1 μ L 70 % mravljične kisline. Tako lahko sklepamo, da naš sistem uspešno odvzame in nanese izbrane kolonije, končni rezultat identifikacije pa bi se lahko dodatno izboljšal z nadaljno obdelavo nanešenih vzorcev.

6 Zaključek

V sklopu doktorske disertacije smo si zadali cilj, da analiziramo primernost kinestetičnega učenja finih robotskih gibov. Tako smo primerjali delovanje kinestetičnega učenja z dvema uveljavljenima pristopoma demonstracije finih gibov ter ocenili primernost uporabe funkcionalnosti, ki omogoča vizualno skaliranje. Nadalje smo analizirali primernost uporabe uveljavljenih metod zapisa izvedenih demonstracij in predlagali morebitne nadgradnje, ki bi izboljšale delovanje metod pri zapisu finih gibov. Nazadnje pa smo želeli preveriti še primernost vpeljave robotov v realno okolje in tako uporabo principa učenja z demonstracijo preverili na kliničnem mikrobiološkem procesu, ki zahteva izvedbo finih gibov (tj. identifikacija bakterijskih kolonij z uporabo masne spektrometrije).

V prvi študiji smo tako primerjali različne pristope demonstracije (kinestetično učenje, teleoperacija in CRT) na dveh različnih nalogah, ki sta zahtevali izvajanje finih gibov. V splošnem so operaterji v sklopu obeh nalog izvedli gib s submilimetrsko natančnostjo, a je pristop CRT dosegel najvišjo stopnjo natančnosti. Kinestetično učenje je zaradi nepopolne kompenzacije dinamičnega modela zahtevalo več truda za natančno pozicioniranje, zaradi česar so bili pogreški pozicioniranja večji. Z uporabo funkcije vizualnega skaliranja pa se je generiran pogrešek zmanjšal do te mere, da so se razlike med posameznimi pristopi zmanjšale in je generiran pogrešek z uporabo kinestetičnega učenja postal primerljiv pogrešku generiranem z uporabo pristopa CRT. Tako smo na podlagi izsledkov prve študije ugotovili, da je kinestetično učenje primerno za generiranje finih gibov, prav tako pa je dobrodošla uporaba orodij, ki omogočajo vizualno skaliranje delovnega okolja, saj to pozitivno vpliva na natančnost izvedenih demonstracij. Tekom načrtovanja te študije smo se odločili, da primerjavo izvedemo na gibih, ki so omejeni v ravnini. To se nam je zdelo primerno, saj smo za primerjavo pristopov potrebovali gibe, ki niso kompleksni. Tekom nadaljnjih iteracij študije lahko namreč nadalje povečujemo kompleksnost z izvajanjem gibov izven ravnine ali celo z

izvajanjem gibov pri kateri je potrebno prilagajati orientacijo vrha robota. Omejitev v ravnino nam je dodatno olajšala in pohitrila razvoj okolja za prikaz referenčne poti, saj bi za kompleksnejše gibe najverjetneje potrebovali razvoj obogatene okolja s prikazom referenčne poti z uporabo očal za navidezno resničnost. Za demonstracijo orientacije vrha robota pa bi bila potrebna v primeru pristopov CRT in teleoperacija tudi nadgradnja sistemov upravljanja. Kot izziv za prihodnje tako ostaja podrobna študija demonstriranja kompleksnejših gibov. V sklopu tretje študije smo že postavili osnove za nadaljnje delo saj smo razvili obogateno okolje za demonstracijo finih gibov.

V sklopu druge študije smo nadalje preučili delovanje posameznih metod zapisa demonstracij in določili njihovo primernost za zapis finih gibov. Analizirali smo metodi DMP in GMM. Ugotovili smo, da sta obe metodi sposobni zapisa finih delov giba z ustreznim povečevanjem števila parametrov (tj. baznih funkcij v primeru metode DMP in komponent v primeru metode GMM). Pri tem se metoda DMP izkaže za boljše, saj je računsko manj zahtevna od metode GMM in povečevanje števila baznih funkcij nima občutnega vpliva na potreben čas računanja. Nasprotno se pri metodi GMM s povečevanjem števila komponent čas računanja močno povečuje. Ker dolgi časi učenja niso želeni, smo v sklopu študije predlagali nadgradnjo metode GMM, ki namesto povečevanja števila komponent, ustrezen zapis finih gibov doseže s prilagajanjem porazdelitve komponent vzdolž podatkov. Z nadgradnjo metode smo omogočili ustrezen zapis finih gibov, brez občutnega povečevanja števila komponent in posledično računske kompleksnosti zapisa. V prihodnje bi se to nadgradnjo lahko implementiralo tudi v sklopu drugih metod zapisa kot so DMP, ProMP in KMP, a je implementacija najbolj smiselna ravno pri metodah, ki so že v osnovi bolj kompleksne in posledično računsko potratnejše (npr. KMP). V sklopu te raziskave bi bilo potrebno izvesti tudi podrobno analizo potrebnega časa izračunavanja, saj bi le tako lahko ustrezno primerjali vpliv nadgradnje na čas izračunavanja pri uporabi različnih metod zapisa. Dodatno pa bi lahko predstavljeno metodo preizkusili ne le na simuliranih poteh, ampak tudi na zajetih demonstracijah.

Na podlagi prve študije smo ugotovili, da je z uporabo kinestetičnega učenja možno generiranje finih gibov. Tako smo v sklopu tretje študije primernost kinestetičnega učenja dodatno preizkusili na klinično mikrobiološkem procesu identifikacije bakterijskih kolonij z uporabo masne spektrometrije. Splošno uspešnost smo določili na podlagi števila uspešnih identifikacij, rezultati procesov odvzema in nanosa pa so nam dali še dodaten vpogled. Tako smo analizirali 56 bakterijskih kolonij dveh različnih vrst (*A.*

baumannii, *S. epidermidis*), pri čemer je bila skupna uspešnost identifikacije 89,2 %. Kljub temu, da skoraj 11 % vzorcev ni bilo mogoče identificirati, so ti rezultati primerljivi z uspešnostjo nanosa laboratorijskih tehnikov brez dodatne obdelave nanešenih vzorcev [62]. Pri analizi procesa odvzema kolonij bakterij je razvidno, da je odvzem, kljub majhnim dimenzijam bakterijskih kolonij, opravljen natančno, brez nepotrebne odvzema gojišča pod kolonijo. To nakazuje primernost uporabe kinestetičnega učenja in metode DMP. Težje podajamo zaključke za proces nanosa kolonije bakterij, saj kljub temu, da je bil proces nanosa izveden konsistentno in z ustrezno kompenzacijo kontaktne sile, bakterijska kolonija ni bila vedno konsistentno porazdeljena po tarči, kar lahko povzroča slabšo identifikacijo. Žal so bile pripadajoče ocene identifikacij ne-konsistentne glede na kvaliteto nanosa, kar nakazuje, da na končno oceno ni vplivala zgolj kvaliteta nanosa, temveč tudi procesi ločeni od robota, kot je na primer nanos matriksa na tarčo po nanosu bakterijske kolonije. Tekom procesa končne analize rezultatov smo namreč dobili informacijo, da smo v sklopu meritev uporabljali matriks nižje kvalitete, kar bi lahko signifikatno znižalo stopnjo identifikacij. Ta matriks je bil uporabljen zaradi poteka pandemije SARS-CoV-2 in posledičnih problemov z dobavnimi roki s strani proizvajalca. Kljub obetavnim rezultatom uporabe kinestetičnega učenja v kliničnem mikrobiološkem okolju, bi zaradi omenjenih problematik v prihodnje lahko izvedli enako študijo, le da se pri tem uporablja matriks standardne kvalitete. Dodatno bi se lahko študijo nadgradilo še z analizo vpliva nadaljne obdelave nanešenih vzorcev (npr. 1 μ L 70 % mravljične kisline). Tako bi uspešnost vpeljave kinestetičnega učenja v klinično mikrobiološko okolje lahko podprli še z bolj zanesljivimi rezultati identifikacije. Ločeno smo v sklopu te študije preverili tudi uporabnost orodja, ki omogoča vizualno skaliranje delovnega prostora, saj se je v sklopu prve študije izkazalo, da signifikantno pripomore k zmanjšanju pogreška pozicioniranja. Tako smo v sklopu učnega vmesnika, ki je vseboval funkcionalnost metode DMP, razvili še sistem obogatene okolja, v sklopu katerega so operaterji lahko demonstrirali nalogo odvzema kolonije, pri čemer jim je sistem omogočal povečan prikaz delovnega okolja. Tudi v sklopu te študije se je izkazalo, da povečan prikaz signifikantno izboljša natančnost demonstracije. To obogateno okolje bi lahko nadalje služilo tudi v sklopu nadgradnje prve študije kot funkcionalna osnova za generiranje referenčne poti v prostoru. Tako smo v sklopu tretje študije ugotovili, da je implementacija sodelujočih robotov v klinično mikrobiološko okolje možna, saj je z uporabo kinestetičnega učenja in ustreznih metod zapisa možno učenje finih gibov. Natančnost učenja pa je še dodatno izboljšana z uporabo pristopov, kot je obogatena resničnost, ki omogočajo skaliranje delovnega prostora in

posledično lažje zaznavanje odstopanj.

Izvirni prispevki doktorske disertacije

- Ocena primernosti kinestetičnega učenja za demonstracijo finih gibov v primerjavi z uveljavljenima pristopoma teleoperacije in kooperativnega robotskega orodja na podlagi dosežene natančnosti in uporabniške izkušnje.
 - Ugotovili smo, da je kinestetično učenje primerno za demonstracijo finih gibov, a da je natančnost še boljša z uporabo vizualnega skaliranja delovnega prostora pod vrhom robota.
 - Baumkircher, Aljaž, Marko Munih, and Matjaž Mihelj. “Performance analysis of learning from demonstration approaches during a fine movement generation.” *IEEE Transactions on Human-Machine Systems* 51.6 (2021): 653-662.
- Ocena primernosti metod DMP in GMM za zapis finih gibov in predlog nadgradnje metode GMM za bolj učinkovit zapis finih gibov.
 - Ugotovili smo, da sta tako metodi DMP kot GMM ustrezni za zapis finih gibov, a je metoda GMM izrazito računsko potratna.
 - S predlogom nadgradnje metode GMM izboljšamo učinkovitost metode, saj omogočimo ustrezen zapis finih gibov z ustrezno prilagoditvijo gostote obstoječih komponent vzdolž podatkov, namesto s povečevanjem števila komponent in posledično računske kompleksnosti.
 - Baumkircher, Aljaž, Marko Munih, and Matjaž Mihelj. “Improving Kinesthetic Teaching of Fine Tasks Using a Teaching Agent.” *International Conference on Robotics in Alpe-Adria Danube Region*. Springer, Cham, 2022.

- Metoda kinestetičnega učenja finih robotskih gibov z uporabo učnega vmesnika za namen klinično mikrobiološke aplikacije in validacija ustreznosti robotskih gibov na osnovi kvalitete izvedenih laboratorijskih preiskav.
 - Ugotovili smo, da je kinestetično učenje primerno za vpeljavo sodelujočega robota v laboratorijsko okolje.
 - Uporaba učnega vmesnika signifikantno izboljša natančnost izvedenih demonstracij ter z uporabo metode DMP omogoči prilagajanje demonstrirane trajektorije trenutnim zahtevam.
 - Baumkircher, Aljaz, Katja Seme, Marko Munih, and Matjaž Mihelj. “Collaborative Robot Precision Task in Medical Microbiology Laboratory.” *Sensors* 22.8 (2022): 2862.

Literatura

- [1] B. Akgun, M. Cakmak, K. Jiang in A. L. Thomaz, “Keyframe-based learning from demonstration,” *International Journal of Social Robotics*, vol. 4, št. 4, str. 343–355, 2012.
- [2] C. Schou, J. S. Damgaard, S. Bøgh in O. Madsen, “Human-robot interface for instructing industrial tasks using kinesthetic teaching,” v *IEEE ISR 2013*, str. 1–6, IEEE, 2013.
- [3] K. Fischer, F. Kirstein, L. C. Jensen, N. Krüger, K. Kukliński, M. V. aus der Wieschen in T. R. Savarimuthu, “A comparison of types of robot control for programming by demonstration,” v *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, str. 213–220, IEEE, 2016.
- [4] B. Akgun, M. Cakmak, J. W. Yoo in A. L. Thomaz, “Trajectories and keyframes for kinesthetic teaching: A human-robot interaction perspective,” v *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, str. 391–398, 2012.
- [5] B. Akgun in A. Thomaz, “Simultaneously learning actions and goals from demonstration,” *Autonomous Robots*, vol. 40, št. 2, str. 211–227, 2016.
- [6] Y. Huang, L. Rozo, J. Silvério in D. G. Caldwell, “Kernelized movement primitives,” *The International Journal of Robotics Research*, vol. 38, št. 7, str. 833–852, 2019.
- [7] A. Paraschos, C. Daniel, J. Peters in G. Neumann, “Using probabilistic movement primitives in robotics,” *Autonomous Robots*, vol. 42, št. 3, str. 529–551, 2018.
- [8] F. Steinmetz, A. Montebelli in V. Kyrki, “Simultaneous kinesthetic teaching of positional and force requirements for sequential in-contact tasks,” v *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, str. 202–209, IEEE, 2015.
- [9] D. Berio, S. Calinon in F. F. Leymarie, “Learning dynamic graffiti strokes with a compliant robot,” v *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, str. 3981–3986, IEEE, 2016.

-
- [10] J. Liang, J. Mahler, M. Laskey, P. Li in K. Goldberg, “Using dvrk teleoperation to facilitate deep learning of automation tasks for an industrial robot,” v *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*, str. 1–8, IEEE, 2017.
- [11] S. Calinon, D. Bruno, M. S. Malekzadeh, T. Nanayakkara in D. G. Caldwell, “Human–robot skills transfer interfaces for a flexible surgical robot,” *Computer methods and programs in biomedicine*, vol. 116, št. 2, str. 81–96, 2014.
- [12] B. Huang, M. Ye, Y. Hu, A. Vandini, S.-L. Lee in G.-Z. Yang, “A multirobot cooperation framework for sewing personalized stent grafts,” *IEEE Transactions on Industrial Informatics*, vol. 14, št. 4, str. 1776–1785, 2017.
- [13] J. J. Abbott, G. D. Hager in A. M. Okamura, “Steady-hand teleoperation with virtual fixtures,” v *The 12th IEEE International Workshop on Robot and Human Interactive Communication, 2003. Proceedings. ROMAN 2003.*, str. 145–151, IEEE, 2003.
- [14] A. Bettini, P. Marayong, S. Lang, A. M. Okamura in G. D. Hager, “Vision-assisted control for manipulation using virtual fixtures,” *IEEE Transactions on Robotics*, vol. 20, št. 6, str. 953–966, 2004.
- [15] R. Kumar, G. D. Hager, A. Barnes, P. Jensen in R. H. Taylor, “An augmentation system for fine manipulation,” v *International Conference on Medical Image Computing and Computer-Assisted Intervention*, str. 956–965, Springer, 2000.
- [16] B. Mitchell, J. Koo, I. Iordachita, P. Kazanzides, A. Kapoor, J. Handa, G. Hager in R. Taylor, “Development and application of a new steady-hand manipulator for retinal surgery,” v *Proceedings 2007 IEEE International Conference on Robotics and Automation*, str. 623–629, IEEE, 2007.
- [17] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor in S. Schaal, “Dynamical movement primitives: learning attractor models for motor behaviors,” *Neural computation*, vol. 25, št. 2, str. 328–373, 2013.
- [18] S. Calinon, F. Guenter in A. Billard, “On learning, representing, and generalizing a task in a humanoid robot,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, št. 2, str. 286–298, 2007.
- [19] S. Calinon, “A tutorial on task-parameterized movement learning and retrieval,” *Intelligent service robotics*, vol. 9, št. 1, str. 1–29, 2016.
- [20] L. S. Garcia, *Clinical microbiology procedures handbook*, vol. 1. American Society for Microbiology Press, 2010.

-
- [21] N. Rupp, K. Peschke, M. Köppl, D. Drissner in T. Zuchner, “Establishment of low-cost laboratory automation processes using autoit and 4-axis robots,” *SLAS technology*, 2022.
- [22] C. Huang, K. He, C. Liu, X. Fu in R. Du, “A colony picking robot with multi-pin synchronous manipulator,” v *2018 IEEE International Conference on Information and Automation (ICIA)*, str. 7–12, IEEE, 2018.
- [23] S. Yang in J. Evans, “Metering and dispensing of powder; the quest for new solid free-forming techniques,” *Powder Technology*, vol. 178, št. 1, str. 56–72, 2007.
- [24] I. Holland in J. A. Davies, “Automation in the life science research laboratory,” *Frontiers in Bioengineering and Biotechnology*, vol. 8, str. 571777, 2020.
- [25] A. L. Bailey, N. Ledebor in C.-A. D. Burnham, “Clinical microbiology is growing up: the total laboratory automation revolution,” *Clinical chemistry*, vol. 65, št. 5, str. 634–643, 2019.
- [26] A. A. Evangelopoulos, M. Dalamaga, K. Panoutsopoulos in K. Dima, “Nomenclature and basic concepts in automation in the clinical laboratory setting: a practical glossary,” *Clinical Laboratory*, vol. 59, št. 11-12, str. 1197–1214, 2013.
- [27] G. Da Rin, M. Zoppelletto in G. Lippi, “Integration of diagnostic microbiology in a model of total laboratory automation,” *Laboratory Medicine*, vol. 47, št. 1, str. 73–82, 2016.
- [28] C. D. Hawker, W. L. Roberts, S. B. Garr, L. T. Hamilton, J. R. Penrose, E. R. Ashwood in R. L. Weiss, “Automated transport and sorting system in a large reference laboratory: part 2. implementation of the system and performance measures over three years,” *Clinical chemistry*, vol. 48, št. 10, str. 1761–1767, 2002.
- [29] R. S. Seaberg, R. O. Stallone in B. E. Statland, “The role of total laboratory automation in a consolidated laboratory network,” *Clinical chemistry*, vol. 46, št. 5, str. 751–756, 2000.
- [30] S. Angeletti, M. De Cesaris, J. G. Hart, M. Urbano, M. A. Vitali, F. Fragiasso in G. Di-cuonzo, “Laboratory automation and intra-laboratory turnaround time: experience at the university hospital campus bio-medico of rome,” *Journal of Laboratory Automation*, vol. 20, št. 6, str. 652–658, 2015.
- [31] G. Hoffmann, J. Aufenanger, M. Födinger, J. Cadamuro, A. von Eckardstein, M. Kaeslin-Meyer in W. Hofmann, “Benefits and limitations of laboratory diagnostic pathways,” *Diagnosis*, vol. 1, št. 4, str. 269–276, 2014.

- [32] D. S. Young, “Laboratory automation: smart strategies and practical applications,” *Clinical chemistry*, vol. 46, št. 5, str. 740–745, 2000.
- [33] S. E. Melanson, N. I. Lindeman in P. Jarolim, “Selecting automation for the clinical chemistry laboratory,” *Archives of pathology & laboratory medicine*, vol. 131, št. 7, str. 1063–1069, 2007.
- [34] I. Burckhardt, “Laboratory automation in clinical microbiology,” *Bioengineering*, vol. 5, št. 4, str. 102, 2018.
- [35] H. Fleischer, D. Baumann, S. Joshi, X. Chu, T. Roddelkopf, M. Klos in K. Thurow, “Analytical measurements and efficient process generation using a dual-arm robot equipped with electronic pipettes,” *Energies*, vol. 11, št. 10, str. 2567, 2018.
- [36] X. Chu, T. Roddelkopf, H. Fleischer, N. Stoll, M. Klos in K. Thurow, “Flexible robot platform for sample preparation automation with a user-friendly interface,” v *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, str. 2033–2038, IEEE, 2016.
- [37] X. Chu, M. Klos, K. Thurow in H. Fleischer, “Efficient application of dual-arm robots in analytical measurements using motion frames,” v *2018 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, str. 1–6, IEEE, 2018.
- [38] M. Wojtynek, J. Leichert in S. Wrede, “Assisted planning and setup of collaborative robot applications in modular production systems,” v *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, str. 387–394, IEEE, 2020.
- [39] C. Rocha, J. Dias, A. P. Moreira, G. Veiga in P. Costa, “A kinesthetic teaching approach for automating micropipetting repetitive tasks,” *The International Journal of Advanced Manufacturing Technology*, vol. 118, št. 1, str. 651–663, 2022.
- [40] A. Baumkircher, K. Seme, M. Munih in M. Mihelj, “Collaborative robot precision task in medical microbiology laboratory,” *Sensors*, vol. 22, št. 8, str. 2862, 2022.
- [41] R. Patel, “Matrix-assisted laser desorption ionization–time of flight mass spectrometry in clinical microbiology,” *Clinical infectious diseases*, vol. 57, št. 4, str. 564–572, 2013.
- [42] M. Mihelj, T. Nef in R. Riener, “A novel paradigm for patient-cooperative control of upper-limb rehabilitation robots,” *Advanced Robotics*, vol. 21, št. 8, str. 843–867, 2007.
- [43] T. Flash in N. Hogan, “The coordination of arm movements: an experimentally confirmed mathematical model,” *Journal of neuroscience*, vol. 5, št. 7, str. 1688–1703, 1985.

-
- [44] P. I. Corke, "A simple and systematic approach to assigning denavit–hartenberg parameters," *IEEE transactions on robotics*, vol. 23, št. 3, str. 590–594, 2007.
- [45] M. Švaco, B. Šekoranja, F. Šuligoj in B. Jerbić, "Calibration of an industrial robot using a stereo vision system," *Procedia Engineering*, vol. 69, str. 459–463, 2014.
- [46] S. Balasubramanian, A. Melendez-Calderon, A. Roby-Brami in E. Burdet, "On the analysis of movement smoothness," *Journal of neuroengineering and rehabilitation*, vol. 12, št. 1, str. 1–11, 2015.
- [47] D. J. Berndt in J. Clifford, "Using dynamic time warping to find patterns in time series.," v *KDD workshop*, vol. 10, str. 359–370, Seattle, WA, USA:, 1994.
- [48] H. Huynh in L. S. Feldt, "Conditions under which mean square ratios in repeated measurements designs have exact f-distributions," *Journal of the American Statistical Association*, vol. 65, št. 332, str. 1582–1589, 1970.
- [49] E. L. Lehmann in H. J. D'Abbrera, *Nonparametrics: statistical methods based on ranks*. Holden-day, 1975.
- [50] L. A. Marascuilo in M. McSweeney, "Nonparametric and distribution-free methods for the social sciences," 1977.
- [51] R. C. Blair in J. J. Higgins, "Comparison of the power of the paired samples t test to that of wilcoxon's signed-ranks test under various population shapes.," *Psychological Bulletin*, vol. 97, št. 1, str. 119, 1985.
- [52] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the american statistical association*, vol. 32, št. 200, str. 675–701, 1937.
- [53] S. Lee in D. K. Lee, "What is the proper way to apply the multiple comparison test?," *Korean journal of anesthesiology*, vol. 71, št. 5, str. 353, 2018.
- [54] A. Paraschos, C. Daniel, J. R. Peters in G. Neumann, "Probabilistic movement primitives," *Advances in neural information processing systems*, vol. 26, 2013.
- [55] J. Kuha, "Aic and bic: Comparisons of assumptions and performance," *Sociological methods & research*, vol. 33, št. 2, str. 188–229, 2004.
- [56] P. Welch, "The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms," *IEEE Transactions on audio and electroacoustics*, vol. 15, št. 2, str. 70–73, 1967.

- [57] B. W. Buchan in N. A. Ledebøer, “Emerging technologies for the clinical microbiology laboratory,” *Clinical microbiology reviews*, vol. 27, št. 4, str. 783–822, 2014.
- [58] R. Dadoun, “Case study: automation’s impact on productivity and turnaround time,” *MLO: medical laboratory observer*, vol. 34, št. 5, str. 36–38, 2002.
- [59] A. Bizzini, C. Durussel, J. Bille, G. Greub in G. Prod’Hom, “Performance of matrix-assisted laser desorption ionization-time of flight mass spectrometry for identification of bacterial strains routinely isolated in a clinical microbiology laboratory,” *Journal of clinical microbiology*, vol. 48, št. 5, str. 1549–1554, 2010.
- [60] A. Croxatto, G. Prod’Hom, F. Faverjon, Y. Rochais in G. Greub, “Laboratory automation in clinical bacteriology: what system to choose?,” *Clinical Microbiology and Infection*, vol. 22, št. 3, str. 217–235, 2016.
- [61] O. Dauwalder, L. Landrieve, F. Laurent, M. De Montclos, F. Vandenesch in G. Lina, “Does bacteriology laboratory automation reduce time to results and increase quality management?,” *Clinical Microbiology and Infection*, vol. 22, št. 3, str. 236–243, 2016.
- [62] K. Chudejova, M. Bohac, A. Skalova, V. Rotova, C. C. Papagiannitsis, J. Hanzlickova, T. Bergerova in J. Hrabák, “Validation of a novel automatic deposition of bacteria and yeasts on maldi target for maldi-tof ms-based identification using maldi colonyst robot,” *PLoS One*, vol. 12, št. 12, str. e0190038, 2017.
- [63] P. Jones, A. Watson, M. Davies in S. Stubbings, “Integration of image analysis and robotics into a fully automated colony picking and plate handling system,” *Nucleic acids research*, vol. 20, št. 17, str. 4599–4606, 1992.
- [64] D. R. Briner, A. D. Sardhara in T. G. Sugar, “A multi-pin end-effector for a robotic colony picker,” v *2009 ASME Early Career Technical Conference*, str. 222–228, 2009.
- [65] C. Huang, K. He, C. Liu, X. Fu in R. Du, “A colony picking robot with multi-pin synchronous manipulator,” v *2018 IEEE International Conference on Information and Automation (ICIA)*, str. 7–12, IEEE, 2018.
- [66] A. Fabritius, D. Ng, A. M. Kist, M. Erdogan, R. Portugues in O. Griesbeck, “Imaging-based screening platform assists protein engineering,” *Cell chemical biology*, vol. 25, št. 12, str. 1554–1561, 2018.
- [67] A. Fuller, Z. Fan, C. Day in C. Barlow, “Digital twin: Enabling technologies, challenges and open research,” *IEEE access*, vol. 8, str. 108952–108971, 2020.

-
- [68] F. Tao, H. Zhang, A. Liu in A. Y. Nee, “Digital twin in industry: State-of-the-art,” *IEEE Transactions on industrial informatics*, vol. 15, št. 4, str. 2405–2415, 2018.
- [69] H.-K. Wu, S. W.-Y. Lee, H.-Y. Chang in J.-C. Liang, “Current status, opportunities and challenges of augmented reality in education,” *Computers & education*, vol. 62, str. 41–49, 2013.
- [70] L. B. Rosenberg, “The use of virtual fixtures as perceptual overlays to enhance operator performance in remote environments.,” tech. rep., Stanford Univ Ca Center for Design Research, 1992.
- [71] J. Steuer, “Defining virtual reality: Dimensions determining telepresence,” *Journal of communication*, vol. 42, št. 4, str. 73–93, 1992.
- [72] P. Milgram in F. Kishino, “A taxonomy of mixed reality visual displays,” *IEICE TRANSACTIONS on Information and Systems*, vol. 77, št. 12, str. 1321–1329, 1994.

Dodatek A: Demonstriranje finih dinamičnih robotskih gibov

Pri nekaterih opravilih je za uspešno izvedbo naloge, bolj kot potek poti, pomembna ustrežna dinamika giba (tj. potek hitrosti, pospeška). To predstavlja dodaten izziv pri pristopih učenja z demonstracijo, saj se operaterji poleg težavnosti upravljanja z roboti soočajo tudi s pomanjkanjem povratne informacije o poteku giba. V primeru demonstracije finih gibov, pri katerih je pomembna submilimetrski natančnost pozicioniranja, ima operater povratno informacijo, saj je zmožen oceniti trenutno točnost pozicioniranja. Za ocenjevanje dinamike giba pa ljudje nimamo razvitih sposobnosti določevanja absolutnih vrednosti odstopanj, temveč ta razdelimo po kategorijah (npr. prepočasi, prehitro), ali pa ocenimo dinamiko giba glede na njegove posledice. Primer slednjega je farmacevtski proces ročnega doziranja praška z uporabo spatule, pri katerem oseba oceni dinamiko giba posredno na podlagi izmerjene mase stresenega praška. Izvedba giba, pri katerem dinamični potek zadostno sledi želenemu poteku, je v primeru omenjenega procesa zaradi relativno kratke opravljene poti giba še toliko bolj zahtevna. Pri tako imenovanih finih dinamičnih gibih operater namreč še težje oceni trenutni potek giba in posledično ne izvede demonstracije z absolutno točnostjo.

6.1 Eksperimentalni sistem in protokol študije

Da smo preverili primernost demonstracije finih dinamičnih gibov z uporabo omenjenih pristopov učenja (kinestetično učenje, teleoperacija, CRT), smo pripravili študijo, v sklopu katere so operaterji demonstrirali enostavne fine dinamične gibe. Za izvedbo eksperimenta smo uporabili sistem predstavljen v študiji demonstriranja finih gibov (poglavje 3) in prikazanem na sliki 3.1.

Fini dinamični gibi so gibi, pri katerih je pomemben dinamični potek, a so izvedeni

na relativno kratki razdalji. Tako smo si za referenčni gib izbrali enostavno periodično gibanje vzdolž 1 cm dolge ravne črte, ki smo jo prikazovali na LCD zaslonu. Za boljšo predstavo operaterja o zahtevani dinamiki giba, smo vzdolž črte izrisovali tarčo, ki je nakazovala trenutno želeno lego in služila kot metronom, ki operaterju daje takt za izvedbo giba. Tarča se je namreč premikala vzdolž črte z enakomerno hitrostjo. Zmožnost demonstracije dinamičnega giba smo preverili pri dveh različnih referenčnih frekvencah metronoma in sicer pri frekvenci $f_1 = 0,8$ Hz in $f_2 = 1,6$ Hz. Frekvence smo izbrali na podlagi izkušenj pridobljenih tekom načrtovanja eksperimenta, pri čemer so operaterji gib, v primeru frekvence f_1 , izvajali brez potrebe po uvajanju, medtem ko je bilo operaterjem generiranje giba pri frekvenci f_2 že občutno zahtevnejše.

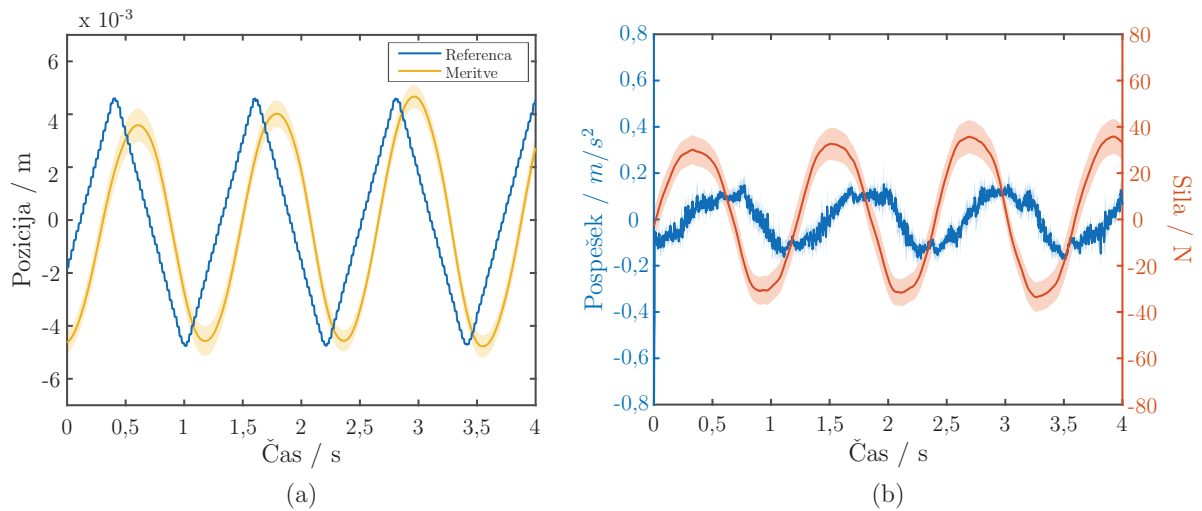
Študijo smo izvedli na 12-ih osebah, pri čemer je vsaka oseba dinamični gib pri izbrani frekvenci demonstrirala približno 15 sekund ter za vsako frekvenco izvedla po tri demonstracije. Za namene analize smo nato iz vsake demonstracije uporabili izsek dolg 4 s, pri čemer se je začetek izseka začel po 8 s demonstracije. To smo izvedli za vse tri pristope, pri čemer so v primeru teleoperacije vrednosti elementov diagonalne matrike \mathbf{K}_{sp} enake 1 (Enačba 2.11).

6.2 Rezultati

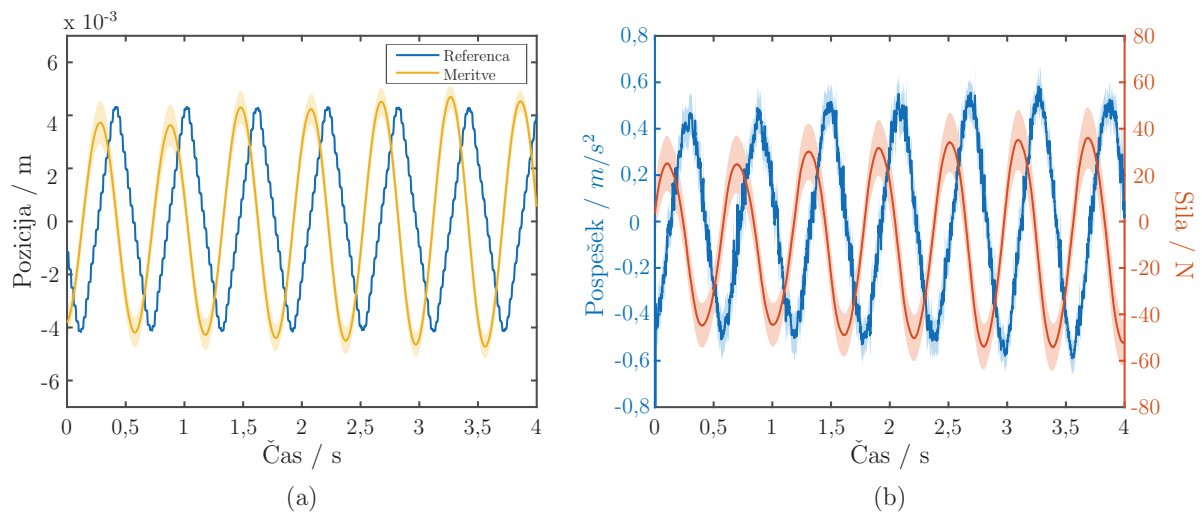
Razultate študije bomo prikazali ločeno glede za posamezen pristop demonstracije. Za ustrezno analizo študije smo prikazali referenčne in izmerjene časovne poteke pozicije vrha robota, na podlagi katerih smo ocenili dejansko frekvenco giba, časovno zakasnitev med referenčnim ter demonstriranim gibom in odstopanja v amplitudi izvedenega giba. Poleg tega smo prikazali tudi generirane pospeške vrha robota in sile izmerjene s senzorjem sile ATI Nano17.

Kinestetično učenje

Časovni potek demonstracije z uporabo pristopa kinestetičnega učenja je v primeru referenčne frekvence giba f_1 prikazan na sliki 6.1, medtem ko je v primeru referenčne frekvence giba f_2 prikazan na sliki 6.2. Na podlagi časovnega poteka izmerjene pozicije vrha robota smo lahko določili frekvenco, časovno zakasnitev ter amplitudna odstopanja demonstriranega giba. Na podlagi časovnega poteka izmerjene sile pa smo določili vrednost maksimalnih vsiljenih sil v sklopu posameznega giba. Vse te vrednosti so predstavljene v tabeli 6.1.



Slika 6.1: Prikaz demonstriranega dinamičnega giba z uporabo kinestetičnega učenja pri referenčni frekvenci f_1 . Graf (a) prikazuje časovni potek pozicije vrha robota v primerjavi z referenco, medtem ko graf (b) prikazuje pospeške vrha robota in vrednost vsiljenih sil.



Slika 6.2: Prikaz demonstriranega dinamičnega giba z uporabo kinestetičnega učenja pri referenčni frekvenci f_2 . Graf (a) prikazuje časovni potek pozicije vrha robota v primerjavi z referenco, medtem ko graf (b) prikazuje pospeške vrha robota in vrednost vsiljenih sil.

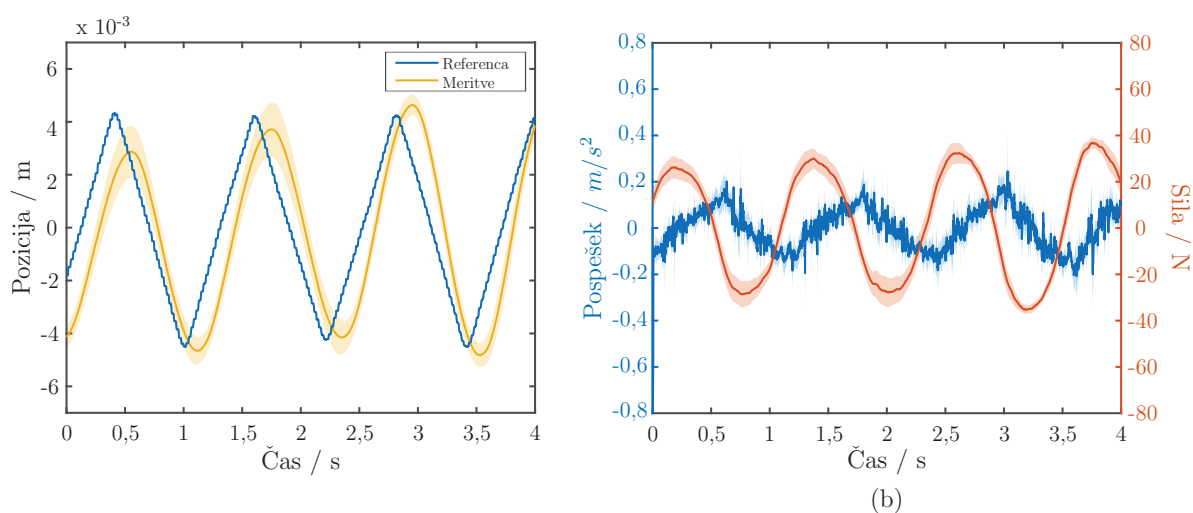
Kooperativna robotska orodja

Časovni potek demonstracije z uporabo pristopa CRT je v primeru referenčne frekvence giba f_1 prikazan na sliki 6.3, medtem ko je v primeru referenčne frekvence giba f_2 prikazan na sliki 6.4. Na podlagi časovnega poteka izmerjene pozicije vrha robota

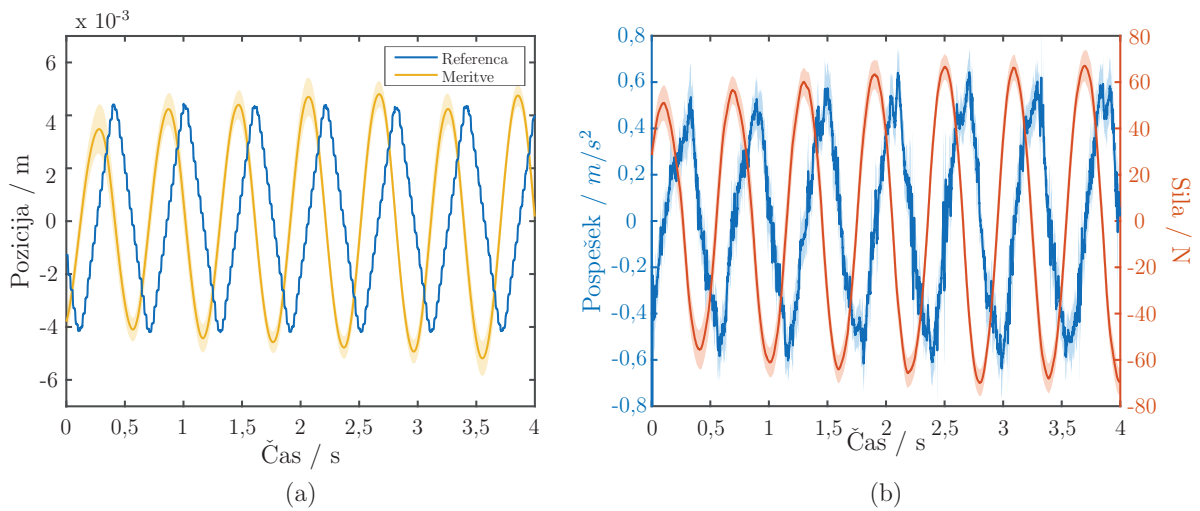
	$f_1 = 0,8 \text{ Hz}$	$f_2 = 1,6 \text{ Hz}$
<i>Izmerjena frekvenca</i> [Hz]	$0,81 \pm 0,02$	$1,61 \pm 0,03$
<i>Časovni zamik</i> [ms]	171 ± 25	480 ± 9
<i>Amplitudno odstopanje</i> [mm]	$0,19 \pm 0,49$	$0,22 \pm 0,36$
<i>Max. generirane sile</i> [N]	$34,7 \pm 20,7$	$48,9 \pm 28,6$

Tabela 6.1: Prikaz izračunanih značilnk, ki omogočajo oceno kvalitete izvedbe dinamičnega giba z uporabo kinestetičnega učenja.

smo lahko določili frekvenco, časovno zakasnitev ter amplitudna odstopanja demonstriranega giba. Na podlagi časovnega poteka izmerjene sile pa smo določili vrednost maksimalnih vsiljenih sil v sklopu posameznega giba. Vse te vrednosti so predstavljene v tabeli 6.2.



Slika 6.3: Prikaz demonstriranega dinamičnega giba z uporabo pristopa CRT pri referenčni frekvenci f_1 . Graf (a) prikazuje časovni potek pozicije vrha robota v primerjavi z referenco, medtem ko graf (b) prikazuje pospeške vrha robota in vrednost vsiljenih sil.



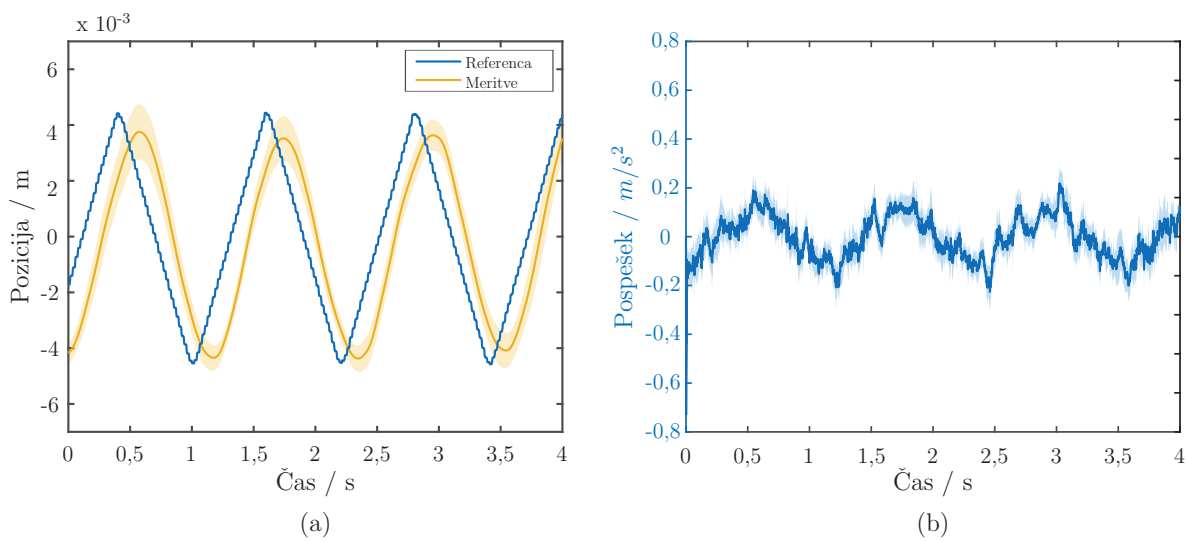
Slika 6.4: Prikaz demonstriranega dinamičnega giba z uporabo pristopa CRT pri referenčni frekvenci f_2 . Graf (a) prikazuje časovni potek pozicije vrha robota v primerjavi z referenco, medtem ko graf (b) prikazuje pospeške vrha robota in vrednost vsiljenih sil.

	$f_1 = 0,8 \text{ Hz}$	$f_2 = 1,6 \text{ Hz}$
Izmerjena frekvenca [Hz]	$0,81 \pm 0,04$	$1,60 \pm 0,08$
Časovni zamik [ms]	128 ± 22	482 ± 13
Amplitudno odstopanje [mm]	$0,32 \pm 0,63$	$0,30 \pm 0,52$
Max. generirane sile [N]	$32,6 \pm 9,9$	$63,5 \pm 18,1$

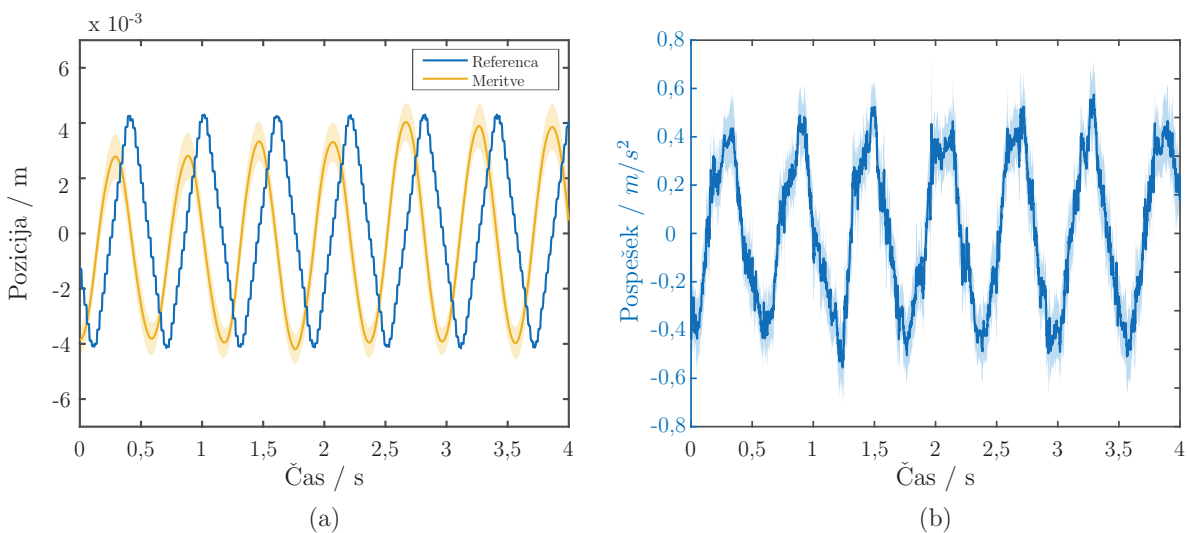
Tabela 6.2: Prikaz izračunanih značilnk, ki omogočajo oceno kvalitete izvedbe dinamičnega giba z uporabo pristopa CRT.

Teleoperacija

Časovni potek demonstracije z uporabo pristopa teleoperacije je v primeru referenčne frekvence giba f_1 prikazan na sliki 6.5, medtem ko je v primeru referenčne frekvence giba f_2 prikazan na sliki 6.6. Na podlagi časovnega poteka izmerjene pozicije vrha robota smo lahko določili frekvenco, časovno zakasnitev ter amplitudna odstopanja demonstriranega giba. Na podlagi časovnega poteka izmerjene sile pa smo določili vrednost maksimalnih vsiljenih sil v sklopu posameznega giba. Vse te vrednosti so predstavljene v tabeli 6.3.



Slika 6.5: Prikaz demonstriranega dinamičnega giba z uporabo pristopa teleoperacije pri referenčni frekvenci f_1 . Graf (a) prikazuje časovni potek pozicije vrha robota v primerjavi z referenco, medtem ko graf (b) prikazuje pospeške vrha robota in vrednost vsiljenih sil.



Slika 6.6: Prikaz demonstriranega dinamičnega giba z uporabo pristopa teleoperacije pri referenčni frekvenci f_2 . Graf (a) prikazuje časovni potek pozicije vrha robota v primerjavi z referenco, medtem ko graf (b) prikazuje pospeške vrha robota in vrednost vsiljenih sil.

	$f_1 = 0,8 \text{ Hz}$	$f_2 = 1,6 \text{ Hz}$
<i>Izmerjena frekvenca</i> [Hz]	$0,82 \pm 0,04$	$1,61 \pm 0,06$
<i>Časovni zamik</i> [ms]	149 ± 15	482 ± 14
<i>Amplitudno odstopanje</i> [mm]	$0,39 \pm 0,82$	$0,36 \pm 0,64$

Tabela 6.3: Prikaz izračunanih značilnk, ki omogočajo oceno kvalitete izvedbe dinamičnega giba z uporabo teleoperacije.

6.3 Diskusija

Iz rezultatov je razvidno, da so ob začetku izseka operaterji uspešno sinhronizirani z referenčnim gibanjem, saj je frekvenca poteka demonstriranega giba tako pri frekvenci f_1 kot tudi pri frekvenci f_2 zelo podobna referenčni, pri čemer prihaja do minimalnih odstopanj. Operaterji so sicer konsistentno generirali gib s časovnim zamikom, ki pa je, v primeru referenčne frekvence f_1 , najverjetneje posledica časovnega zamika med operaterjevim zaznavanjem spremembe referenčne pozicije in generiranjem ustreznega premika. Tako sklepamo, ker so vrednosti časovnih zamikov podobne med posameznimi pristopi demonstriranja. V primeru referenčne frekvence f_2 je časovni zamik večji. Zaradi nepopolno kompenzirane dinamike robota je bilo namreč generiranje giba bolj zahtevno. Kljub temu so bili zakasnitveni časi konsistentni. Prihajalo je tudi do manjših amplitudnih odstopanj, saj operaterji niso konsistentno dosegali skrajnih leg. Zanimivo je, da prihaja do odstopanj tako pri nizki kot tudi višji frekvenci. Zaradi lažje izvedbe giba z nižjo frekvenco bi namreč pričakovali natančnejše doseganje skrajnih leg. Odstopanja so razumljiva, saj je bila prioriteta operaterjev posnemanje dinamike giba in ne natančno sledenje referenčni poziciji.

Iz časovnih potekov pospeškov je razvidno, da so generirani pospeški v primeru frekvence f_2 višji, kar je razumljivo, saj ima operater na voljo manj časa, da opravi zahtevano pot. Kljub temu da so si časovni poteki pospeškov glede na izbran pristop demonstracije med seboj podobni, so iz poteka v primeru teleoperacije razvidne nenadne spremembe v pospešku, ki niso razvidne pri ostalih pristopih. Razlog za nenadne spremembe je verjetno v majhni vztrajnosti in dušenju upravljalne naprave.

Za oceno primernosti pristopov pri demonstraciji finih dinamičnih gibov pa je pomemben tudi potek izmerjenih sil na vrhu robota. Meritve med posameznimi pristopi žal niso primerljive, saj je v primeru uporabe pristopa CRT operater silo generiral ne-

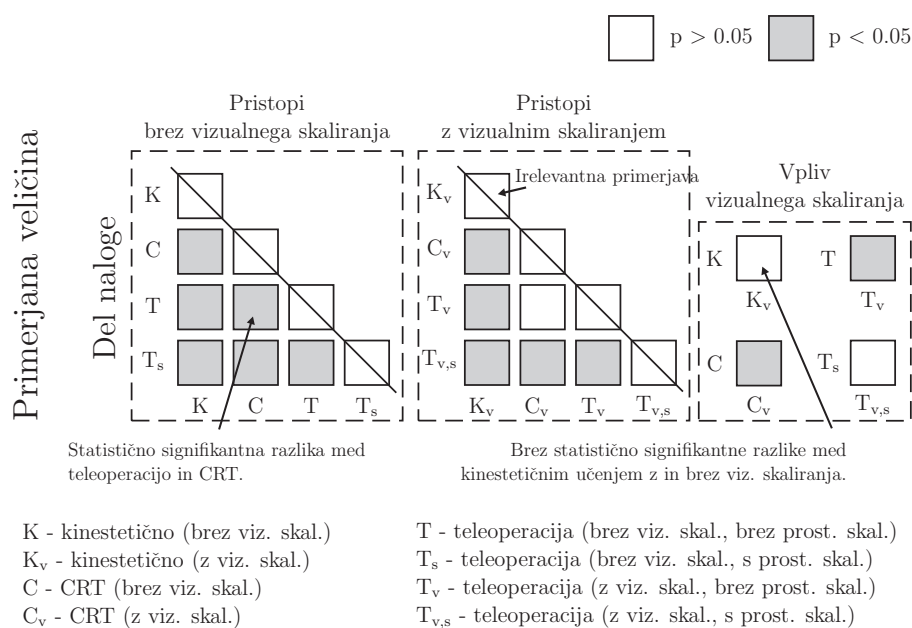
posredno na senzor sile, medtem ko je v primeru kinestetičnega učenja del sile generiran tudi na ostale segmente mehanizma. Pri pristopu teleoperacije operater ni bil v stiku z robotom, tako da so meritve sile irelevantne. Iz prikazanih potekov meritev sile je razvidno, da so izmerjene sile relativno visoke in večje v primeru višje referenčne frekvence f_2 . Razlog za tako visoke sile je nepopolno kompenziran dinamični model robota, ki z vplivi dušenja in vztrajnostjo onemogočijo demonstracijo zahtevanih dinamičnih gibov z uporabo majhnih sil.

Kljub temu da so bili operaterji sposobni generirati zahtevane gibe, kar je razvidno iz izračunanih frekvenc generiranih gibov in relativno majhnih amplitudnih odstopanj, so za izvedbo zahtevanih gibov generirali visoke sile. Zahtevani gibi so bili enostavni, saj so zahtevali zgolj premik vzdolž ene osi pri čemer natančnost pozicioniranja ni bila zahtevana. V kolikor bi referenčni gibi zahtevali gib v večih oseh, z manjšo potjo giba in z zahtevano natančnostjo pozicioniranja, lahko sklepamo, da operaterji z uporabo tako visokih sil kot so bile potrebne v naši študiji, ne bi bili sposobni generirati ustreznega giba. Težko je namreč pričakovati natančno pozicioniranje operaterja med generiranjem sile, ki so v določenih primerih večje od 50 N. V primeru teleoperacije sicer operaterji niso omejeni z generiranimi silami, nastopa pa problem hkratne uporabe večih naprav (npr. robot Phantom in krmilna palica), ki zaradi povečane kompleksnosti sistema operaterju otežijo prenos zelenega giba na izvršno napravo. Zato menimo, da so izbrani pristopi neprimerni za generiranje finih dinamičnih gibov.

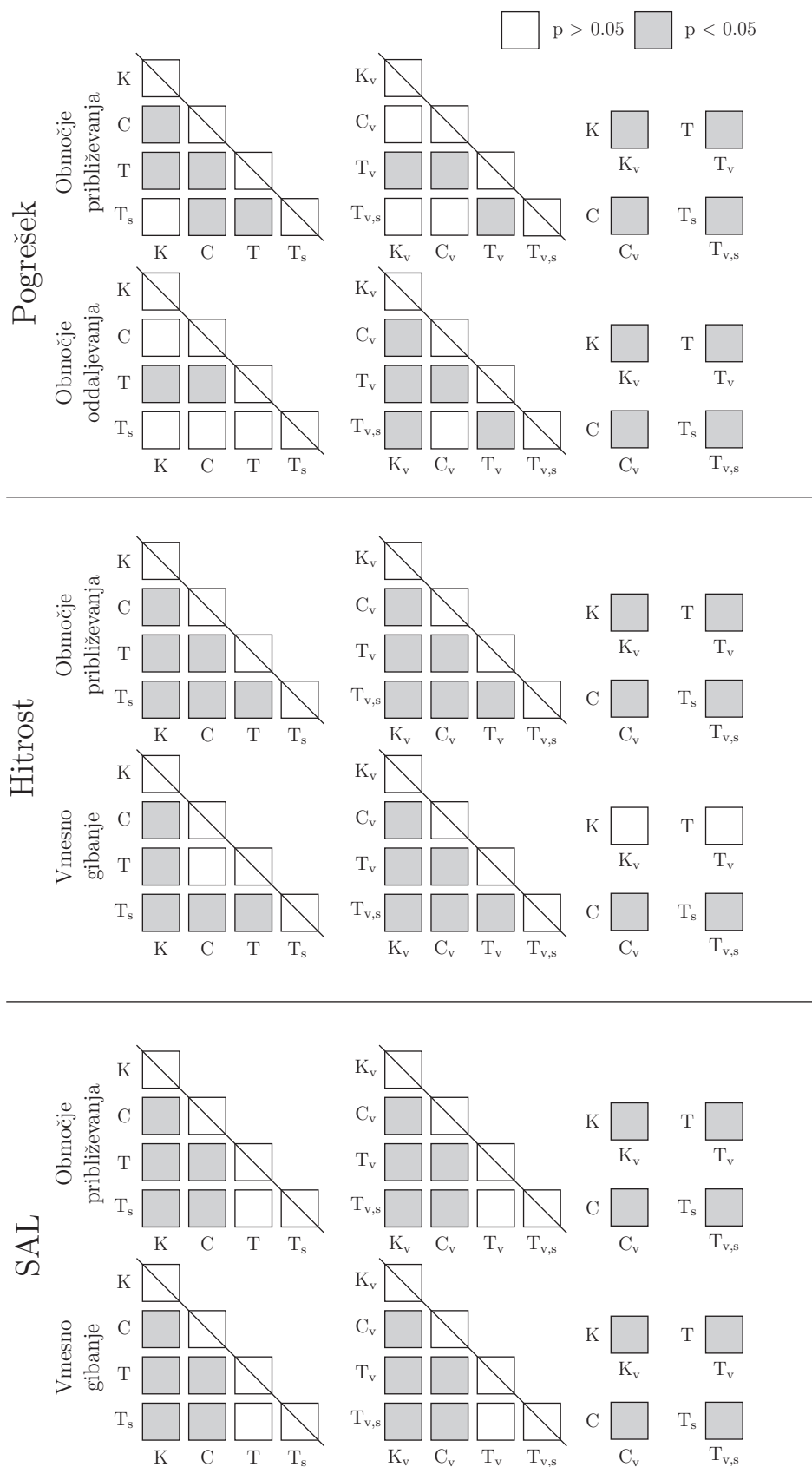
Študija učenja finih dinamičnih gibov je bila nadalje bolj podrobno raziskana v magistrski nalogi z naslovom "Precizno doziranje praškastih materialov s sodelujočim robotom" avtorja Aleša Ručigaja, mentorja Matjaža Mihelja in somentorja Aljaža Baumkircherja.

Dodatek B: Statistična analiza rezultatov študije demonstriranja finih gibov

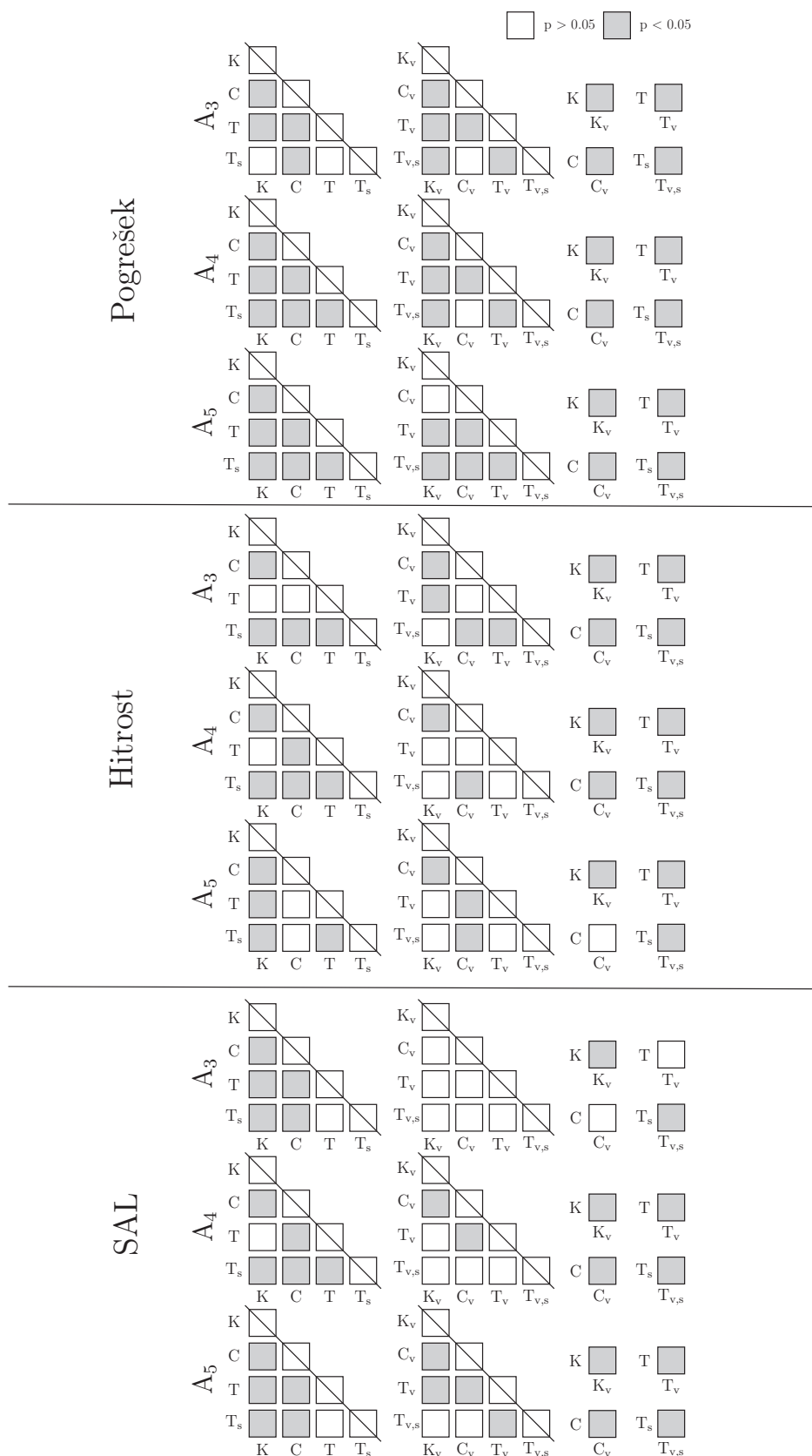
V sklopu tega poglavja prikazujemo statistično analizo rezultatov. Rezultati so predstavljeni ločeno za vsak tip naloge na slikah 6.8 in 6.9. Zaradi velikega števila primerjav smo uporabili grafični prikaz, ki je za lažjo interpretacijo rezultatov razložen na sliki 6.7. Kot že omenjeno v poglavju 3, je bila meja statistične signifikance nastavljena na 0.05. V kolikor je bila vrednost p večja, je bila ničta hipoteza sprejeta, kar pomeni, da med podatki ni statistično signifikantne razlike. V kolikor je bila ničta hipoteza zavržena, je med podatki obstajala statistično signifikantna razlika.



Slika 6.7: Predloga grafičnega prikaza statistične analize.



Slika 6.8: Premik od točke do točke - statistični pregled testiranja ničtih hipotez primerjave različnih pristopov demonstriranja ter vpliva prostorskega in vizualnega skaliranja.



Slika 6.9: Sledenje poti - statistični pregled testiranja ničtih hipotez primerjave različnih pristopov demonstriranja ter vpliva prostorskega in vizualnega skaliranja.