

Demonstracijom podržano robotsko brušenje

Drobac, Pero

Master's thesis / Diplomski rad

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

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:168:373517>

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



Repository / Repozitorij:

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repository](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 104

DEMONSTRACIJOM PODRŽANO ROBOTSKO BRUŠENJE

Pero Drobac

Zagreb, veljača 2025.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 104

DEMONSTRACIJOM PODRŽANO ROBOTSKO BRUŠENJE

Pero Drobac

Zagreb, veljača 2025.

Zagreb, 30. rujna 2024.

DIPLOMSKI ZADATAK br. 104

Pristupnik: **Pero Drobac (0036523011)**
Studij: Informacijska i komunikacijska tehnologija
Profil: Automatika i robotika
Mentor: izv. prof. dr. sc. Matko Orsag

Zadatak: **Demonstracijom podržano robotsko brušenje**

Opis zadatka:

Cilj zadatka je razviti programsko rješenje i pripadajuće elektromehaničke komponente kojim će se omogućiti demonstracijom podržano programiranje robotskih sustava. U sklopu rada potrebno je pregledati i analizirati postojeću literaturu u području demonstracijom podržanog programiranja robotskih sustava. Odabranu metodu potrebno je prilagoditi i implementirati na robotskom sustavu s više stupnjeva slobode. Potrebno je prikupiti podatke o demonstraciji, obraditi ih i prilagoditi kinematičkim i dinamičkim ograničenjima robotskog sustava. Sustav je potrebno ispitati na zadatku brušenja.

Rok za predaju rada: 14. veljače 2025.

Hvala Brunu Mariću na pomoći s implementacijom mog diplomskog rada. Također, zahvala mojem mentoru prof. Matku Orsagu na pruženoj prilici rada na temi ovog diplomskog rada i na susretljivosti i strpljenju. Hvala Tomu Sjekavici i informatičkom klubu Futura na inspiraciji za rad s robotima od malih nogu. Konačno, hvala obitelji i prijateljima za pruženu potporu tijekom studija.

SADRŽAJ

1. Uvod	1
2. Primitivi dinamičkog gibanja	3
2.1. Matematički model	3
2.2. Prilagodba algoritma za različite potrebe	5
3. Robotsko brušenje	10
3.1. Odabir površina za pokrete	10
3.2. Analiza objekta	11
3.3. Programska arhitektura	15
3.4. Demonstracija pokreta	17
3.5. Upravljanje robotom	20
4. Eksperimentalna provjera	21
4.1. Dizajn nosača za alat	21
4.2. Kalibracija	23
4.3. Simulacija robotskog brušenja	24
4.4. Praktična izvedba zadatka	27
5. Zaključak	28
Literatura	29

1. Uvod

Automatizacija je proces korištenja tehnologije, strojeva ili softvera za izvođenje jednostavnih i repetitivnih zadataka uz minimalnu ljudsku intervenciju. Naširoko se koristi u raznim područjima, uključujući proizvodnju, poslovne procese, razvoj softvera i umjetnu inteligenciju.

Mnogo je različite koristi od automatizacije, ali tri su ključna aspekta: smanjenje ljudskog napora, smanjenje dugoročnih troškova i skalabilnost. Obavljanjem repetitivnih zadataka automatizacija omogućava ljudima da više vremena posvete dijelu posla koji zahtjeva kritično razmišljanje ili ljudsku interakciju dok samo izvođenje automatiziranog zadatka može postati efikasnije. Automatizirane zadatke je lakše proširiti - za povećati količinu izvedenih zadataka često je dovoljno nabaviti nove strojeve s istim softverom koji će identičnim izvođenjem osigurati uniformnost rezultata na većoj skali.

Jedno od polja koje ima najveću korist od robotizacije je industrijski sektor koji u tvornicama koristi robote opremljene alatima za izvođenje specifičnih koraka procesa manufacture. Iako se mnogi industrijski procesi mogu podijeliti na zadatke koje je jednostavno zamijeniti klasičnom automatizacijom, strojevi koji obavljaju te zadatke su najčešće visoko specijalizirani za taj specifičan zadatak. Također, neki zadatci mogu biti kompleksnije prirode te je upravljanje alatom potrebno prilagoditi ljudskoj demonstraciji izvođenja zadatka.

Ovaj se rad bavi postavljanjem osnovice rješenja automatizacije koje može biti prilagođeno za drukčije svrhe unutar svojeg radnog okruženja, poglavito prilagodbu izvođenja ovisnu o demonstraciji zadatka. Zadatak je upogoniti podatnu robotsku ruku da obavlja demonstrirani zadatak brušenja metalnog kalupa. Za izvođenje zadatka potrebno je pričvrstiti alat na prihvatnicu alata robotske ruke, umjeriti poziciju kalupa u radnom okruženju te modificirati i replicirati demonstrirani pokret brušenja snimljen specijaliziranim sustavom kamera te efektivno izbrusiti cijelu unutarnju površinu kalupa.

Uređaji korišteni za realizaciju danog zadatka su podatna robotska ruka Kinova

Gen3 sa 7 stupnjeva slobode, metalni kalup za izradu staklene boce, precizna električna bušilica s nastavkom za brušenje i OptiTrack Trio sustav kamera za praćenje pokreta. Programska oprema uključuje alat Motive za upravljanje kamerama i programsku platformu ROS (Robot Operating System) za upravljanje robotskom rukom.

Glavna problematika zadatka je definicija i realizacija softverskog rješenja. Sustav treba koristiti algoritamsku podlogu koja je sposobna snimiti, naučiti, modificirati i replicirati demonstrirani pokret. Primjena također može zahtijevati dodatnu prilagodbu pokreta radnoj okolini.

Kretanje vrha alata je moguće opisati kao skup trodimenzionalnih točaka u Kartezijском prostoru, gdje se svakoj točki pridjeljuje vremenska oznaka, što od putanje čini trajektoriju. Ovu trajektoriju je tada moguće modelirati kao točkasti predmet čije je gibanje opisano kao slijed pomaka do svake sljedeće točke. Derivacijom se svakog pomaka po vremenu izračuna brzina predmeta u vremenskoj oznaci pripadajuće točke, te ponovnom derivacijom dobivenih brzina izračunati akceleracija predmeta. Akceleracija, brzina i pomak su u ovakvom sustavu vektori s tri člana koji opisuju skup vremenski ovisnih točaka u Kartezijском prostoru. Za replikaciju pokreta potrebno je replicirati akceleracije, uz modifikaciju po potrebi. Čak sitnije modifikacije pomaka mogu drastično promijeniti akceleracije koje djeluju na modelirani predmet, stoga je potrebno definirati model koji može prihvatiti promjene u pomaku i sukladno tome definirati potrebne akceleracije.

Koristan pristup rješenju ovog problema pružaju primitivi dinamičkog gibanja (dynamical movement primitives, DMP), utemeljeni na dobro poznatom stabilnom dinamičkom sustavu s dodatnim nelinearnim članom za ispravljanje oblika [1], [2], [3]. Ovo rješenje već ima primjene na praktičnim zadacima s robotskim sustavima, primjerice prenošenje trupova drveta [4].

2. Primitivi dinamičkog gibanja

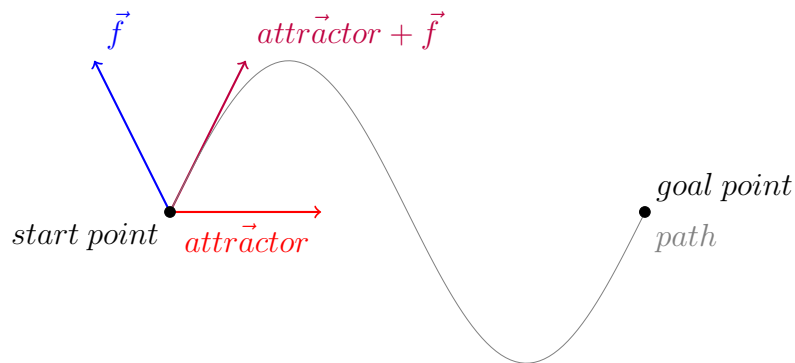
2.1. Matematički model

DMP se temelje na modelu prigušene opruge koja je linearan sustav drugog reda uz dodan nelinearan oblikotvoran član f . Izražava se preko sljedeće jednadžbe:

$$\tau^2 \ddot{y} = \alpha_z (\beta_z (g - y) - \tau \dot{y}) + f, \quad (2.1)$$

gdje τ predstavlja vremensku konstantu trajanja cjelokupne operacije, vektor y predstavlja pomak u trodimenzionalnom prostoru, vektor g predstavlja cilj, odnosno završnu poziciju objekta, α_z i β_z predstavljaju pozitivne konstante koje upravljaju prigušenjem sustava te f predstavlja nelinearan oblikotvoran član. Uz uvjet $f = 0$, ovaj sustav je jednostavan linearni atraktor u točki definiranom vrijednošću g . Konvergencija ovog sustava prema konačnom cilju g oblikovana je parametrima α_z i β_z koji modeliraju prigušenje opruge atraktora. Optimalna se konvergencija (najbrža bez titranja) postiže kritičnim prigušenjem $\beta_z = \alpha_z/4$.

Kako bi putanja atrakcije točke poprimila tražen oblik, f u fizikalnom smislu predstavlja skup dodatnih čimbenika koji utječu na kretanje promatranog objekta i time formiraju oblik putanje. Na slici 2.1 prikazan je primjer podjele vektora smjera na prigušeni atraktor i oblikotvorni član.



Slika 2.1: Skica interakcije vektora atrakcije i oblikotvornog člana f .

Pošto se radi o učenju iz demonstracije, reformulirana je jednadžba 2.1 da koristeći poznate pomake, brzine i akceleracije u diskretnim vremenskim trenucima računa ciljne vrijednosti oblikotvornog člana f :

$$f_{target}(t) = \tau^2 \ddot{y}_{demo}(t) - \alpha_z(\beta_z(g - y_{demo}(t)) - \tau \dot{y}_{demo}(t)), \quad (2.2)$$

gdje t predstavlja diskretne vrijednosti vremenske oznake za svaku točku demonstracije.

Definicija člana f mora biti sposobna s određenim stupnjem preciznosti aproksimirati putanju objekta koja može biti kompleksna krivulja. Putanja se aproksimira koristeći obitelj otežanih radijalnih baznih funkcija u obliku Gaussiana (Fourierov aproksimator):

$$f(t) = \frac{\sum_{i=1}^N \psi_i(t) w_i}{\sum_{i=1}^N \psi_i(t)}, \quad (2.3)$$

gdje je $\Psi_i(t)$ Gaussijanska radijalna bazna funkcija te je w_i težina kojom je skalirana (ili otežana) njoj uparena radijalna bazna funkcija [5].

Ovako definiran sustav ima ključnu slabost - ovisnost o vremenu. Ako zadatak zahtjeva mijenjanje brzine izvođenja ili skaliranje prostorne dimenzije pokreta održavajući istu brzinu alata, potrebno je modificirati pristup. Za razriješiti ovisnosti sustava o vremenu moguće je zamijeniti komponentu vremena t faznom varijablom x , koja je definirana preslikavanjem linearnom funkcijom prvog reda:

$$\tau \dot{x} = -\alpha_x x. \quad (2.4)$$

Proizvoljnim odabirom x_0 , vrijednost x se monotonno približava 0 dok se t približava vremenskoj oznaci cilja. Za jednostavnost izvođenja, definirana je početna vrijednost $x_0 = 1$.

Primjenjujući zamjenu, član f redefiniran je kao funkcija faze x :

$$f(x) = \frac{\sum_{i=1}^N \psi_i(x) w_i}{\sum_{i=1}^N \psi_i(x)} x(g - y_0), \quad (2.5)$$

gdje y_0 predstavlja početnu poziciju modeliranog objekta. Konačni faktor $(g - y_0)$ je potreban zbog invarijantnosti na skaliranje prostornih dimenzija pokreta. Sukladno tome, jednadžba 2.2 može biti redefinirana da bude ovisna o fazi umjesto o vremenu:

$$f_{target}(x) = \tau^2 \ddot{y}_{demo}(x) - \alpha_z(\beta_z(g - y_{demo}(x)) - \tau \dot{y}_{demo}(x)). \quad (2.6)$$

Sljedeći je korak pronalazak vrijednosti svih težina za koje definirani član $f(x)$ prima ciljne vrijednosti $f_{target}(x)$. Ovakav je problem iznimno čest u području optimizacije te se svodi na linearnu regresiju, odnosno minimizaciju kvadrata razlike J_i dviju

funkcija, prikazanu sljedećom jednađbom:

$$J_i = \sum_{x=x_0}^{x_N} \psi_i(x) (f_{target}(x) - w_i x (g - y_0))^2. \quad (2.7)$$

Izračun svih težina izvediv je jednom vektorskom operacijom. Radijalne bazne funkcije mogu biti prikazane kao matrica u kojoj svaki stupac predstavlja ekvivalentnu normaliziranu radijalnu baznu funkciju prikazanu kao listu vrijednosti za svaku vremensku oznaku, definicija oblikotvornog člana f tada zauzima oblik vektora koji je matrični umnožak $f = \Psi w$. Jednađbu 2.7 se tada minimizira za svaku radijalnu baznu funkciju i uparenu težinu istovremeno određivanjem Moore-Penrose pseudo-inverza matrice radijalnih baznih funkcija i izračunom vektor težina $w = \Psi^+ f$ uvrštavanjem ciljnih vrijednosti oblikotvornog člana f_{target} u vektor f .

Alternativna definicija oblikotvornog člana f koja također optimizira repliciranje putanje može biti vektorska razlika između dvije slijedne ciljne vrijednosti dobivene jednađbom 2.6. Ova definicija ne zahtijeva izračun težina za replikaciju, već samo modelu prigušene opruge trivijalno zbraja razliku ciljnih vrijednosti.

2.2. Prilagodba algoritma za različite potrebe

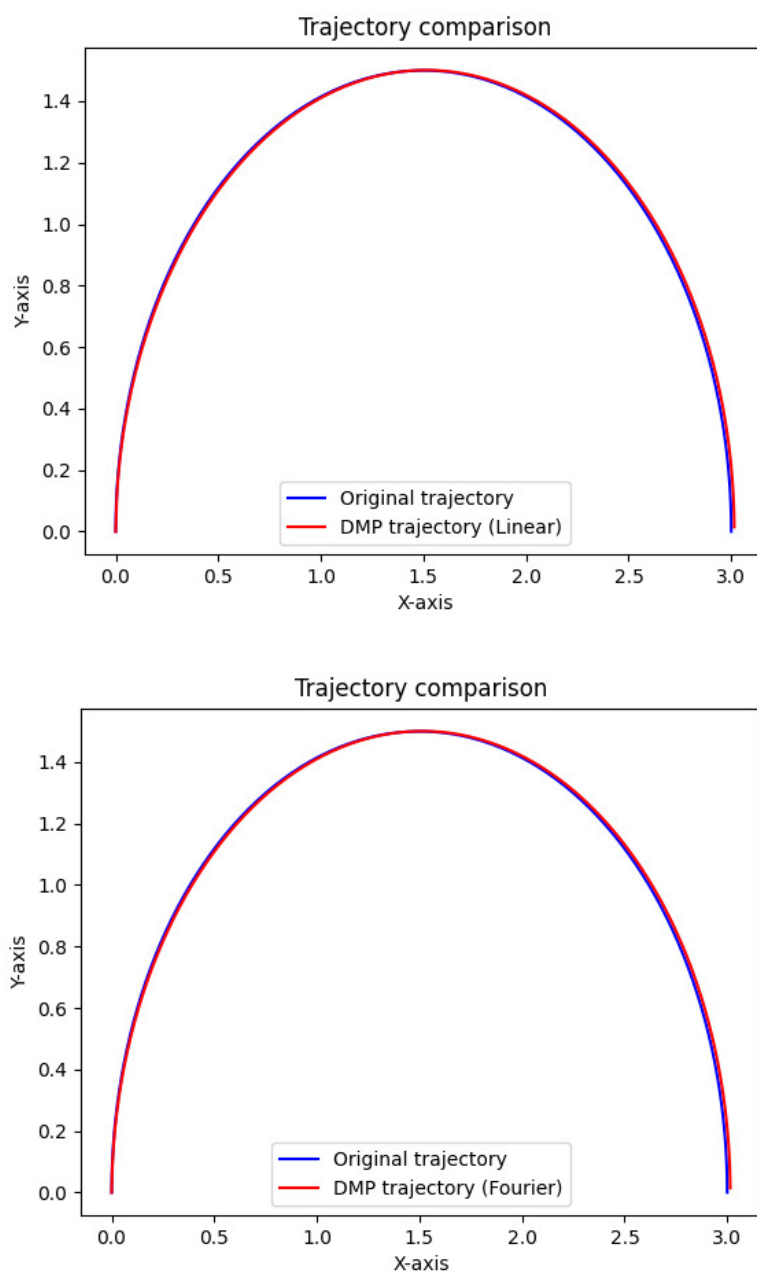
Na javnoj platformi GitHub postoji više rješenja utemeljenih na principu DMP-a te je u svrhu ovog zadatka iskorišteno i modificirano rješenje *dmp*¹. Glavna skripta ovog paketa prima listu putnih točaka kojoj pridjeljuje periodične vremenske oznake i zatim primjenjuje aproksimator za učenje demonstriranog pokreta. Unutar paketa je definirano više aproksimatora, te je po zadanim postavkama iskorišten linearni aproksimator. Uz njega je definiran i Fourierov aproksimator koji definira radijalne bazne funkcije i uparene težine prema matematičkom modelu opisanom ranije u ovom poglavlju. Jedna se ključna stvar razlikuje između opisanog algoritma i definiranog aproksimatora: nedostaje faktor nelinearnog oblikotvornog člana f koji osigurava invarijantnost na skaliranje - $(g - y_0)$.

Variranje aproksimatorima

Za ustvrditi koje rješenje će biti iskorišteno za glavni zadatak projekta, uspoređene su performanse aproksimatora na jednostavnom primjeru replikacije polukružne putanje. Na grafovima 2.2 prikazane su originalne i replicirane putanje za oba aproksimatora. Iz preklapanja putanja na oba grafa slijedi zaključak da su oba aproksimatora

¹Rješenje je dostupno na: <https://github.com/sniekum/dmp>

ekvivalentna i prikladna za 1:1 replikaciju pokreta. U daljnjim eksperimentima koristi se Fourierov aproksimator u svrhu fokusiranja eksperimenata na implementaciju opisanog matematičkog modela, te ujedno olakšanja izmjene skalirajućeg faktora.

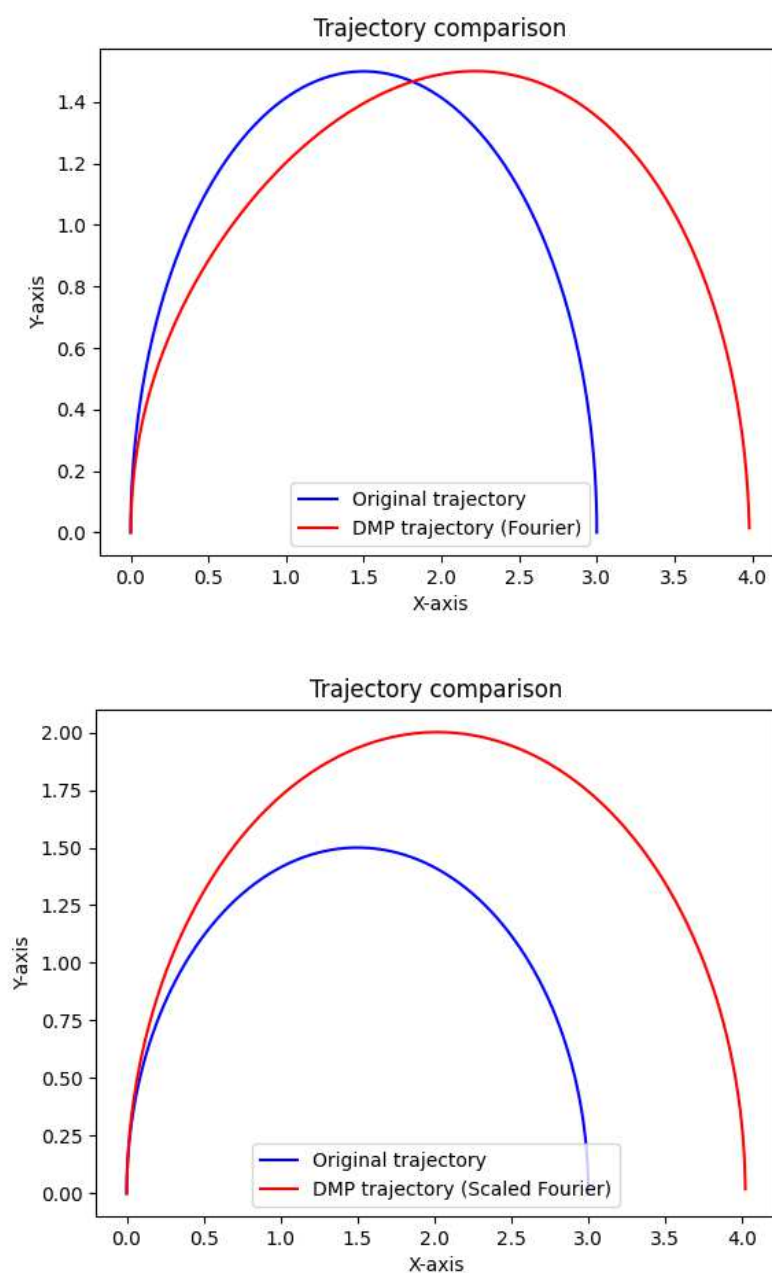


Slika 2.2: Grafovi usporedbe aproksimatora na polukružnoj putanji.

Variranje skaliranjem

Odredbom cilja koji je dalji od originalnog nastaju razlike u repliciranju zbog ranije spomenutog nedostatka skalirajućeg faktora. Grafovi na slici 2.3 uspoređuju per-

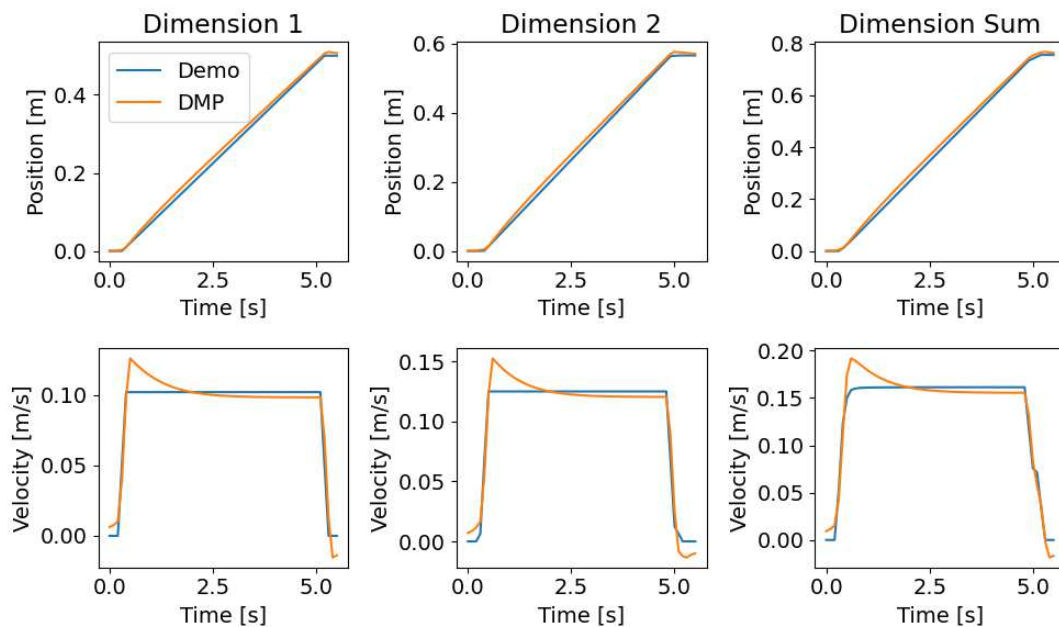
formanse zadanog Fourierovog aproksimatora i modificiranog Fourierovog aproksimatora koji uključuje skalirajući faktor. Iako je skalirani aproksimator točnija adaptacija matematičkog modela, oba imaju svoju korist. Primjerice, ako alat obavlja neki pokret s prostornom limitacijom na visinu koju alat smije doseći, pogodno je iskoristiti neskaliranu verziju, dok skalirana verzija osigurava očuvanje proporcija oblika pri imitaciji pisanog teksta. Za replikaciju se zato izabire verzija čiji su atributi prikladniji za dani zadatak.



Slika 2.3: Grafovi usporedbe skaliranja na polukružnoj putanji.

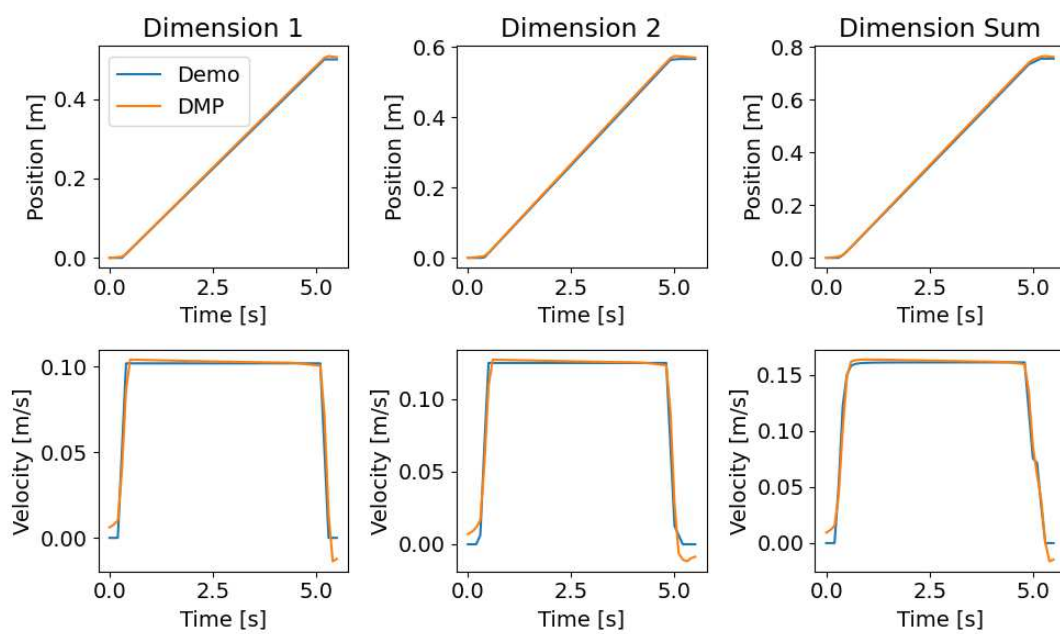
Upravljanje brzinom

Dok DMP omogućava visok stupanj preciznosti replikacije pokreta, neki zadatci koji mogu zahtijevati dodatne prilagodbe prilikom replikacije koje algoritam utemeljen na matematičkom modelu DMP-a ne pruža. Primjer zadatka koji zahtjeva jedno od takvih svojstava (upravljanje brzinom kretanja vrha alata) je bušenje čelika [6]. Bušenje čelika zahtjeva jednoliko gibanje vrha alata, u protivnom se povećava rizik sloma vrha bušilice. Iako je zadatak limitiran na brušenje, šira svrha projekta uključuje sposobnost izmjenjivanja danog zadatka te je zato dio pažnje posvećen modifikaciji algoritma kako bi se omogućilo upravljanje brzinom vrha alata. Definiranjem jednolike pravocrtno-putanje kakvu bi alat mogao obavljati tijekom bušenja i izvedbom replikaciju uz dodan prikaz brzine primjetno je da iako DMP prati oblik demonstrirane putanje, generira različite brzine. Graf pozicija i brzina alata prikazan je na slici 2.4.



Slika 2.4: Graf pozicija i brzina na pravocrtnoj putanji.

Iako sam model ne omogućava upravljanje brzinom, neki aspekti upravljanja se mogu definirati ručno. Sačuva li se maksimalna brzina kretanja vrha alata prilikom demonstracije, ona se može skalirati slično skaliranju oblikotvornog člana f u jednadžbi 2.5 i iskoristiti kao limit brzine te se u slučaju nadmašenja skalirane brzine može smanjiti brzina replicirane putanje. Slika 2.5 prikazuje rezultat izvođenja modificiranog algoritma koji limitira brzinu vrha alata tijekom replikacije.



Slika 2.5: Graf pozicija i brzina na pravocrtnoj putanji s korekcijom brzina.

3. Robotsko brušenje

3.1. Odabir površina za pokrete

Sustav treba prilagoditi zadatak kojeg obavlja samome objektu kojeg obrađuje, te zbog toga rješenje može biti drastično različito ovisno o obliku objekta. U ovom zadatku radi se o polutci kalupa za lijevanje staklene boce. Ovakvi kalupi mogu biti različitih oblika, no generalno imaju nekoliko zajedničkih obilježja koja je dobro imati na umu prilikom automatizacije zadatka:

1. Većinski konkavna i "duboka" površina

Površinu objekta je konkavna ako se u svojoj cijelosti savija prema unutra, nalik unutrašnjosti zdjele ili špilje (laički se može reći da je "udubljena"). Boce su najčešće polukružno valjkaste ili imaju presjek oblika konveksnog mnogokuta te su time njihovi kalupi konkavni. Međutim, jedno je obilježje zajedničko gotovo svim bocama te ujedno i njihovim kalupima - pošto je vrat boce uži od tijela većina boca ima grlo koje narušava konkavnost kalupa. Ovisno o vrsti obavljenog zadatka, pametniji pristup može biti razdvojiti bocu na dvije ili više odvojenih površina zbog ovog svojstva, poglavito na vrat boce i ostatak.

Specifično kod izvođenja zadataka koristeći veće alate konkavne površine imaju problem: ako su "duboke", alat kojim se izvodi zadatak možda neće moći ortogonalno pristupiti površini. Prevelikom alatu može smetati dio objekta naprotiv onog na kojem se trenutačno izvodi zadatak. Ovaj problem se dodatno pogoršava ako zadatak obavlja nefleksibilna ili široka robotska ruka te u nekim slučajevima može biti potrebno promijeniti dizajn alata i ruke ili odustati od ortogonalnosti alata, ako ona nije ključna.

2. Djeljivost na glatke površine

Površina je glatka ako duž nje nema prekida, odnosno ako je ujednačena i bez neravnina ili nepravilnosti. Isključujući fine detalje (primjerice ugravirani logotip

pića) ovo svojstvo vrijedi za jednostavne primjere kružnih ili eliptičnih presjeka boce u cijelosti, dok za boce čiji je presjek mnogokut ovo svojstvo vrijedi za odvojene plohe kalupa. Ovim je svojstvom omogućeno jednostavnije obavljanje zadataka na svakoj glatkoj površini. Ako su površine međusobno slične, kao na primjer plohe mnogokutne boce, moguće je iskoristiti sličnost između svake za izvođenje zadatka na jednoj i zatim replikaciju zadatka na svakoj sličnoj površini. Ako jedan pokret ne može pokriti cijelu površinu, jednostavno je rješenje izvesti uzastopne paralelne pokrete [7].

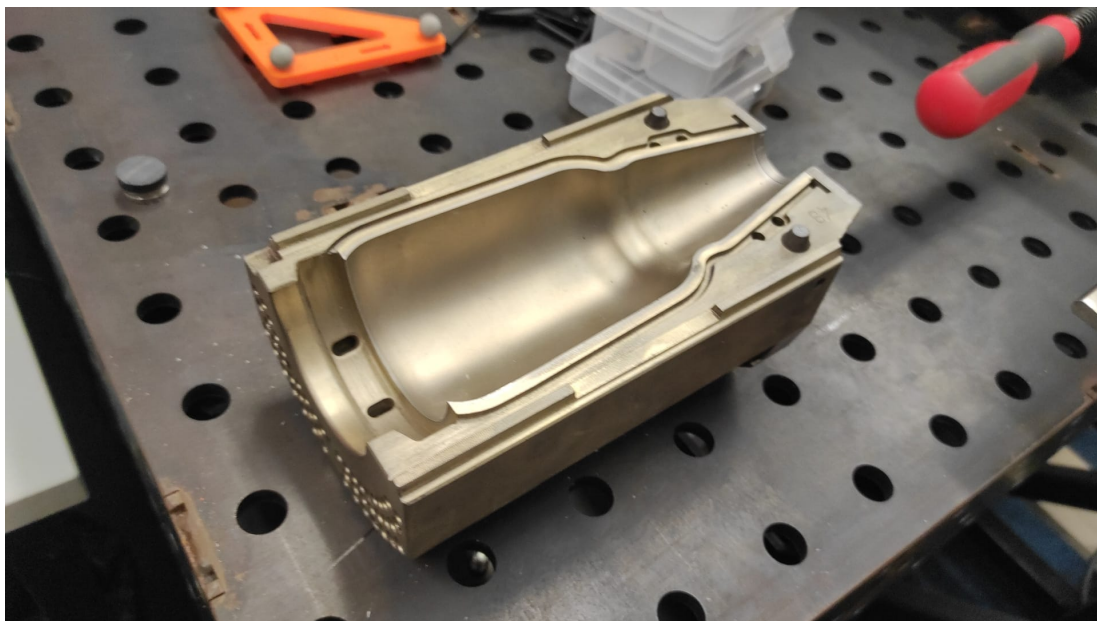
3. Simetričnost

Isključujući opet fine detalje (primjerice ugravirani logotip pića), boce su pretežito centralno ili planarno simetrične. Podjelom boce na režnjeve od vrha do dna, svaki će režanj imati dio vrata, grla i tijela boce. Uzimajući u obzir iznad navedenu djeljivost na glatke površine, moguće je dodatno pojednostaviti izvođenje zadataka segmentirajući kalup na identične ili zrcalno simetrične isječke koristeći ravnine simetrije.

Zbog ovih obilježja dobar je pristup podijeliti sve površine na režnjeve, svaki od kojih je kružni isječak kalupa od vrha do dna. Svaki režanj jedne površine je međusobno sličan te je moguće svakog brusiti zasebno uz translatiranje procesa na svaki sljedeći režanj. Postoje rješenja koja skeniraju objekt i automatiziraju planiranje putanje alata [8], no takva rješenja nisu prigodna za definirani zadatak na kalupu: koriste napredniju i skuplju opremu, obavljaju zadatak na značajno većim površinama ili ne koriste replikaciju demonstriranih pokreta.

3.2. Analiza objekta

Za ovaj rad iskorišten je kalup prikazan na slici 3.1. Boca ovakvog kalupa je savršeno zrakasto simetrična, odnosno rotacijski invarijantna - svaki presjek od vrha do dna je identičan te se razlikuje samo po rotaciji. Brušenje se dakle može daljnje pojednostaviti pokrivanjem cijele površine kalupa međusobno identičnim uniformno udaljenim pokretima brušenja od vrha do dna. Kalup se time aproksimira kao polovica krnjeg stošca.



Slika 3.1: Slika kalupa za staklenu bocu na kojem se obavlja brušenje.

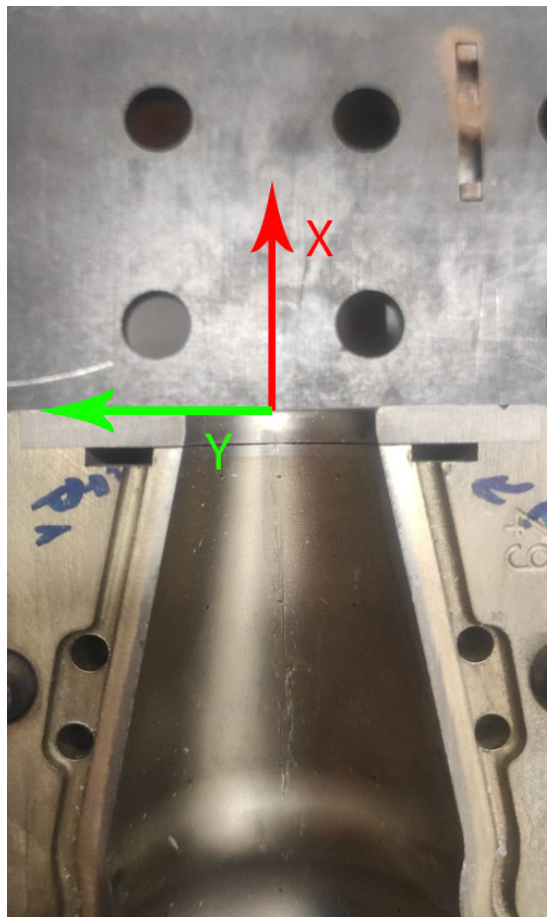
Zadatak kod brušenja je brusilicom prijeći cijelu ciljnu površinu objekta, zbog čega posebna briga treba biti posvećena da razmak između svakog pokreta brušenja ni u jednom trenutku ne bude širi od efektivne širine brusilice. Na aproksimiranom modelu stošca razmak pokreta brušenja mora biti manji ili jednak širini brusilice u dijelu koji je najudaljeniji od pravca visine stošca, što je ujedno donja polukružnica kalupa.

Za obaviti zadatak brušenja, potrebno je definirati točke u kojima se mora naći vrh brusilice u svakom trenutku brušenja. Pošto se mogu promijeniti vanjski uvjeti kao što je poza kalupa u robotovom koordinatnom sustavu, korisno je definirati koordinatni sustav kalupa te naći sve pozicije unutar njega jer tako definirane koordinate se neće mijenjati. Koordinatni sustav kalupa moguće je proizvoljno definirati, te je za svrhu ovog zadatka definiran u tablici 3.1.

Poza ishodišta koordinatnog sustava kalupa na samom kalupu prikazana je na slici 3.2.

Varijabla	Definicija
Pozicija	U centru gornje polukružnice kalupa.
X-os	U smjeru visine krnjeg stošca od donje do gornje polukružnice kalupa (ako je kalup položen, "prema naprijed").
Y-os	Definiran prema ostalim osima (ako je kalup položen, "prema lijevo").
Z-os	U smjeru suprotnom od centra kalupa (ako je kalup položen, "prema gore").

Tablica 3.1: Pregled definicija varijabli koordinatnog sustava kalupa.



Slika 3.2: Prikaz koordinatnog sustava na kalupu.

Uz definiranu širinu brsilice (*tool_width*) i polumjer polukružnice kalupa (*radius*), za pronaći relativnu poziciju početne i krajnje točke svakog pokreta iskorišten je sljedeći algoritam:

1. Broj pokreta

Minimalan broj pokreta je jednak omjeru polovice opsega kružnice danog polumjera i širini brusilice (zaokružen prema gore):

$$\begin{aligned} effective_length &= \pi * radius, \\ num_strokes &= \lceil \frac{effective_length}{tool_width} \rceil. \end{aligned} \quad (3.1)$$

2. Početni kut te odmak kuta

Početni i završni pokret trebaju biti maksimalno polovicu širine brusilice udaljeni od ruba površine te svaki kut između treba biti udaljen maksimalno za jednu širinu brusilice. Uz definiciju polukružnice između kuteva $-\frac{\pi}{2}$ i $\frac{\pi}{2}$, početni kut ($start_angle$) i pomak kuta ($angle_step$) matematički mogu biti definirani sljedećim jednažbama:

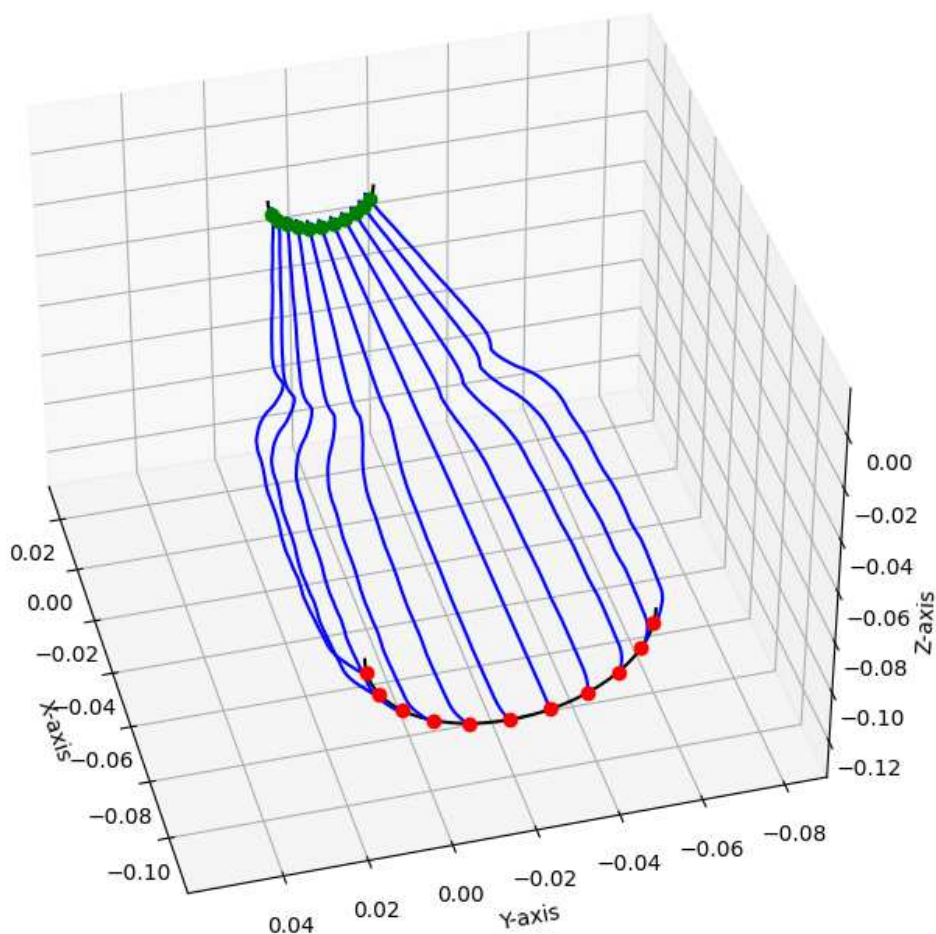
$$\begin{aligned} angle_step &= \frac{\pi}{num_strokes}, \\ start_angle &= \frac{angle_step - \pi}{2}. \end{aligned} \quad (3.2)$$

3. Relativne pozicije

Koristeći kut svakog pokreta mogu se izračunati relativne pozicije u koordinatnom sustavu kalupa, gdje je x-os smjer visine stošca koji se koristio za aproksimaciju kalupa i z-os smjer nasuprot sredine kalupa:

$$\begin{aligned} \theta_i &= start_angle + i * angle_step, \\ x_i &= 0, \\ y_i &= -radius * \sin(\theta), \\ z_i &= -radius * \cos(\theta). \end{aligned} \quad (3.3)$$

Ovaj je algoritam primijenjen za izračun relativnih pozicija svakog pokreta za gornju i donju polukružnicu stošca (koristeći odvojene polumjere) uz posmicanje koordinata donjeg u x-osi (negativna vrijednost). Pozicije na gornjoj polukružnici predstavljaju početnu točku pokreta, dok pozicije na donjoj polukružnici predstavljaju završnu točku pokreta. Vizualizacija točaka prikazana je na slici 3.3.



Slika 3.3: Prikaz ključnih točaka povezanih estimiranim oblikom kalupa.

Preostaje još dodati sami pokret između tih točaka te po potrebi definirati dodatne točke kretanja. U ovom su projektu definirane pomoćne "lebdeće" točke koje su blago okomito odmaknute od površine kalupa koje služe za neometano premještanje vrha alata u sljedeću početnu točku. One su definirane na isti način kao i početne i završne točke, uz skalirane y i z vrijednosti da budu bliže centru rotacije. Alternativno, mogu se definirati "lebdeće" točke u centru rotacije.

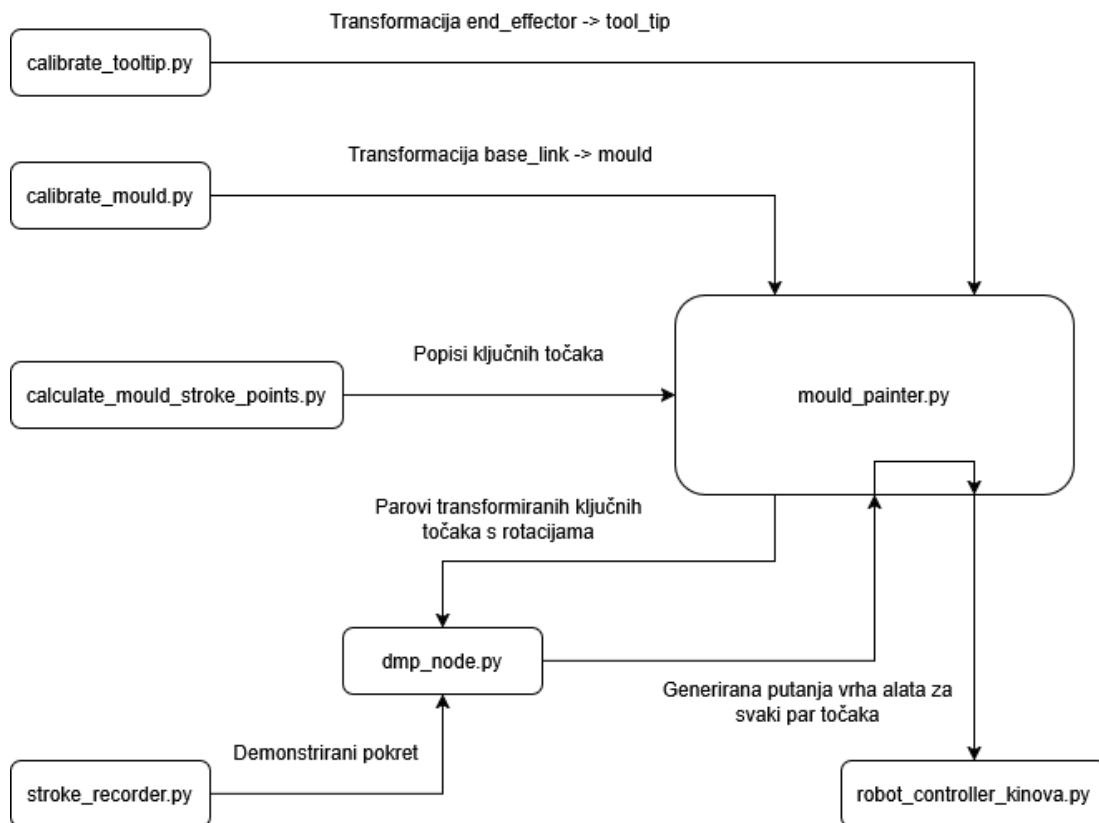
3.3. Programska arhitektura

Programska podrška razvijena za sustav sadržana je u paketu "*semicylPainter*". Skripte su pisane u jeziku Python3 te je njihov opis prikazan u tablici 3.2.

Ime Skripte	Opis
mouldPainter.py	Glavni program koji koristi ostale skripte.
calculate_mould_stroke_points.py	Skripta koja definira početne, završne i pomoćne točke svakog pokreta brušenja.
stroke_recorder.py	Skripta koja služi za snimanje i spremanje putanje pokreta brušenja koji se kasnije koristi za replikaciju.
robot_controller_kinova.py	Skripta koja izravno upravlja robotskom rukom koristeći MoveIt.
dmp_node.py	Skripta koja računa putanju pokreta brušenja koristeći snimljenu putanju alata (uz potrebne modifikacije).
calibrate_tooltip.py	Skripta koja služi za kalibraciju transformacije iz koordinatnog sustava baze robota u koordinatni sustav kalupa.
calibrate_mould.py	Skripta koja služi za kalibraciju transformacije iz koordinatnog sustava prihvatnice alata robota u koordinatni sustav vrha alata (brusilice).

Tablica 3.2: Pregled skripti i njihovih funkcija.

Interakcija skripti detaljnije je prikazana na slici 3.4.



Slika 3.4: Graf interakcije skripti programske podrške paketa semicylPainter.

U sljedećim će poglavljima zadatci kojima se ove skripte bave biti detaljnije objašnjeni. Osim podrške razvijene za ovaj zadatak, sustav koristi dodatne pakete - *rf_loc*¹, *dmp*², *ros_kortex*³ i *vrpn_client_ros*⁴. Paket *rf_loc* služi za lokalizaciju poze kalupa, *dmp* definiranja trajektoriju vrha alata prema ranije opisanom matematičkom modelu, *ros_kortex* upravlja simuliranom ili stvarnom robotskom rukom koristeći paket MoveIt i *vrpn_client_ros* služi za komunikaciju s Motiveom za snimanje demonstrirane poze.

3.4. Demonstracija pokreta

Za izvođenje pokreta brušenja programska podrška sustava treba način dohvatiti informacije o demonstriranom pokretu kojeg replicira. Pošto se radi o pokretu čiji je oblik od ključne važnosti i veoma specifičan, jedan od jednostavnijih načina za spre-

¹Detaljnije objašnjenje u potpoglavlju "kalibracija"

²Rješenje je dostupno na: <https://github.com/sniekum/dmp>

³Rješenje je dostupno na: https://github.com/Kinovarobotics/ros_kortex

⁴Više na: https://wiki.ros.org/vrpn_client_ros

manje informacija o pokretu je slijedna lista točaka u kojima se nalazi vrh alata tijekom izvođenja pokreta.

Snimanje poze vrha alata brusilice ostvareno je korištenjem alternativnog modela alata koji omogućava precizno praćenje njegove putanje u prostoru: umjesto stvarne brusilice, koristi se specijalan alat opremljen reflektivnim optičkim markerima. Ova metoda omogućava precizno bilježenje pokreta bez poteškoća koje proizlaze iz rada s pravom brusilicom. Slika 3.5 prikazuje alat s reflektivnim markerima tijekom snimanja pokreta.



Slika 3.5: Slika specijalnog alata s optičkim markerima u procesu demonstracije brušenja po kalupu.

Poza alata koju snima sustav kamera OptiTrack Trio izračunata je kao centroid svih optičkih markera u alatu Motive, dok je orijentacija definirana prilikom označavanja skupa markera kao jedno tijelo relativno na označeni marker ravnine osnovice (ground plane). Poza markera ravnine osnovice je proizvoljna, stoga je moguće izabrati pozu koja pojednostavljuje kasniji posao. Izborom poze koja je iste orijentacije kao baza

robota ili zakrenuta za 90 stupnjeva eliminirana je briga o transformacijama između koordinatnog sustava kamere, odnosno ravnine osnovice i baze robota - može se koristiti jednostavna rotacija za izračun orijentacije pokreta u koordinatnom sustavu baze robota.

Svi ROS paketi su pokrenuti u operacijskom sustavu Ubuntu 20.04.6 LTS, dok je Motive namijenjen za pokretanje u operacijskom sustavu Windows. Kako bi se sustav mogao pokrenuti na jednom računalu jedan od operacijskih sustava može se pokrenuti unutar virtualnog stroja. Motive omogućava streamanje poza svih označenih tijela preko protokola VRPN periodičnim slanjem trenutačne poze. Na Linux sustavu se tada VRPN poruke mogu prevesti u ROS poruke koristeći paket *vrpn_client_ros*⁵, gdje poza svakog tijela zauzima svoju ROS temu. Jednostavnom se skriptom može prisluškivati ROS tema koja ispisuje poze alata, translirati poze kako bi prikazale pozu vrha alata umjesto centroida optičkih markera i sačuvati pozicije u tekstualnoj datoteci. Označivanjem optičkih markera specijalnog alata kao jedno tijelo u Motive-u dok je ortogonalan na ravninu osnovice, izračun translacije pojednostavljen je na pomak isključivo duž jedne osi.

Translacija poza je izračunata umjeravanjem dimenzija specijalnog alata pomičnom mjerkom, te iznosi -10 cm duž z-osi. Translacija duž z-osi može se prikazati u obliku transformacijske matrice:

$$T_{centroid}^{tool_tip} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -0.1 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.4)$$

Za izračun matričnog oblika poze vrha alata koristi se matrični umnožak dvaju transformacijskih matrica. Pošto se radi o lokalnoj transformaciji, odnosno transformaciji unutar vlastitog koordinatnog sustava, matrica $T_{centroid}^{tool_tip}$ nalazi se s lijeve strane umnoška. Umnožak matrica prikazan je u sljedećoj jednadžbi:

$$T_{ground_plane}^{tool_tip} = T_{centroid}^{tool_tip} * T_{ground_plane}^{centroid}, \quad (3.5)$$

gdje je $T_{ground_plane}^{centroid}$ transformacijska matrica koja predstavlja pozu centroida optičkih markera i $T_{ground_plane}^{tool_tip}$ predstavlja transformacijsku matricu poze vrha alata.

Za dodatno pojednostavljenje snimljeni se pokret može izgladiti eliminacijom varijacije snimljenih pozicija u y-osi (odnosno lijevo-desno), jer je jedan pokret brušenja izveden duž x-osi uz variranje visine, time razlike u y-vrijednosti nisu važne za sam pokret i proizlaze samo iz nepreciznosti ljudskog pokreta.

⁵ Više na: https://wiki.ros.org/vrpn_client_ros

3.5. Upravljanje robotom

Upravljanje robotom je implementirano koristeći programski paket MoveIt uz već gotovo rješenje unaprijedne i inverzne kinematike za robotsku ruku Kinova Gen3 po imenu *ros_kortex*, potrebno je samo definirati metapodatke za *moveit_commander* objekt te je preko istog moguće izravnim naredbama upravljati robotom u Kartezijskom prostoru ili prostoru zglobova. Sustav opisan u ovom radu planira gibanje u Kartezijskom prostoru primajući listu poza koje robot zatim slijedno izvrši.

Svaki put kada skripta **robot_controler_kinova.py** primi listu poza, one prvo trebaju biti prevedene iz poza namijenjenih za koordinatni sustav vrha alata u poze namijenjene za koordinatni sustav prihvatnice alata robotske ruke jer *moveit_commander* koristi kinematiku ruke bez hvataljke ili dodatnog alata. Za prijevod poza potrebno je kalibrirati pozu vrha alata i iskoristiti inverz dobivene transformacijske matrice za prevođenje poza.

Dalje se koristi ugrađena funkcija objekta *moveit_commander* koja računa slijed putnih točaka koje tjeraju prihvatnicu alata da se kreće u ravnim segmentima koji slijede ranije definirane i prevedene poze u Kartezijskom prostoru. Ovom funkcijom iz putanje definirane u Kartezijskom prostoru dobiva se planirana trajektorija definirana u prostoru zglobova robotske ruke. Ovu se trajektoriju može izravno izvršiti ili dodatno modificirati.

Izvršavanje modificirane trajektorije neće nužno uvijek biti moguće preko alata MoveIt, bilo radi neispravne kronologije poza u trajektoriji, radi prevelike potrebne brzine nekog zgloba robotske ruke, radi mehaničkih poteškoća kretanja ruke ili raznih drugih razloga. Izračun i izvršavanje trajektorije je zato stavljen u petlju koja u slučaju nemogućnosti izvođenja mijenja određene parametre kako bi se potencijalno razriješio zastoј.

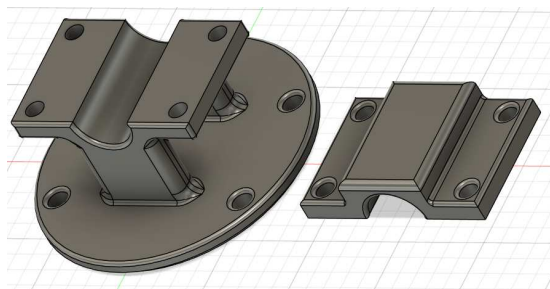
4. Eksperimentalna provjera

4.1. Dizajn nosača za alat

Robotska ruka Kinova Gen3 prema zadanim postavkama koristi Robotiq 2f-85 ili 2f-140 hvataljke. Ove su hvataljke praktične za hvatanje i prijenos objekata, no ne mogu stabilno držati alat kao što je brusilica. Srećom, lako ih je zamijeniti vlastitim nosačem alata jer su za prihvatnicu alata robotske ruke pričvršćene standardiziranim M5 vijcima.

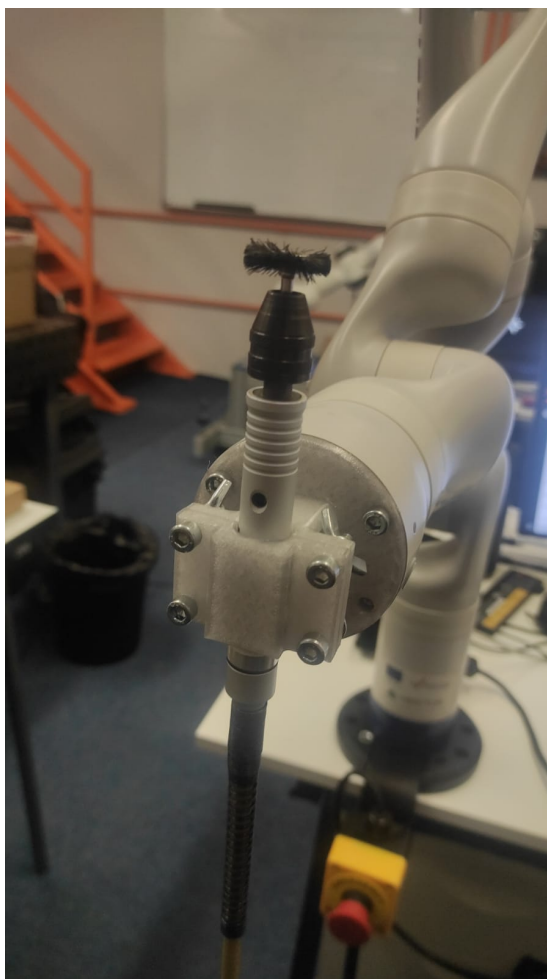
Brusilica korištena za replikaciju brušenja je zapravo precizna električna bušilica s nastavkom za brušenje uz kabel s neovisnom rotacijom od vrha alata kako bi aktivni dio alata bio odvojen od glavnog dijela brusilice za lakše korištenje. Na kraju tog kabela se nalazi valjkasti držač s rupom koja služi za zaglavljivanje rotacije unutarnjeg alata kako bi se zamijenio njegov nastavak na vrhu. Ključno je dakle da je držač kojim robotska ruka upravlja oblika tankog valjka s malom rupom za zaglavljivanje, te je potrebno dizajnirati nosač koji može prihvatiti držač i osigurati neometani pristup rupi za zaglavljivanje.

Donji dio nosača mora biti prilagođen prihvatnici alata robota kako bi se mogao ispravno pričvrstiti koristeći 4 M5 vijaka, dok gornji dio nosača mora čvrsto držati ranije opisani držač alata. Jednostavan dizajn gornjeg dijela nosača je kružna rupa jedva šira od držača kroz koju se može provući držač, uz limitiranu širinu nosača zbog zahatjeva pristupačnosti rupe za zaglavljivanje. Slika 4.1 prikazuje 3D model dizajniranog nosača. Za jednostavniji pristup nosač je razdvojen u dva dijela oko utora za alat koji se također povezuju koristeći 4 M5 vijka i matice. U svrhu neometanog pristupa vijaka za učvršćivanje nosača za prihvatnicu alata robotske ruke, valjkasti utor je odmaknut prema vani na dva segmenta jer je držač alata širi od kraće stranice pravokutnika koji definira pozicije utora za pričvršćenje nosača za prihvatnicu alata robotske ruke.



Slika 4.1: Slika 3D modela nosača.

Uz predviđenu instalaciju alata, nosača i ruke, alat je orijentiran u smjeru Y-osi preddefiniranog koordinatnog sustava prihvatnice alata robotske ruke, s pomakom koji ovisi o poziciji držača alata i vrha alata prilikom pričvršćivanja. Za ovaj projekt je alat pričvršćen da pomak vrha u smjeru Y-osi iznosi 10 cm. Konačna iteracija nosača s alatom prikazana je na slici 4.2, pričvršćena na robotsku ruku.



Slika 4.2: Slika nosača alata pričvršćenog na kraj robotske ruke, s nastavkom za brušenje.

Za drukčije alate ovakav dizajn možda ne bi bio pogodan jer bi pod većim silama

držač alata mogao klizati duž utora, no za ručnu brusilicu sila trenja pruža dovoljan otpor. Ipak, za bolje osiguranje je oko metalnog držača oblijepljen sloj ljepljive trake koja pruža dodatno trenje.

4.2. Kalibracija

Kako bi robotska ruka mogla ispravno brusiti koristeći novi vrh, potrebno je umjeriti sustav i odrediti točne dimenzije dodanog alata.

U programskom alatu MoveIt definirano je već gotovo rješenje unaprijedne i inverzne kinematike za robotsku ruku Kinova Gen3 po imenu *ros_kortex*. Pošto nema nikakve dodatne akcije koja mijenja pozu vrha, definicija poze vrha alata svodi se na statičnu transformaciju iz koordinatnog sustava prihvatnice alata robota u novodefinirani koordinatni sustav vrha alata brusilice.

Izgled nosača već sugerira promjenu u orijentaciji za alate kao što je bušilica, no brusilica ima drukčiji oblik - vrh brusilice je kružna četka. Održavanje orijentacije kakvu ima prihvatnica alata osigurava da je kut pod kojim bi brusilica trebala biti usmjerena ujedno i kut za kojeg je koordinatni sustav vrha alata ortogonalan na podlogu. Definicija se transformacije time svodi samo na translaciju.

Vrh alata se kasnije koristi za kalibraciju kalupa te je od ključne važnosti da transformacija alata bude što preciznija. Za precizniju kalibraciju može se iskoristiti ranije spomenuta rupa za zaglavljivanje u svrhu zamjene vrh alata - četku za brušenje čiji je vrh teško identificirati lako je zamijeniti bušilicom ili sličnim nastavkom s igličastim vrhom. Ovim se stvara dodatan korak: definicija dodatne operacije koja dodaje pomak za četku koja se koristi tijekom brušenja, ali ne tokom kalibracije.

Standardni pristup kalibriranja poze je sakupljanje veće količine poza koje imaju istu poziciju u koordinatnom sustavu vrha alata zapisane u koordinatnom sustavu prihvatnice alata te pronalazak translacije koja sve ranije sakupljene poze preslikava u istu poziciju uz minimalnu grešku. Tražena translacija može biti prikazana kao transformacijska matrica oblika:

$$T_{end_effector}^{tool_tip} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (4.1)$$

gdje t_x , t_y i t_z predstavljaju vrijednosti pomaka duž 3 Kartezijske osi. Nadalje, poza vrha alata također može biti prikazana kao transformacijska matrica $T_{base_link}^{tool_tip}$. Tada je

moguće definirati matični umnožak koji predstavlja transformaciju u lokalnom koordinatnom sustavu:

$$T_{base_link_i}^{tool_tip} = T_{base_link_i}^{end_effector} * T_{end_effector}^{tool_tip}, \quad (4.2)$$

gdje je $T_{base_link_i}^{tool_tip}$ transformacijska matrica koja predstavlja pozu vrha alata, $T_{base_link_i}^{end_effector}$ predstavlja pozu prihvatnice alata robotske ruke i konačno i predstavlja indeks poze. Ukoliko dvije poze imaju jednaku poziciju, u konačnoj transformaciji njihov posljednji stupac matrice $T_{base_link_i}^{tool_tip}$ mora biti jednak. Pronalazak transformacije se dalje može prevesti u optimizacijski problem pronalaska ranije definiranih pomaka za minimizaciju razlike matrica za svaki par poza [9]:

$$J_{ij} = T_{base_link_i}^{tool_tip} - T_{base_link_j}^{tool_tip}, \quad (4.3)$$

gdje $i \neq j$. Ovaj problem optimizacije razriješen je koristeći funkciju *fmin* iz Python3 paketa *scipy.optimize*.

Konačna se transformacija tada emitira kao ROS TF2 okvir ili sprema u tekstualnu datoteku za kasnije korištenje.

Koordinate točaka koje vrh alata treba dostići zadane su u koordinatnom sustavu kalupa te je potrebno naći iste u koordinatnom sustavu baze robota. Za odrediti transformaciju iz koordinatnog sustava baze robota u koordinatni sustav kalupa, iskorišten je sličan pristup kao za kalibraciju poze vrha alata: vrh alata je doveden u četiri a priori uparene točke kalupa u vlastitom koordinatnom sustavu i zatim izračunata statična transformacija koja minimizira grešku transformacije jednog skupa točaka u drugi [10, str. 130]. Ovu kalibraciju obavlja dodatni paket *rf_loc* čija glavna skripta dohvaća točke preko ROS teme *"/mould_pose_array"*. Rezultat izvođenja je tražena transformacija zapisana u obliku transformacijske matrice koje se sprema u tekstualnu datoteku i dohvaća po potrebi emitiranjem kao ROS TF2 okvir.

4.3. Simulacija robotskog brušenja

Prije simulacije izvođenja ciljanog zadatka, slijedi kratak opis slijeda izvođenja skripti definiranog sustava uz prethodnu kalibraciju i snimanje pokreta:

- 1) Skripta **calculate_mould_stroke_points.py** generira ključne točke u koordinatnom sustavu kalupa koje dohvaća glavni program **mouldPainter.py**.
- 2) Za svaki par ključnih točaka, glavni program **mouldPainter.py** definira i kut rotacije pokreta za ispravno izvođenje, transformira pozicije kako bi se ispravno

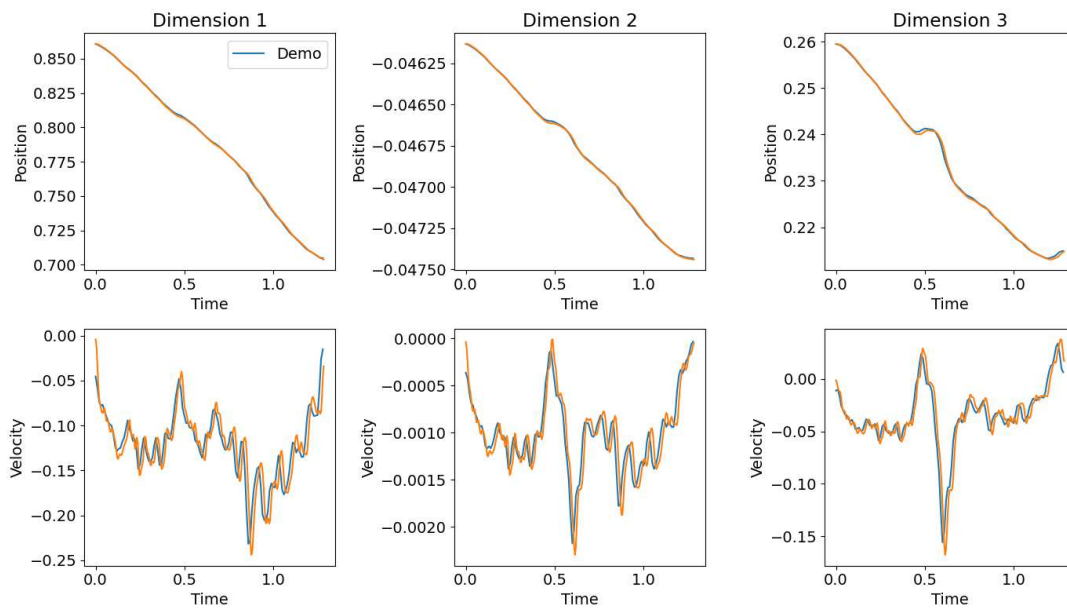
izvele s dodanim vrhom alata u koordinatnom sustavu baze robota, pridjeljuje svakoj točki orijentaciju te pokreće glavnu metodu skripte **dmp_node.py**.

- 3) Glavna metoda skripte **dmp_node.py** dohvaća, rotira i uglađuje snimljenu putanju pokreta i vraća prilagođenu putanju.
- 4) Glavni program **mouldPainter.py** svaku prilagođenu putanju šalje skripti za upravljanje robota **robot_controller_kinova.py** za izvođenje na simuliranoj ili pravoj robotskoj ruci.

Skripta **calculate_mould_stroke_points.py** pruža više funkcionalnosti od ranije objašnjenog algoritma pronalaska pozicije početnih i krajnjih točaka pokreta. Umjesto preddefiniranih polumjera i odnosa donje polukružnice, koriste se 3 točke: početna i završna točka lijeve strane kalupa i završna točka desne strane kalupa. Iako takav pristup sadrži redundantne informacije, omogućava redefiniciju centra koordinatnog sustava kalupa ili skaliranje dimenzija kalupa promjenom pozicija definiranih točaka. Također su podržane opcije dodavanja preklopa između pokreta brušenja i zadebljanja (padding) kalupa. Točke koje algoritam generira nisu samo pozicije već poze - sadrže poziciju i orijentaciju svake od generiranih točaka kako bi se omogućilo jednostavnije izvođenje zadataka koji zahtijevaju ortogonalnost alata, pošto se orijentacijom može naznačiti u kojem će smjeru alat biti ortogonalan na površinu. Ova je funkcionalnost u glavnom programu neiskorištena i zamijenjena uniformnim orijentacijama jer robotska nije sposobna osigurati ortogonalnost izvođenja zadatka zbog problema kojeg uvodi većinski konkavna i "duboka" površina kalupa. U konačnici se izračunate poze šalju ostatku sustava koristeći ROS Publishere, gdje ih preuzima glavni program.

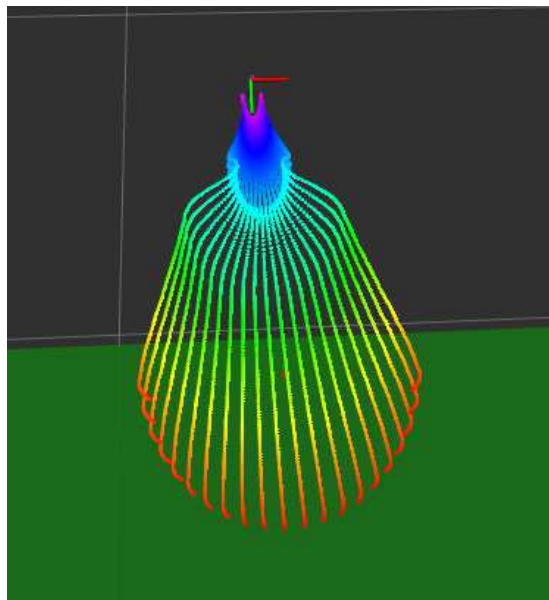
Glavni program primjenjuje uniformnu orijentaciju svakoj pozi, umjesto da je z-os uperena ravno u podlogu primijenjena je dodatna rotacija 45 stupnjeva oko x-osi. Razlog ove rotacije je osiguravanje neometanog pristupa robotske ruke zadanoj točki.

Algoritam DMP je primijenjen za replikaciju pokreta brušenja. Na slici 4.3 prikazani su grafovi pomaka i brzine kretanja vrha alata u demonstriranom i repliciranom pokretu. Za replikaciju je iskorištena varijanta algoritma DMP s Fourierovim aproksimatorom i limitacijom brzine, no svaka varijanta daje isti rezultat - skaliranje nema utjecaj jer nije promijenjena udaljenost cilja.



Slika 4.3: RViz vizualizacija simulacije izvođenja demonstriranih pokreta brušenja na kalupu.

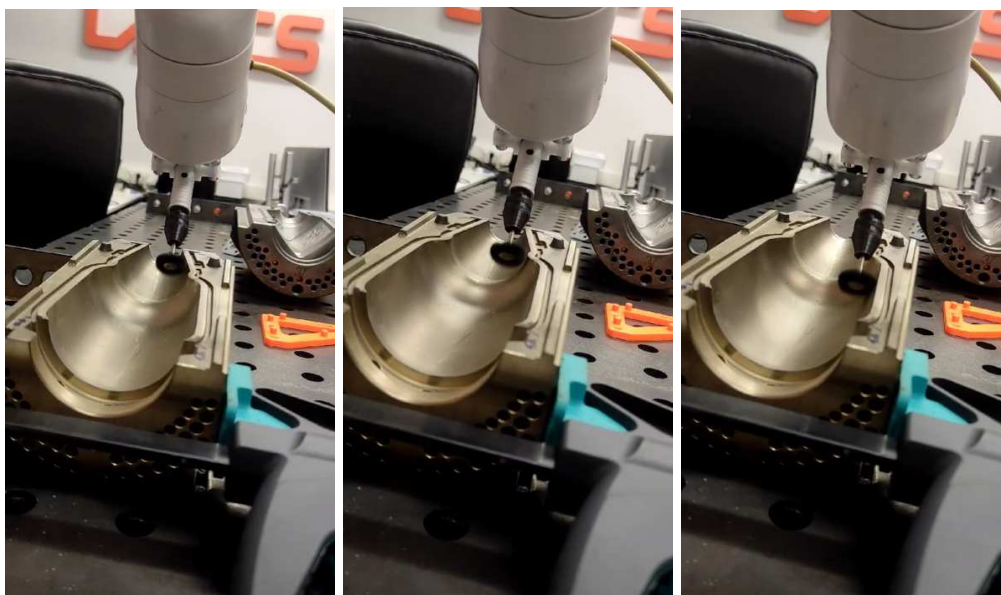
Slika 4.4 prikazuje rezultat simuliranja brušenja uz vizualizaciju svakog pokreta. Oblik presjeka kojeg stvaraju putanje je eliptičan zbog dodatne translacije koju prouzrokuje četka brusilice - iako je kalup polukružnog presjeka, putanje će prolaziti bliže sredini sa strana i bliže donjoj polukružnici kalupa pri sredini.



Slika 4.4: RViz vizualizacija simulacije izvođenja demonstriranih pokreta brušenja na kalupu.

4.4. Praktična izvedba zadatka

Tri su glavne metrike uspjeha praktičnog izvođenja: prva je uspješno repliciranje demonstriranog pokreta brušenja, druga je uspješno pozicioniranje i praćenje putanja bez sudara alata s kalupom i treća je uspješno pokriće cijele unutarnje površine. Prva metrika je zadovoljena, što je vidljivo iz preklapanja demonstriranog i repliciranog pokreta na slici 4.3. Druga metrika je u potpunosti zadovoljena - robotska ruka uspješno dovodi vrh alata u svaku zadanu točku bez sudara s kalupom. Time se može reći da je kalibracija dovoljno precizna i slijed izvođenja dovoljno konzistentan. Međutim, nije u potpunosti zadovoljena treća metrika - iako su svi pokreti konzistentni, vrh brusilice se odvaja od kalupa kod grla i opet dolazi s njim u kontakt pri drugoj polovici trupa boce. Slika 4.5 prikazuje izvođenje pokreta brušenja s vidljivim odmakom brusilice od desne površine kalupa. Uzimajući u obzir da su početak i kraj svakog pokreta ispravni i da je replikacija demonstriranog pokreta uspješna s dovoljnim stupnjem preciznosti, neispravnost pokreta može proizlaziti iz samog snimanja pokreta specijalnog alata s optičkim markerima.



Slika 4.5: Odvajanje brusilice od desne strane kalupa kod grla.

5. Zaključak

Za replikaciju ljudski demonstriranog pokreta brušenja definiran je programski paket *semicyl_painter* koji preko platforme ROS upravlja robotskom rukom na koju je pričvršćen alat s vrhom za brušenje. Paket koristi implementaciju matematičkog modela primitiva dinamičkog gibanja (DMP) za učenje i repliciranje krivulje pokreta brušenja. Implementacija algoritma DMP je dodatno modificirana kako bi bila pogodnija za korištenje u industrijske svrhe limitiranjem brzine kretanja vrha alata tijekom replikacije pokreta. Pokret brušenja snimljen je sustavom kamera koji prati kretanje specijalnog alata označenog reflektivnim optičkim markerima i prilagođen robotskom izvođenju. U svrhu korištenja alata je za robotsku ruku dizajniran nosač za alat koji se može pričvrstiti na prihvatnicu alata robotske ruke. Sustav koristi kalibraciju kako bi efektivno umjerio pozu vrha alata na nosaču i koristeći kalibriranu pozu vrha alata umjerio pozu kalupa kojeg treba brusiti u radnom prostoru.

Robotska ruka uspješno upravlja brusilicom i prati definirane pokrete brušenja utemeljene na demonstraciji. Precizno pozicioniranje alata je uspješno izvedeno kalibracijom poze vrha alata. Pokreti brušenja pravilno su pozicionirani unutar kalupa, indicirajući uspjeh kalibracije poze kalupa.

Međutim, sustav gubi kontakt s kalupom prilikom izvođenja zadatka što se može dogoditi ako je snimka pokreta brušenja neispravna ili loše prilagođena. Ovaj sustav može se poboljšati ispravljanjem neispravnog ponašanja replikacije pokreta, što se može postići boljom integracijom snimanja demonstriranog pokreta s ostatkom sustava ili korištenjem adaptivne metode upravljanja, primjerice sa senzorom sile, koja može utjecati na trajektoriju izvedbe. Alternativno se rad može prilagoditi modifikacijom matematičkog modela kakva bi omogućila liberalnije upravljanje dodatnim parametrima izvedbe kao što je brzina kretanja vrha alata i konačno prilagodbom sustava za druge zadatke slične brušenju.

LITERATURA

- [1] Peter Pastor, Heiko Hoffmann, Tamim Asfour, i Stefan Schaal. Learning and generalization of motor skills by learning from demonstration. U *2009 IEEE International Conference on Robotics and Automation*, stranice 763–768, 2009. doi: 10.1109/ROBOT.2009.5152385.
- [2] Auke Jan Ijspeert, Jun Nakanishi, Heiko Hoffmann, Peter Pastor, i Stefan Schaal. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation*, 25(2):328–373, 2013.
- [3] Peter Pastor, Mrinal Kalakrishnan, Sachin Chitta, Evangelos Theodorou, i Stefan Schaal. Skill learning and task outcome prediction for manipulation. U *2011 IEEE international conference on robotics and automation*, stranice 3828–3834. IEEE, 2011.
- [4] Pedro La Hera, Daniel Ortiz Morales, i Omar Mendoza-Trejo. A study case of dynamic motion primitives as a motion planning method to automate the work of forestry cranes. *Computers and Electronics in Agriculture*, 183:106037, 2021. ISSN 0168-1699. doi: <https://doi.org/10.1016/j.compag.2021.106037>. URL <https://www.sciencedirect.com/science/article/pii/S0168169921000557>.
- [5] David Lowe i D Broomhead. Multivariable functional interpolation and adaptive networks. *Complex systems*, 2(3):321–355, 1988.
- [6] Bruno Marić, Frano Petric, Dario Stuhne, Vanja Ranogajec, i Matko Orsag. Replicating human skill for robotic deep-micro-hole drilling. U *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, stranice 2238–2244, 2022. doi: 10.1109/CASE49997.2022.9926630.
- [7] Hong Zhao, Ke Wen, Tianjian Lei, Yinan Xiao, i Yang Pan. Automatic aluminum alloy surface grinding trajectory planning of industrial robot based on weld seam

- recognition and positioning. *Actuators*, 12(4), 2023. ISSN 2076-0825. doi: 10.3390/act12040170. URL <https://www.mdpi.com/2076-0825/12/4/170>.
- [8] Benjamin Alt, Florian Stöckl, Silvan Müller, Christopher Braun, Julian Raible, Saad Alhasan, Oliver Rettig, Lukas Ringle, Darko Katic, Rainer Jäkel, et al. Robogrid: Intuitive and interactive surface treatment with industrial robots. *arXiv preprint arXiv:2402.16542*, 2024.
- [9] Bruno Marić, Marsela Polić, Tomislav Tabak, i Matko Orsag. Unsupervised optimization approach to in situ calibration of collaborative human-robot interaction tools. U *2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, stranice 255–262, 2020. doi: 10.1109/MFI49285.2020.9235229.
- [10] Richard Hartley i Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

Demonstracijom podržano robotsko brušenje

Sažetak

Ovaj rad se bavi robotskom replikacijom ljudski demonstriranog pokreta brušenja metalnog kalupa za izradu staklenih boca. Ljudski pokret je sačuvan kao skup pozicija alata kojeg zatim nauči, modificira i replicira implementacija matematičkog modela primitiva dinamičkog gibanja (dynamic movement primitives, DMP). Ispitane su performanse različitih verzija algoritma DMP. Za snimanje pokreta je iskorišten specijalni sustav kamera koji prati pozicije optičkih markera. U svrhu replikacije brušenja je dodatno izrađen nosač za alat koji je pričvršćen na robotsku ruku i provedena kalibracija poze vrha alata i kalibracija poze kalupa.

Ključne riječi: Robotizacija, Robotsko Brušenje, Robotska Manipulacija, Kalibracija Poze, Demonstracijom Podržano Robotsko Učenje, Replikacija Pokreta, DMP, Primitivi Dinamičkog Gibanja

Demonstration-supported robotic sanding

Abstract

This thesis centers around the robotic replication of human-handled demonstrated sanding movements of a metal mould designed for making glass bottles. Human-handled tool movement is saved as a set of tool positions which is then learned, modified and replicated by the implementation of a mathematical model of dynamic movement primitives (DMP). The performance of different versions of the DMP algorithm was tested. A special camera system was used to record the movements, which tracked the positions of optical markers. For the purposes of sanding replication, a tool holder was additionally made to be attached to the robotic arm, as well as implementation of the pose calibration of the tool tip and the pose calibration of the mould.

Keywords: Robotization, Robotic Sanding, Robot Manipulation, Pose Calibration, Demonstration-Supported Robot Learning, Movement Replication, DMP, Dynamic Movement Primitives