

РОБОТИКА И ФЛЕКСИБИЛНА АУТОМАТИЗАЦИЈА
/
ROBOTICS AND FLEXIBLE AUTOMATION
(PO/ROI)

Elektrotaktilni *feedback* za prepoznavanje osobina predmeta manipuliranih mekim robotom

Gorana Marković, Jovana Malešević, Milica Isaković, Miloš Kostić, Matija Štrbac, Kosta Jovanović

Apstrakt— Glavni preduslov da se u potpunosti iskoriste prednosti telemanipulacionih sistema jeste dvosmjerna razmjena haptičkih informacija između operatora i udaljene okoline, što omogućuje operatoru da percipira kolizije, kontaktne sile, težinu, oblik, veličinu, strukturu objekta i sl. Da bismo odgovorili na neke od ovih zahtjeva, u ovom radu predlažemo upotrebu elektrotaktilne stimulacije s prostornim i frekvencijskim kodiranjem informacija. Korišćeni sistem sastoji se od robotskog aktuatora konačne mehaničke krutosti, električnog stimulatora i površinske elektrode s više polja (engl. *multi-pad*) koja se postavlja na vrh kažiprsta subjekta. Za razliku od sličnih postavki haptične povratne sprege, u našoj studiji ispitanici su se oslanjali isključivo na taktilne, bez vizuelnih ili auditivnih povratnih informacija. Eksperimentalni rezultati pokazali su da elektrotaktilna stimulacija može poslužiti za prenos informacija o mekoći (prosječna stopa prepoznavanja 3 nivoa mekoće iznosila je 90%) i veličini (prosječna stopa prepoznavanja 2 veličine iznosila je 98%) predmeta koji se hvata udaljenim aktuatorom.

Ključne reči—Telemanipulacija; Elektrotaktilni *feedback*; Meki robot

I. UVOD

Daljinski upravljani (telemanipulacioni, teleoperacioni) robotski sistemi koriste se u sredinama u kojima direktni ljudski kontakt može biti ili nebezbedan (poput nuklearnih postrojenja [1] i svemira [2]) ili nedovoljno precizan (robotski asistirana telehirurgija [3]). Kontrola ranih telemanipulacionih sistema bazirala se prvenstveno na vizuelnim povratnim informacijama (eng. *feedback*), pri čemu ljudski operater nije dobijao povratne informacije o interakciji aktuatora sa okolinom, što je obično rezultiralo velikim kognitivnim opterećenjem za njega.

Da bi se nadoknadila ograničenja vizuelnih i auditivnih povratnih informacija, počevši od sredine 1990-ih, razvoj haptičkih interfejsa započeo je džojsticima u industriji igara [4], a nastavio je brzi rast, posebno u smjeru vibrotaktilnog i *feedback*-a sile. Studije o učinku haptičkog *feedback*-a na zadacima manipulacije objektima otkrile su da haptička povratna sprega poboljšava performanse i efikasnost telemanipulacije utičući na smanjenje kontaktnih sila, potrošnje energije, vremena i broja grešaka prilikom izvršenja zadatka [3], [5]–[7]. Međutim, vizuelni senzori (najčešće

različite vrste kamera) u udaljenoj okolini i dalje ostaju nezamjenjiv dio *feedback* sistema [8].

Prevalencija vibrotaktilne stimulacije u odnosu na druge modalitete *feedback*-a postoji i dalje u većini današnjih primjena, uprkos određenim nedostacima. U posljednjih 20 godina realizovani su vibrotaktilni *feedback* sistemi namijenjeni različitim primjenama bilo samostalno [9] ili u kombinaciji s drugim *feedback* modalitetima [10]. Za razliku od vibrotaktilnih, sistemi elektrotaktilnog *feedback*-a su efikasni u pogledu potrošnje energije, jednostavni su za proizvodnju i sposobni su proizvesti senzaciju čiji se parametri mogu pouzdano kontrolisati [11]. Ideja da se povratna informacija obezbjedi pomoću sistema koji putem površinskih elektroda stimuliše kožu strujama malog intenziteta, pojavila se prije više od 40 godina [11].

Takva ideja korištena je u *feedback* sistemima za proteže gornjih udova [12]–[15], za pojačavanje interakcije za korisnike virtuelne realnosti [16], te pri upravljanju robotskom šakom tokom zadataka poput izbjegavanja prepreka i klin-urupu (engl. *peg-in-a-hole*) [17]. Pamungkas i Ward su u svoja tri istraživanja kombinovali elektrotaktilnu stimulaciju sa sistemom za stereo viziju [18], [19] i infra crvenim (engl. *infra-red, IR*) sensorima daljine i sensorima sile postavljenim na hvataljku robota [17]. Identifikaciju predmeta pomoću elektrotaktilnog *feedback*-a ispitalo je nekoliko istraživačkih grupa. Li i saradnici su testirali stopu tačnosti prepoznavanja tri objekta različite težine: lakog, srednje teškog i teškog, koristeći različite šeme kodiranja, uključujući amplitudsku, frekvencijsku i modulaciju širine impulsa [20]. Arakeri je sa svojom grupom pokazao da su ispitanici s povezom preko očiju uspješno naučili da primjene elektrotaktilne senzacije za razlikovanje 27 predmeta različite težine, širine i popustljivosti [21]. Chai i njegov tim su pokazali da ispitanici mogu prepoznati predmete u tri veličine, tri mekoće kao i četiri nivoa sile hvatanja tih predmeta sa relativno visokom stopom tačnosti koristeći prostorno, amplitudsko i mješovito kodiranje [22].

U ovom radu, pokazali smo da elektrotaktilna stimulacija može na intuitivan način pružiti informacije ljudskom operatoru o mekoći i veličini predmeta kog hvata udaljeni aktuator. Ispitanici su se oslanjali isključivo na taktilne senzacije, dok je dotok vizuelnih i auditivnih informacija bio onemogućen. Hvatanje je izvedeno pomoću aktuatora promjenive krutost (engl. *Variable Stiffness Actuator*), QBmove Maker Pro [23], a podaci o položaju i momentu sile hvataljke prikupljenih sa njenih senzora su kodirani lokacijom i učestalošću elektrotaktilne stimulacije na vrhu kažiprsta.

U prikazanoj studiji testirali smo sljedeće hipoteze:

1. Elektrotaktilna stimulacija može proizvesti dovoljan broj različitih taktilnih senzacija tako da pouzdano prenose

Gorana Marković, Kosta Jovanović – Elektrotehnički fakultet, Univerzitet u Beogradu, Bulevar Kralja Aleksandra 73, 11020 Beograd, Srbija (e-mail: kostaj@etf.bg.ac.rs)

Gorana Marković, Jovana Malešević, Milica Isaković, Miloš Kostić, Matija Štrbac – TecNALIA Serbia Ltd, Deligradska 9/39, 11000 Beograd, Srbija (e-mails: {gorana.markovic} {jovana.malesevic} {milica.isakovic} {milos.kostic} {matija.strbac}@tecnalia.com).

informacije o dvije veličine i tri mekoće predmeta;

2. Prostorna i frekvencijska modulacija su nezavisne, a mješovito prostorno-frekvencijsko kodiranje je intuitivno i može se razlikovati bez međusobnog maskiranja;

3. Visok nivo tačnosti prepoznavanja (iznad 90%) može se postići bez dodatnih vizuelnih ili auditivnih povratnih informacija.

II. METOD

A. Postavka sistema

Postavku sistema čine:

1. QBmove maker pro aktuator i hvataljka;
2. Tactility stimulator;
3. Posebno dizajnirana Tactility *multi-pad* elektroda;
4. Personalni laptop računar (Intel Core i7-10510U CPU @ 1.8 GHz, 16 GB RAM) i monitor (dijagonale jednake 60.45 cm).

1) Qbmove maker pro aktuator

QBmove maker pro je bidirekcion, antagonistički aktuator promjenjive krutosti, čija ekvilibrijumska pozicija izlazne osovine zavisi od pozicija dva motora q_1 i q_2 (1). Položaj izlazne osovine (x) direktno se mjeri enkoderom, dok se moment sile (τ) procjenjuje iz (2).

$$x_{eq} = \frac{q_1 + q_2}{2}. \quad (1)$$

$$\tau = k_1 * \sinh(a_1 * (x - q_1)) + k_2 * \sinh(a_2 * (x - q_2)). \quad (2)$$

Parametri $k_1 = 0,0227$ Nm, $k_2 = 0,0216$, $a_1 = 6,7328$ 1/rad i $a_2 = 6,9602$ 1/rad, baš kao i jednačine, dobijeni su iz tehničkog lista [24].

Aktuator komunicira s računarom putem USB komunikacije. Softver s jednostavnim grafičkim korisničkim interfejsom (engl. *Graphical User Interface, GUI*) za skeniranje portova, povezivanje, aktiviranje i podešavanje aktuatora preuzet je s veb stranice QB robotics [23]. Hvataljka korištena u ovoj studiji je štampana 3D štampačem i sastoji se iz dva dijela: nepokretnog i pokretnog. Pokretni dio je povezan direktno na osovinu, tako da je apertura hvataljke direktno proporcionalna položaju izlazne osovine.

2) Sistem za elektrotaktilnu stimulaciju

Elektrotaktilni sistem se sastojao od stimulatora i posebno dizajnirane *multi-pad* elektrode za površinsku stimulaciju. Razvojni Tactility stimulator (Tecnalia R&I, San Sebastian, Španija) je strujno kontrolisan stimulator koji generiše simetrične, bifazne impulse čija je impulsna širina u opsegu od 30 μ s do 500 μ s (sa korakom od 10 μ s), frekvencija u opsegu između 1 Hz i 200 Hz (sa korakom od 1 Hz) i amplituda u rasponu od 0,1 mA - 9 mA (sa korakom od 0,1 mA). Elektroda je proizvedena procesom sitoštampe komercijalnim mastilima za biomedicinsku upotrebu na komercijalnoj PET (polietilen tereftalat) podlozi. Provodni sloj napravljen je od srebra (srebro-srebro-hlorid, Ag/AgCl,

proizvođača Henkel Electrodag 6037E SS), sa izolacionim premazom (Henkel Electrodag PF-455B) koji pokriva provodna polja.

Čitav sistem elektrotaktilne stimulacije, zajedno s QBmove aktuatorom i predmetima koji se koriste za hvatanje, prikazan je na Sl. 1. Elektroda i način njenog pozicioniranja na kažiprstu prikazani su na Sl. 2.

B. Kodiranje feedback-a

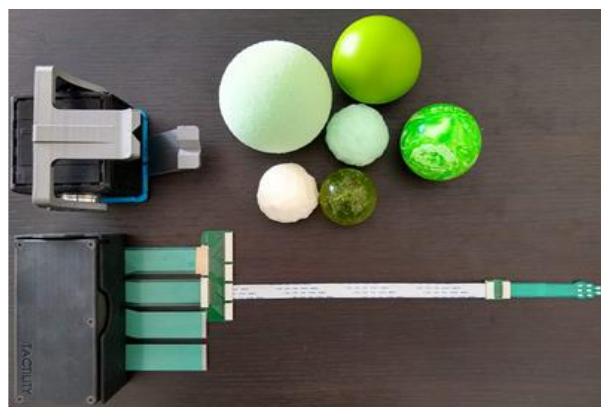
Informacija o mekoći predmeta je prenijeta putem frekvencijske modulacije stimulacije, dok su veličine kodirane prostorno. Prostorno kodiranje odnosi se na upotrebu različitih polja elektrode za izazivanje senzacija na različitim lokacijama, u ovom slučaju na vrha kažiprsta. Za ovaj eksperiment korištena su dva od osam polja elektrode: br. 2 i br. 7 (Sl. 2, lijevi panel). Pored toga, tri nivoa frekvencije su empirijski odabrana: niska = 5 Hz, srednja = 20 Hz i visoka = 50 Hz.

Prilikom prepoznavanja mekoće, maksimalni moment sile postignut tokom hvatanja procjenjen je pomoću (2) i zatim mapiran u jedan od tri nivoa frekvencije tako da najmanja mekoća odgovara najmanjoj frekvenciji. S druge strane, u zavisnosti od izmjerenog položaja izlazne osovine pokretnog dijela hvataljke u trenutku kontakta s objektom pali se određeno polje: br. 2 ako je predmet mali, br. 7 ako je predmet veliki. Granice nivoa momenta sile i pozicije određene su na osnovu pilot mjerenja prilikom hvatanja šest različitih lopti za unaprijed definisanu vrijednost krutosti aktuatora.

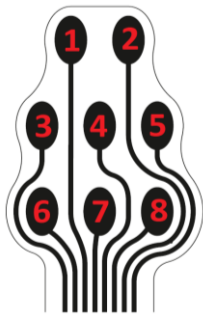
Tabela I rezimira prostornu i frekvencijsku mapu kodiranja.

C. Protokol

Deset zdravih dobrovoljaca ($26,1 \pm 3,7$ godina, pol: 6 žena / 4 muškaraca, dominantna ruka: desna 9/lijeva 1) učestvovalo je u tri psihometrijska testa. Ispitanicima je objašnjen protokol i potencijalni rizici, nakon čega su potpisali pristanak za učešće u studiji. Ispitanici su bili smješteni u sjedeći položaj, s podlakticom oslonjenom na ravnu površinu i leđima okrenutim prema osobi koja vrši eksperiment, a samim tim i postavci sistema. Eksperiment je izveden u tihom ambijentu, a ispitanici su nosili izolacione slušalice u slučaju jače buke.



Sl. 1. Stimulator, *multi-pad* elektroda, Qbmove aktuator i objekti korišćeni u eksperimentu. Lopte su prečnika 5,4 cm (velika) i 4 cm (mala).



Sl. 2. Tehnički crtež elektrode (lijevo) i njeno pozicioniranje na vrh kažiprsta (desno).







Osoba koja vrši eksperiment je postavila *multi-pad* elektrodu na kažiprst nedominantne ruke ispitanika i navukla silikonski naprstak preko elektrode (Sl. 2, desno). Silikonski naprstak je korišćen za obezbjeđivanje dobrog kontakta elektrode sa kožom i sprječavanje neželjenog pomjeranja elektrode. Prije tri glavna testa, protokol je uključivao kalibraciju i trening.

Kalibracija je podrazumjevala postupak ručnog podešavanja amplitude stimulusa kako bi se ispod oba polja izazvale jasne, ugodne i lokalizovane senzacije. Podešavanjem jednakih intenziteta senzacije ispod oba korišćena polja izbjegnuto je maskiranje slabijih stimulusa onim jačim. Preostali parametri stimulacije, frekvencija i širina impulsa, bili su konstantni tokom kalibracije intenziteta i postavljeni na 20 Hz, odnosno 400 μ s, redom.

Trening se sastojao od familijarizacije i učenja sa podsticanjem. Tokom familijarizacije, u posebno pripremljenoj aplikaciji, ispitanik samostalno bira različite kombinacije parametara stimulacije (Tabela I), na taj način se upoznaje sa različitim osjećajima i vježba lokalizaciju osjećaja izazvanih aktiviranjem dva polja i diskriminaciju između tri nivoa frekvencije. Na osnovu samoprocjene ispitanika da je spreman, prelazi se na učenje sa podsticanjem. Učenje sa podsticanjem uključivalo je provjeru identifikacije šest poruka (2 lokacije i 3 frekvencije), dok je ispitanik primao usmene povratne informacije o tačnom odgovoru. Nakon deset uzastopno tačno identifikovanih poruka, ispitanik je smatran spremnim za testove. Trajanje treninga mjereno je za sve ispitanike.

Protokol testova obuhvatao je nekoliko koraka: osoba koja vrši eksperiment prvo stavlja 1 od 6 lopti (Sl. 1) između prstiju hvataljke, a potom pokreće robota putem specijalno dizajniranog LabVIEW (National Instruments, Austin, TX) grafičkog korisničkog interfejsa. Hvataljka se konstantnom brzinom zatvara dok ne dostigne unaprijed definisani položaj i uhvati predmet, a ispitanik prima taktilni *feedback* o veličini i/ili mekoći lopte i daje verbalni odgovor o svojoj interpretaciji primljenih povratnih informacija. Prvi i drugi test obuhvatali su po 10 hvatanja pseudo-nasumično odabranih loptica, dok je zadatak ispitanika bio da da usmeni odgovor o mekoći odnosno veličini lopte koju je robot

TABELA I
MAPA KODIRANJA SENZORSKIH PODATAKA U PARAMETRE
ELEKTROTAKTILNE STIMULACIJE

Lopta	Frekvencija [Hz]	Moment sile (Nm)	Aktivno polje	Pozicija izlazne osovine (°)
<i>Mala meka</i>	5	$\tau < 1.6$		$x > 8$
<i>Mala srednje meka</i>	20	$1.6 < \tau < 2.2$		$x > 8$
<i>Mala tvrda</i>	50	$\tau > 2.2$		$x > 8$
<i>Velika meka</i>	5	$\tau < 1.6$		$x < 8$
<i>Velika srednje meka</i>	20	$1.6 < \tau < 2.2$		$x < 8$
<i>Velika tvrda</i>	50	$\tau > 2.2$		$x < 8$

uhvatio. Tokom prvog testa, polje br. 7 uvijek je bilo aktivno nezavisno od veličine predmeta, dok se frekvencija stimulacije mijenjala zavisno od mekoće lopte. Frekvencija stimulacije u drugom testu bila je 20 Hz, nezavisno od mekoće predmeta, dok se aktivno polje mijenjalo zavisno od njegove veličine. Treći test bio je kombinacija prethodna dva, i cilj mu je bio da ispita sposobnost istovremenog prepoznavanja i veličine i mekoće lopte u 12 pseudo-nasumičnih pokušaja. Ostale vrijednosti parametara (amplituda i širina impulsa) ostale su nepromijenjene u sva tri testa.

Nakon završetka eksperimenta, svaki od 10 ispitanika ocijenio je sljedeće tvrdnje na skali od 1 do 10 (1 - uopšte se ne slažem, 10 - u potpunosti se slažem):

Tvrdnja 1: Vjerujem da sam bio dobar u prepoznavanju mekoće (test 1)

Tvrdnja 2: Vjerujem da sam bio dobar u prepoznavanju veličina (test 2);

Tvrdnja 3: Vjerujem da je trening značajno doprinio uspješnosti u testovima.

III. ANALIZA PODATAKA

Glavna izlazna mjera u svim testovima bila je stopa uspješnosti prepoznavanja (SUP) (%) mekoće (test 1), veličine (test 2) ili obje osobine lopte (test 3). Pored toga, u testu 3 izračunali smo marginalne SUP za dvije osobine. S obzirom na to da je Anderson-Darling test pokazao da podaci nisu normalno raspoređeni, za statističku analizu su korišćeni neparametarski testovi, a rezultati u tekstu su prikazani u formi medijan (interkvartilni opseg - IKO). Wilcoxon-ov test rangova korišćen je za upoređivanje uspješnosti ispitanika prilikom prepoznavanja informacija o osobinama predmeta koje su prenijete pojedinačno i u kombinaciji, odnosno mekoće u testu 1 u odnosu na test 3 i veličine u testu 2 u odnosu na test 3. Prag statističke značajnosti je postavljen na $p < 0.05$. Uticaj trajanja treninga na performanse testa ispitan je računanjem korelacije između ove dvije varijable.

IV. REZULTATI

Sl. 3 prikazuje matrice konfuzije koje slikovito prikazuju rezultate ispitanika prilikom prepoznavanja mekoće lopte (test

1 - gore lijevo), veličine (test 2 - gore desno) i obje osobine u kombinaciji (test 3 - dole). U testu 1, svi ispitanici su svaki put prepoznali tvrdu loptu, dok su rijetko mijenjali srednje meku i meku loptu za onu jedan nivo tvrdu. U testu 2, svi ispitanici su prepoznali malu loptu u svim pokušajima, a jedini izvor greške bilo je pogrešno tumačenje velike lopte kao male u par navrata. U testu 3, ispitanici su češće pogrešno interpretirali mekoću lopte nego njenu veličinu, dok je SUP od 100% evidentna za najrazličitije lopte, tj. veliku tvrdu i malu meku.

Tabela II prikazuje pojedinačne i grupne rezultate po ispitanicima: SUP u tri testa, marginalne SUP prilikom identifikacije mekoće i veličine u testu 3, trajanje treninga i informacije o prethodnom iskustvu ispitanika u električnoj stimulaciji. Ispitanici su mogli prepoznati mekoću u testu 1 i veličinu u testu 2 sa vrijednošću medijane (IKO) SUP jednakom 100 (20)%, odnosno 100 (0)%. Marginalne SUP ove dvije osobine bile su vrlo slične SUP u pojedinačnim testovima: mekoća 96 (25)% i veličina 100 (0)%. Statistička analiza nije pokazala značajne razlike između SUP osobina u pojedinačnim i kombinovanom testu. Prosječno trajanje treninga bilo je ispod 5 minuta i pokazuje umjerenu negativnu korelaciju ($c = -0,49$) sa SUP u testu 3.

Tabela III pokazuje rezultate upitnika. Subjektivno mišljenje o prepoznavanju veličine (Tvrđnja 2) ima veću

ocjenu od subjektivnog mišljenja o prepoznavanju mekoće (Tvrđnja 1).

V. DISKUSIJA

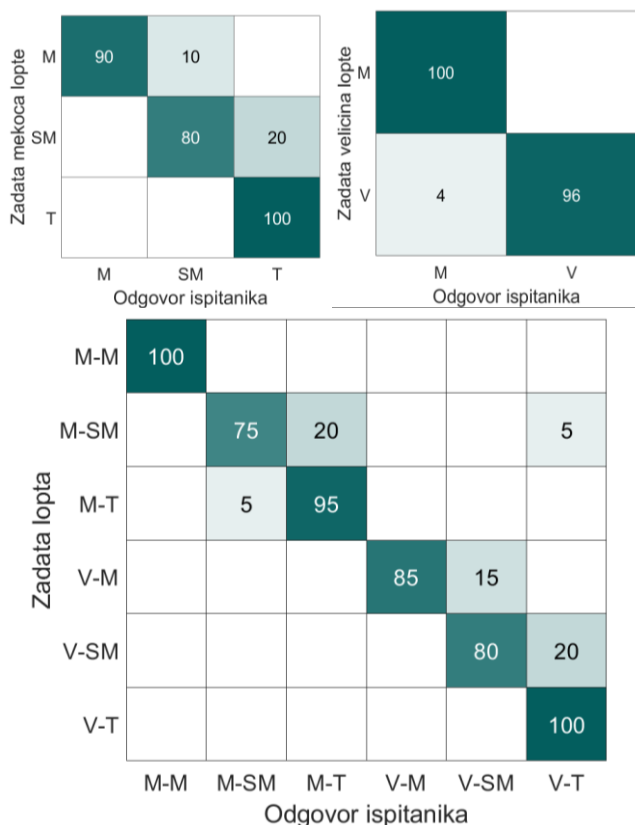
A. Rezultati testova

Tabela III sugerira da su subjekti bili uspješniji u prepoznavanju veličine, što odgovara prepoznavanju 2 lokacije stimulacije, u odnosu na prepoznavanje mekoće, što odgovara prepoznavanju 3 frekvencije stimulacije. To se može primjetiti upoređivanjem rezultata prvog i drugog testa, ali i na osnovu rezultata trećeg testa u kojem su ispitanici rijetko pogrešno protumačili informaciju o veličini objekta. Marginalna SUP mekoće i marginalna SUP veličine u kombinovanom zadatku u potpunosti su usklađene sa SUP postignutim u pojedinačnim testovima. Statistička analiza pokazala je da ne postoji značajna razlika u SUP u pojedinačnom i kombinovanom zadatku, čime je potvrđena početna hipoteza da se prostorna i frekvencijska modulacija mogu koristiti kao dva nezavisna *feedback* kanala. Prosječna SUP veličine (test 2) od 100 (0)%, bio je očekivan ishod s obzirom da je u prethodnim istraživanjima zabilježena prosječna tačnost od 97% za prostornu diskriminaciju 4 polja [12]. Međutim, budući da su u pomenutom istraživanju ova 4 polja bila locirana oko podlaktice s razmakom između polja jednakim ~ 4,8 cm, a u našoj studiji polja su smještena na vrhovima prstiju s razmakom od samo ~1,5 cm, ovi rezultati ohrabruju i sugeriraju da bi i više lokacija stimulacije moglo pouzdano da se prepozna na vrhu prsta.

U zadatku prepoznavanja mekoće, ispitanici su također postigli visoku SUP od 100 (20)%. Ovaj rezultat je u skladu s prethodnim rezultatima [12] gdje su autori pokazali da se 3 nivoa frekvencije (4 Hz, 27 Hz i 100 Hz) mogu prepoznati sa SUP jednakom 99% nakon kratkog treninga. Na gornjem, lijevom panelu na Sl. 3 može se primjetiti da je najniža SUP od 80% postignuta prilikom prepoznavanja srednje meke lopte, koja je pogrešno prepoznata kao tvrda, dok su s druge strane svi ispitanici tačno prepoznali tvrdu loptu. U kombinovanom testu, SUP je bila najniža prilikom identifikacije srednje meke lopte, i velike (80%) i male (75%). Matrica konfuzije (Sl. 3, dole) pokazuje da su ispitanici često mijenjali srednje meku loptu za tvrdu. Štaviše, velika, meka lopta je relativno često interpretirana kao velika, srednje meka lopta. Generalno se više grešaka javljalo kada su se u testu koristile velike lopte.

B. Individualni rezultati i uticaj dužine trajanja treninga

Četiri od deset ispitanika imalo je SUP jednaku 100% u sva tri testa, a prosječnu SUP jednaku ili veću od 90% postiglo je 7 ispitanika u testu 1, 8 ispitanika u testu 2 i 6 ispitanika u testu 3, koji se smatra najtežim. Primjetno je da je svaki ispitanik koji je griješio u prvom testu, gotovo sigurno postigao nižu SUP u trećem testu, što opet potvrđuje da je prepoznavanje mekoće bio izazovniji zadatak. Uzimajući u obzir subjektivni utisak, može se primjetiti da su ispitanici mogli dobro procijeniti sopstvene performanse u testovima (Tvrđnje 1 i 2, Tabela III). Na primjer, ISP8 je ocjenio



Sl. 3. Matrice konfuzije sa SUP (%) mekoće (gore, lijevo), veličine (gore, desno) i obje osobine zajedno (dole). Mekoća: 'M' = meka, 'SM' = srednje meka, 'T' = tvrda; Veličina: 'M' = mala, 'V' = velika; Kombinovano: 'M-M' = mala meka, 'M-SM' = mala srednje meka, 'M-T' = mala tvrda, 'V-M' = velika meka, 'V-SM' = velika srednje meka, 'V-T' = velika tvrda.

TABELA II

SUP (%) U TRI TESTA, MARGINALNA SUP U TESTU 3, TRAJANJE TRENINGA I INFORMACIJE O PRETHODNOM ISKUSTVU 10 ISPITANIKA. USREDNJENE VRIJEDNOSTI SU PRIKAZANE KAO MEDIJANA (INTERKVARTILNI OPSEG). PRETHODNO ISKUSTVO JE OZNAČENO KAO: 'F' = ISPITANICI SA DUGOGODIŠNJIM ISKUSTVOM U ELEKTROTAKTILNOM *FEEDBACK*-U; 'S' = ISPITANICI SA ISKUSTVOM U NEKOM VIDU ELEKTRIČNE STIMULACIJE; 'B' = ISPITANICI BEZ IKAKVOG ISKUSTVA

Isпитаници	Test 1 – mekoća [%]	Test 2 – veličina [%]	Test 3 – oboje [%]	Test 3 – marginalna mekoća [%]	Test 3 – marginalna veličina [%]	Trajanje familiarizacije [min]	Trajanje učenja sa podsticanjem [min]	Prethodno iskustvo (F/S/B)
ISP1	100	90	100	100	100	1.4	2.9	S
ISP2	100	100	83.3	83.3	91.7	2.0	1.4	S
ISP3	50	100	75	75	100	2.4	1.5	S
ISP4	100	100	100	100	100	3.4	2.3	F
ISP5	100	100	100	100	100	1.5	1.9	F
ISP6	100	90	75	75	100	1.9	5.2	S
ISP7	100	100	100	100	100	1.1	1.2	F
ISP8	60	100	66.7	66.7	100	4.9	4.6	B
ISP9	100	100	100	100	100	6.3	1.1	B
ISP10	80	100	91.7	91.7	100	2.4	2.5	S
Mediana (IKO)	100 (20)	100 (0)	96 (25)	96 (25)	100 (0)	2.2 (1.9)	2.1 (1.5)	

Tvrđnju 1 ocjenom 7 i postigao SUP mekoće jednaku 60% i 66,7% u testu 1 i testu 3, respektivno. To ukazuje na to da su ispitanici bili svjesni u kojoj mjeri su se elektrotaktilne informacije dobro razumjele, a kada su bile nejasne i nisu bili sigurni u svoj odgovor.

Nemoguće je izvući jasne zaključke o direktnom odnosu između dužine trajanja treninga i pojedinačne SUP postignute u eksperimentu. Neki ispitanici koji su duže trenirali postigli su odlične rezultate, dok su se neki izjednačili ili bili gori od subjekata koji su dosta kraće trenirali. Međutim, primjetno je da su ispitanici bez prethodnog iskustva u bilo kojoj vrsti električne stimulacije (ISP8 i ISP9) imali najduže trajanje treninga (Tabela II). Štaviše, ispitanici sa dugogodišnjim iskustvom (ISP4, ISP5 i ISP7) uglavnom su se mogli pripremiti za test nakon kratkog treninga (do 5 minuta). Rezultati upitnika pokazuju da su i iskusni i neiskusni subjekti istakli važnost treninga (Tvrđnja 3, Tabela III).

Uzimajući u obzir da su sva tri ispitanika sa dugogodišnjim iskustvom imala ukupnu SUP jednaku 100%, možemo zaključiti da rezultati u velikoj mjeri zavise od prethodnog iskustva ispitanika u elektrotaktilnom *feedback*-u, što je u skladu s našim prethodnim istraživanjem koje je pokazalo dugoročni efekat učenja [25]. Međutim, treba skrenuti pažnju na to da je ISP8 postigao identičan SUP u prepoznavanju

veličine lopte bez ikakvog prethodnog iskustva, što ukazuje na to da su poruke bile intuitivne i da ih je lako bilo naučiti nakon kratkog treninga (manje od 10 minuta). S druge strane, bilo je i ispitanika koji, uprkos protokolu koji je obuhvatao prenos samo šest različitih poruka i činjenici su imali neograničeno vrijeme za trening, nisu mogli postići visoku SUP. To se može pripisati drugim faktorima poput taktike učenja ispitanika, brzine učenja, mentalne koncentracije, stope navikavanja, fizičkog stanja (poput umora) i sl. koji mogu da igraju važnu ulogu. U literaturi [12], [25] postoje naznake da se SUP stimulusa povećava tokom treninga, ali da bi se taj i slični fenomeni ispitali, trebalo bi provesti dalje eksperimente koji bi uključili veći broj ponavljanja, duži i trening s više pokušaja.

VI. ZAKLJUČAK

Cilj ove studije bio je ispitati mogućnost pružanja povratnih informacija sa telemanipulisanog aktuatora promjenjive krutosti o veličini i mekoći uhvaćenog predmeta putem elektrotaktilnog sistema za stimulaciju na vrhu prsta. U tu svrhu izvedeni su eksperimenti na Qbmove Maker Pro aktuatoru, dok je elektrotaktilni *feedback* obezbjeđen zahvaljujući programibilnom stimulatoru i specijalno dizajniranoj, fleksibilnoj, površinskoj, *multi-pad* elektrodi.

Aktuator promjenjive krutosti omogućio je izvođenje eksperimenta s različitim osobinama predmeta. Uz odgovarajuće podešavanje krutosti, mogla bi se postići osjetljivost na meke ili tvrde materijale i u oba slučaja operatoru pružiti raspoznatljiva povratna informacija. Veličina i mekoća predmeta procjenjivani su na osnovu položaja hvataljke pri kontaktu sa objektom, i podataka o momentu sile izmjerenim senzorima unutar aktuatora, a kodirani su prostornom (tj. promjenom aktivnog polja elektrode) i frekvencijskom modulacijom.

Testovi su pokazali da se tri nivoa frekvencije stimulacije (koja predstavljaju mekoću objekta) i dvije lokacije stimulusa (koje predstavljaju veličinu objekta) mogu prepoznati sa prosječnim SUP jednakom 100 (20)% i 100 (0)%,

TABELA III

REZULTATI UPITNIKA O POSTIGNUTIM PERFORMANSAMA PRILIKOM PREPOZNAVANJA MEKOĆE I VELIČINE (TVRDNJA 1 I TVRDNJA 2) I VAŽNOSTI TRENINGA (TVRDNJA 3)

	Tvrđnja 1	Tvrđnja 2	Tvrđnja 3
ISP1	10	8	10
ISP2	8	10	10
ISP3	6	10	9
ISP4	10	9	8
ISP5	10	10	10
ISP6	8	10	9
ISP7	9	9	9
ISP8	7	10	9
ISP9	10	10	8
ISP10	7	9	7

respektivno. U kombinovanom frekvencijskom i prostornom kodiranju, prosječna SUP bila je 96 (25)%, i ne pokazuje statistički značajne razlike sa SUP osobina objekata u pojedinačnim testovima (test 1 i test 2). Preliminarni rezultati predstavljeni u ovom radu ohrabruju dalja istraživanja primjene elektrotaktilnog *feedback*-a u robotskim sistemima za telemanipulaciju i drugim scenarijima od interesa.

ZAHVALNICA

Rad u ovoj studiji izveden je u okviru projekta TACTILITY, koji je finansiran od strane Evropske Unije H2020-ICT-2018-2020/H2020-ICT-2018-3 prema sporazumu o dodjeli bespovratnih sredstava broj 856718. Ovom prilikom, autori bi se željeli zahvaliti svim partnerima u projektu na radu koji nam je omogućio predstavljeno istraživanje. Zahvaljujemo se svim zdravim volonterima koji su učestvovali u našoj studiji.

LITERATURA

- [1] J. Van Oosterhout, D. A. Abbink, J. F. Koning, H. Boessenkool, J. G. W. Wildenbeest, and C. J. M. Heemskerk, "Haptic shared control improves hot cell remote handling despite controller inaccuracies," *Fusion Eng. Des.*, vol. 88, no. 9–10, pp. 2119–2122, 2013.
- [2] T. B. Sheridan, "Space teleoperation through time delay: Review and prognosis," *IEEE Trans. Robot. Autom.*, vol. 9, no. 5, pp. 592–606, 1993.
- [3] C.-H. King *et al.*, "Tactile feedback induces reduced grasping force in robot-assisted surgery," *IEEE Trans. Haptics*, vol. 2, no. 2, pp. 103–110, 2009.
- [4] M. Ouhyoung, W.-N. Tsai, M.-C. Tsai, J.-R. Wu, C.-H. Huang, and T.-J. Yang, "A low-cost force feedback joystick and its use in PC video games," *IEEE Trans. Consum. Electron.*, vol. 41, no. 3, pp. 787–794, 1995.
- [5] Y.-Y. Liao, L.-R. Chou, T.-J. Horng, Y.-Y. Luo, K.-Y. Young, and S.-F. Su, "Force reflection and manipulation for a VR-based telerobotic system," *Proc.-Nat. Sci. Coun. Repub. China Part Phys. Sci. Eng.*, vol. 24, no. 5, pp. 382–389, 2000.
- [6] M. Zhou, S. Tse, A. Derevianko, D. B. Jones, S. D. Schweitzberg, and C. G. L. Cao, "Effect of haptic feedback in laparoscopic surgery skill acquisition," *Surg. Endosc.*, vol. 26, no. 4, pp. 1128–1134, 2012.
- [7] G. Tholey, J. P. Desai, and A. E. Castellanos, "Force feedback plays a significant role in minimally invasive surgery: results and analysis," *Ann. Surg.*, vol. 241, no. 1, p. 102, 2005.
- [8] C. R. Wagner, R. D. Howe, and N. Stylopoulos, "The role of force feedback in surgery: analysis of blunt dissection," in *Haptic Interfaces for Virtual Environment and Teleoperator Systems, International Symposium on*, 2002, pp. 73–73.
- [9] H. Kim, C. Seo, J. Lee, J. Ryu, S. Yu, and S. Lee, "Vibrotactile display for driving safety information," in *2006 IEEE Intelligent Transportation Systems Conference*, 2006, pp. 573–577.
- [10] I. Zubrycki and G. Granosik, "Novel haptic device using jamming principle for providing kinaesthetic feedback in glove-based control interface," *J. Intell. Robot. Syst.*, vol. 85, no. 3–4, pp. 413–429, 2017.
- [11] A. Y. Szeto and F. A. Saunders, "Electrocutaneous stimulation for sensory communication in rehabilitation engineering," *IEEE Trans. Biomed. Eng.*, no. 4, pp. 300–308, 1982.
- [12] M. Štrbac *et al.*, "Integrated and flexible multichannel interface for electrotactile stimulation," *J. Neural Eng.*, vol. 13, no. 4, p. 046014, 2016.
- [13] M. Isaković *et al.*, "Electrotactile feedback improves performance and facilitates learning in the routine grasping task," *Eur. J. Transl. Myol.*, vol. 26, no. 3, 2016.
- [14] S. Dosen *et al.*, "Multichannel electrotactile feedback with spatial and mixed coding for closed-loop control of grasping force in hand prostheses," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 3, pp. 183–195, 2016.
- [15] G. K. Patel, S. Dosen, C. Castellini, and D. Farina, "Multichannel electrotactile feedback for simultaneous and proportional myoelectric control," *J. Neural Eng.*, vol. 13, no. 5, p. 056015, Oct. 2016, doi: 10.1088/1741-2560/13/5/056015.
- [16] D. S. Pamungkas and K. Ward, "Electro-Tactile Feedback System to Enhance Virtual Reality Experience," *Int. J. Comput. Theory Eng.*, vol. 8, no. 6, pp. 465–470, Dec. 2016, doi: 10.7763/IJCTE.2016.V8.1090.
- [17] D. S. Pamungkas and K. Ward, "Tele-operation of a robot arm with electro tactile feedback," in *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Wollongong, NSW, Jul. 2013, pp. 704–709. doi: 10.1109/AIM.2013.6584175.
- [18] D. S. Pamungkas and K. Ward, "Electro-tactile feedback system for achieving embodiment in a tele-operated robot," in *2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*, Singapore, Dec. 2014, pp. 1706–1711. doi: 10.1109/ICARCV.2014.7064573.
- [19] D. Pamungkas and K. Ward, "Immersive teleoperation of a robot arm using electro-tactile feedback," in *2015 6th International Conference on Automation, Robotics and Applications (ICARA)*, Queenstown, New Zealand, Feb. 2015, pp. 300–305. doi: 10.1109/ICARA.2015.7081164.
- [20] K. Li, P. Boyd, Y. Zhou, Z. Ju, and H. Liu, "Electrotactile Feedback in a Virtual Hand Rehabilitation Platform: Evaluation and Implementation," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 4, pp. 1556–1565, Oct. 2019, doi: 10.1109/TASE.2018.2882465.
- [21] T. J. Arakeri, B. A. Hasse, and A. J. Fuglevand, "Object discrimination using electrotactile feedback," *J. Neural Eng.*, vol. 15, no. 4, p. 046007, Aug. 2018, doi: 10.1088/1741-2552/aabc9a.
- [22] G. Chai, J. Briand, S. Su, X. Sheng, and X. Zhu, "Electrotactile Feedback with Spatial and Mixed Coding for Object Identification and Closed-loop Control of Grasping Force in Myoelectric Prostheses," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Berlin, Germany, Jul. 2019, pp. 1805–1808. doi: 10.1109/EMBC.2019.8856508.
- [23] "qrobotics: robotics and automation." <https://qrobotics.com/> (accessed Feb. 02, 2021).
- [24] "45c5a1_792590e00b134129b2b6363a1ea7de45.pdf." Accessed: Feb. 02, 2021. [Online]. Available: https://www.qrobotics.com/wp-content/uploads/2016/03/45c5a1_792590e00b134129b2b6363a1ea7de45.pdf
- [25] M. Štrbac *et al.*, "Short- and Long-Term Learning of Feedforward Control of a Myoelectric Prosthesis with Sensory Feedback by Amputees," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 11, pp. 2133–2145, Nov. 2017, doi: 10.1109/TNSRE.2017.2712287.

ABSTRACT

The major precondition to take full advantage of teleoperation systems is a bidirectional exchange of haptic information between an operator and the remote environment, enabling the operator to perceive collisions, contact forces, weight, object shapes, surface textures, etc. To address some of these requirements, we propose the use of electrotactile stimulation with spatial and frequency coding of information. The envisioned system is comprised of a remote compliant robot actuator, stimulator unit and a surface multi-pad electrode that is placed on the subject's index fingertip. Unlike similar haptic feedback setups, in our study the subjects relied exclusively on haptic feedback, without visual or auditory cues. Experimental results showed that tactile stimulation can serve as feedback regarding the softness (90% average recognition rate of 3 softness levels) and the size (98% average recognition rate of 2 sizes) of the object grasped by a remote actuator.

Electrotactile feedback on object properties manipulated by a compliant robotic actuator

Gorana Marković, Jovana Malešević, Milica Isaković, Miloš Kostić, Matija Štrbac, Kosta Jovanović

Pronalazak Optimizacione Funkcije Kretanja iz Simulirane Demonstracije Pokreta Čučnja

Filip Bečanović, Član, IEEE, Vincent Bonnet, Član, IEEE,
Samer Mohammed, Član, IEEE, Kosta Jovanović, Član, IEEE

Sažetak—U ovom radu, posebno je uspostavljen pojednostavljeni ravanski biomehanički model čoveka za zadatak čučnja. Koristeći taj model, generišu se višestruki optimalni pokreti u odnosu na različite biomehaničke optimizacione funkcije i njihove linearne kombinacije. Iz optimalnih pokreta, optimizacione funkcije kojim su generisani pronalaze se pomoću inverzne optimizacije. Optimizaciona funkcija koja daje kvalitativno najbližnje kretanje ljudskom je kombinacija minimalnog zglobnog momenta, ubrzanja i snage. Međutim, čini se da stvarno ljudsko kretanje minimizira i neke optimizacione funkcije koje nisu razmatrane u ovom radu.

I. Uvod

Ideja generisanja ljudskih pokreta optimizacijom je zastupljena u literaturi. Pregled takvih pristupa za pokret ustajanja sa stolice i dizanja kutije dat je u [1]. Unutar [2], autor koristi planarni model sa 5 stepeni slobode kako bi generisao zglobne trajektorije za dizanje kutije sa minimalnim zglobnim momentom u članku, i pokazuje da to dovodi do stabilnosti modela po principu Tačke Nultog Momenta. U [3], autori koriste trodimenzionalni model sa 55 stepeni slobode da bi generisali optimalne trajektorije za pokret dizanja kutije, koristeći dve optimizacione funkcije.

Ideja modelovanja ljudskog kretanja kao optimizacije, zatim pronalaska optimizacione funkcije isto je obradjena u literaturi. Unutar [4], autori rada modeluju ljudsko kretanje unutar prostorijske kao kretanje materijalne tačke u dvodimenzionalnom prostoru (iz ptičije perspektive), a izbor trajektorije modeluju kao optimizacioni proces. Uz pomoć takozvane "bi-level" metode za inverznu optimizaciju, oni uspevaju da pronadju optimizacionu funkciju koja generiše trajektorije koje su najpodudarnije sa snimljenim podacima pravih ljudskih trajektorija. U [5] koristeći se sličnom metodom, autori upoređuju rezultate simulacije

Filip Bečanović is a joint Ph.D. student with the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia, and the University of Paris-Est Créteil, Laboratory of Images, Signals and Intelligent Systems, 120 rue Paul Armangot, 94400 Vitry sur Seine, (email: becaphilipe@gmail.com).

Vincent Bonnet is with the Laboratory for Analysis and Architecture of Systems, 7 avenue du Colonel Roche, 31031 Toulouse, France, (email: bonnet.vincent@gmail.com).

Samer Mohammed is with the University of Paris-Est Créteil, Laboratory of Images, Signals and Intelligent Systems, 120 rue Paul Armangot, 94400 Vitry sur Seine, (email: samer.mohammed@upec.fr).

Kosta Jovanović is with the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia, (email: kostaj@etf.rs).

sa ljudskim podacima za pokret hvata rukom u transversalnoj ravni. Dok u [6], izučavan pokret je hvat rukom u sagitalnoj ravni.

U [7], predstavljena je nova metoda za inverznu optimizaciju zasnovana na zadovoljavanju Karuš-Kun-Takerovih (KKT) uslova. Ova metoda iskorišćena je u [8] za izučavanje pokreta otvaranja vrata, ili izvlačenja i zatvaranja fioke.

U ovom radu izučavaćemo pokret čučnja, kao jednostavan simetričan pokret koji se može posmatrati samo u sagitalnoj ravni ljudskog tela, i koji je određen većinski zglobovima donjeg dela tela. Upotrebićemo splajn reprezentaciju za trajektorije kao što se to radi u robotici [9], upotrebićemo standardnu formulaciju optimizacionog problema za generisanje trajektorija, a za inverznu optimizaciju upotrebićemo KKT metodu.

II. Biomehanički model

Ova deonica predstaviće modelovanje ljudskog tela za pokret dizanja kutije.

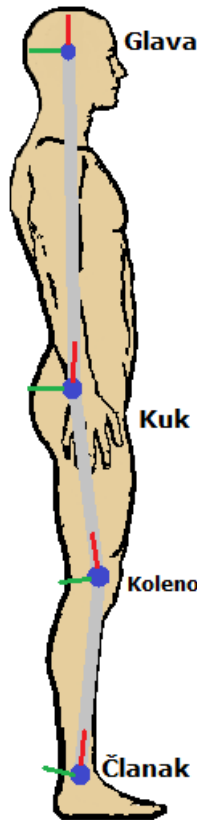
Nalik robotici, ljudsko telo je modelovano kao uzastopna veza krutih tela spojenih zglobovima. Da bi se uprostio model, iskrorišćena je simetrija zadatka kako bi se apstrahovala kretnja i svela na ravansku kretnju u sagitalnoj ravni ljudskog tela. Tim postupkom dobijamo model nalik onome na Slici 1. Osnova ljudskog modela je stopalo, dok segmente čine potkolenica, butina, i skup svih delova tela iznad trupa. Istim redosledom, povezuju ih članak, koleno, i kuk. Zajedno, ovi zglobovi daruju modelu tri ravanska stepena slobode. Treba primetiti da zbog ravanske abstrakcije, simetrični delovi tela bivaju duplirani, što znači da se noge i njihovi zglobovi svode na jednu nogu sa istim zglobovima.

Visina i masa modela fiksirane su, i jednake 1.83m i 81kg respektivno. Kinematski i inercijalni parametri delova tela određeni su korišćenjem antropometričkih tabela iz [10], iz kojih se mogu odrediti dužine segmenata, masa segmenata, pozicije centara mase segmenata, kao i matrice inercije segmenata. Duplirani segmenti modela imaju duplirane mase kao i duplirane inercije. Istovremeno, sve iznad trupa je spojeno u jedan segment, a samim tim spojene su njihove mase i inercije.

S obzirom na to da ćemo izučavati kinematiku zglobova u zglobnom prostoru, kao i kinematiku završnog uređaja (ruke) u Kartezijanskom prostoru, ali i dinamiku u zglobnom prostoru (momenti sile u zglobovima), kao

i eksterne sile i momente u Kartezijanskom prostoru, potrebno je postaviti direktan kinematski, kao i inverzni dinamički model. Direktan kinematski model, koji mapira ugaone položaje zglobova u Kartezijanski položaj završnog uređaja, može se lako proračunati korišćenjem takozvanih proksimalnih Denavit-Hartenberg parametara [11]. Za inverznu dinamiku koja mapira zglobne ugaone pozicije, brzine, ubrzanja, zajedno sa eksternim silama i momentima, u zglobne momente, korišćen je softverski paket SYMORO+ [12] za automatsko generisanje koda.

Ograničavajući zglobni uglovi kao i maksimalni zglobni momenti koji su neophodni za kasnije proračune, određeni su iz [13].



Slika 1. Ravanski biomehanički model čoveka sa tri stepena slobode

III. Kvintni Splajn

Ova deonica diskutovane splajn predstavu [14] zglobnih trajektorija.

Pokret čučnja može se zamisliti kao da model počinje iz stajaćeg položaja, zatim spušta kuk ispod određene visine, i vraća se u stajaći položaj. Naša namera je da napravimo algoritam za planiranje pokreta od početnog do krajnjeg položaja, tako da poštuje određena ograničenja i ispunjava određene zahteve, istovremeno minimizirajući neku složenu optimizacionu funkciju.

Kako bismo to uspeali, neophodna nam je matematička predstava za zglobne putanje svih zglobova. Ona mora

biti istovremeno dovoljno ekspresivna kako bi mogla predstaviti sve detalje jedne složene kretanje, ali ne isuviše kompleksna kako bi vreme proračunavanja bili relativno kratko. U ovom radu, opredelili smo se za kvintnu splajn predstavu.

Pretpostavljajući da je q funkcija vremena koja predstavlja trajektoriju jednog zgloba,

$$q \in \{q_{Članak}, q_{Koleno}, q_{Kuk}\}$$

može se predstaviti preko N parova vreme-zglobni ugao, koje nazivamo putne tačke:

$$(t_0, q_0^*), (t_1, q_1^*), \dots, (t_N, q_N^*). \quad (1)$$

Ukoliko je u početnom i krajnjem trenutku čovek u stanju mirovanja¹ postoji jedinstvena deo-po-deo polinomska trajektorija petog stepena (kvintna), koja prolazi kroz sve putne tačke i zadovoljava određene uslove glatkosti. Tajektorija se modeluje, između vremena t_i i t_{i+1} , kao polinom 5-og stepena izražen na sledeći način

$$q_i(t) = a_i + b_i t + c_i t^2 + d_i t^3 + e_i t^4 + f_i t^5. \quad (2)$$

Celokupna trajektorija je deo-po-deo kvintna funkcija koja je sastavljena od sleda pojedinačnih kvintnih polinomskih trajektorija.

$$q(t) = \begin{cases} q_0(t), & \text{if } t_0 \leq t < t_1 \\ \vdots \\ q_{N-1}(t), & \text{if } t_{N-1} \leq t < t_N \end{cases} \quad (3)$$

Ukoliko je dat skup putnih tačaka, postoji jedinstveni skup koeficijenata $\{a_i, b_i, c_i, d_i, e_i, f_i\}_{i=0, \dots, N-1}$ takvih da konačna trajektorija prolazi kroz sve putne tačke i jeste neprekidna, zajedno sa svojim prvih četiri izvoda, u intervalu vremena (t_0, t_N) .

Da bi se jedinstveno izračunali ovih $6N$ koeficijenata, možemo postaviti isti broj linearnih jednačina koje ih sadrže. Da bismo osigurali da svi kvintni polinomi prolaze kroz putne tačke na njihovim krajevima, izjednačavamo ih sa vrednošću funkcije u tim putnim tačkama.

$$q_i(t_i) = q_i^* \quad (4)$$

$$q_i(t_{i+1}) = q_{i+1}^* \quad (5)$$

Svaki pojedinačan kvintni polinom mora proći kroz putne tačke na sopstvenim krajevima, što znači da jednačine (4) i (5) važe za $i = 0, \dots, N-1$, i samim tim su odgovorne za $2N$ jednačina.

Neprekidnost izvoda trajektorije u unutrašnjim putnim tačkama može se osigurati izjednačavanjem izvoda susednih kvintnih polinoma u njihovoj zajedničkoj putnoj tački.

$$\dot{q}_i(t_{i+1}) = \dot{q}_{i+1}(t_{i+1}) \quad (6)$$

$$\ddot{q}_i(t_{i+1}) = \ddot{q}_{i+1}(t_{i+1}) \quad (7)$$

$$\dddot{q}_i(t_{i+1}) = \dddot{q}_{i+1}(t_{i+1}) \quad (8)$$

$$q_i^{(4)}(t_{i+1}) = q_{i+1}^{(4)}(t_{i+1}) \quad (9)$$

¹Početna i krajnja brzina, kao i ubrzanje, su jednaki nuli.

S obzirom da (6), (7), (8), i (9) važe za $i = 1, \dots, N - 1$, i zbog toga su odgovorne za $4(N - 1)$ jednačina.

Ovom sistemu linearnih jednačina nedostaje samo 4 jednačine da bi bio kvadratan. Preostale jednačine će podstaći to da model čoveka bude u stanju mirovanja na početku i kraju pokreta.

$$\dot{q}_0(t_0) = 0 \quad (10)$$

$$\ddot{q}_0(t_0) = 0 \quad (11)$$

$$\dot{q}_{N-1}(t_N) = 0 \quad (12)$$

$$\ddot{q}_{N-1}(t_N) = 0 \quad (13)$$

Rešavanjem ovog sistema od $6N$ jednačina sa $6N$ nepoznatih dobija se jedinstven skup koeficijenata, dakle i jedinstvena deo-po-deo kvintna trajektorija. Štaviše, ovi koeficijenti će linearno zavisiti od vrednosti funkcije u putnim tačkama.

Takodje je vredno pomenuti da izračunavanjem funkcije ili neke od njenih izvoda u nekom vremenu t' takvom da je $t_i < t' < t_{i+1}$, dobija se vrednost koja je linearno zavisna od koeficijenata i -tog polinoma, a samim tim i od vrednosti funkcije u putnim tačkama.

$$q(t') = a_i + b_i t' + c_i t'^2 + d_i t'^3 + e_i t'^4 + f_i t'^5 \quad (14)$$

Može se zaključiti da će vrednost funkcije $q(t')$ u bilo kom trenutku u vremenu $t_0 < t' < t_N$ biti linearno zavisna od vrednosti funkcije u putnim tačkama.

IV. Optimizacija

U ovoj deonici biće opisan optimizacioni problem koji se rešava prilikom generisanja trajektorija.

Kao što je pokazano u Deonici III, trajektorija svakog zgloba se može predstaviti korišćenjem N putnih tačaka. Pretpostavljajući sada da imamo trodimenzionalnu trajektoriju

$$q = [q_{Clanak}, q_{Koleno}, q_{Kuk}]^T$$

i da su putne tačke date za takvu trajektoriju, slično tome kako su date u (1), možemo ih upakovati u optimizacioni vektor

$$x = [q_0^{*T} \quad q_1^{*T} \quad \dots \quad q_N^{*T}]^T$$

gde je svaka od njih vektor $q_i^* \in \mathbb{R}^3$, $i = 0, \dots, N$.

Problem pronalaženja najprikadnijih zglobnih trajektorija u skladu sa nekom optimizacionom funkcijom, podešavajući pozicije putnih tačaka u prostoru zglobova, može se formalno predstaviti kao

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t.} \quad & h(x) = 0 \\ & g(x) \leq 0 \end{aligned} \quad (15)$$

gde je $x \in \mathbb{R}^{3(N+1)}$ optimizaciona promenljiva koja predstavlja vektor putnih tačaka, $f(x) : \mathbb{R}^{3(N+1)} \mapsto \mathbb{R}$ je optimizaciona funkcija, $h(x) : \mathbb{R}^{3(N+1)} \mapsto \mathbb{R}^m$ predstavlja ograničenja u vidu sistema jednakosti, a $g(x) : \mathbb{R}^{3(N+1)} \mapsto \mathbb{R}^p$ predstavlja ograničenja u vidu sistema nejednakosti.

U ovom radu, optimizaciona funkcija predstavlja linearnu kombinaciju skupa optimizacionih funkcija definisane vektorskom funkcijom $\tilde{f}(x)$. Optimizaciona funkcija može se izraziti kao

$$f(x) = \sum_{i=1}^{N_{CF}} \alpha_i f_i(x) = \alpha^T \tilde{f}(x). \quad (16)$$

Gde koeficijenti moraju biti pozitivni $\alpha_i \geq 0$, $i = 1, \dots, N_{CF}$ kako bi se održala lokalna konveksnost funkcija, kao i normalizovani $\sum_{i=1}^{N_{CF}} \alpha_i = 1$ kako bi svakom rešenju odgovarala jedinstvena linearna kombinacija.

Skup optimizacionih funkcija koje se razmatraju u ovom radu date su u Tabeli I. Dok su ograničenja predstavljena u Tabeli II.

Vrednosti trajektorije, brzine i ubrzanja u pojedinim trenucima ($q(t')$, $\dot{q}(t')$, $\ddot{q}(t')$) mogu se dobiti iz koeficijenata dobijenih pomoću procedure opisane u Deonici III. Kartezijanska pozicija kuka (x_{Kuk} , y_{Kuk}) može se dobiti pomoću direktnog kinematskog modela. Zglobni momenti Γ , mogu se dobiti pomoću inverznog dinamičkog modela, kao i centar pritiska COP [15].

Tabela I
Optimizacione funkcije

Ime	Izraz
Zbir Kvadrata Zglobnih Momenata	$f_1(x) = \sum_{j=1}^6 \sum_{i=0}^{N_S} \Gamma_{ji}^2$
Zbir Kvadrata Zglobnih Ubrzanja	$f_2(x) = \sum_{j=1}^6 \sum_{i=0}^{N_S} \ddot{q}_{ji}^2$
Zbir Kvadrata Zglobnih Snaga	$f_4(x) = \sum_{j=1}^6 \sum_{i=0}^{N_S} (\Gamma_{ji} \cdot \dot{q}_{ji})^2$

Tabela II
Funkcije ograničenja

Ime	Izraz
Sistemi nejednakosti	
Uslov Stabilnosti	$x_{peta} \leq COP(t_i) \leq x_{prst}, i = 0, \dots, N_{itp}$
Ograničenja Zglobnog Momenta	$\Gamma_{min} \leq \Gamma(t_i) \leq \Gamma_{max}, i = 0, \dots, N_{itp}$
Uslov Čučnja	$y_{Kuk}(t_i) \leq y_{Prag}, i = 0, \dots, N_{itp}$
Zglobna Ograničenja	$q_{min} \leq q(t_i) \leq q_{max}, i = 0, \dots, N_{itp}$
Sistemi jednakosti	
Početni Uslovi	$q(t_0) = q_{uspravno}$
Krajnji Uslovi	$q(t_N) = q_{uspravno}$

V. Inverzna Optimizacija

U ovoj deonici biće opisano kako izvršiti inverznu optimizaciju na osnovu Karuš-Kun-Takerovih uslova optimalnosti [7].

Ako je dat rezultat x^* optimizacionog problema (15) gde je optimizaciona funkcija linearna kombinacija različitih funkcija kao u (16), cilj inverzne optimizacije jeste da

se pronadju koeficijenti α_i , $i = 1, \dots, N_{CF}$ te linearne kombinacije. Pretpostavljajući da je data snimljena ljudska trajektorija u obliku putnih tački, inverzna optimizacija bi trebalo da odredi koje optimizacione funkcije je čovek koristio prilikom generisanja takve trajektorije, i time nam omogućiti generisanje čovekolikih trajektorija u budućnosti, korišćenjem pronađene optimizacione funkcije.

Koristićemo KKT uslove da rešimo problem inverzne optimizacije. Za svaki optimizacioni problem izražen kao u (15), možemo definisati Lagranžijan uz pomoć promenljivih λ i μ koje nazivamo Lagranžovim multiplikatorima.

$$L(x, \lambda, \mu) = f(x) + \lambda^T \cdot h(x) + \mu^T \cdot g(x) \quad (17)$$

Poznato je da optimalno rešenje x^* takozvanog primalnog problema (15) mora imati komplementarno dualno rešenje (λ^*, μ^*) . Zajedno, oni moraju zadovoljavati KKT uslove koji su dati u jednačinama (18)-(22).

$$\nabla_x L(x^*, \lambda^*, \mu^*) = 0 \quad (18)$$

$$h(x^*) = 0 \quad (19)$$

$$g(x^*) \leq 0 \quad (20)$$

$$\mu_i^* g_i(x^*) = 0 \quad i = 1, \dots, p \quad (21)$$

$$\mu^* \geq 0 \quad (22)$$

S obzirom na to da je naša optimizaciona funkcija linearna kombinacija skupa funkcija, možemo Lagranžijan izraziti na sledeći način, i uočiti njegovu zavisnost od koeficijenata α koje želimo povratiti.

$$L(\alpha, x, \lambda, \mu) = \alpha^T \cdot \tilde{f}(x) + \lambda^T \cdot h(x) + \mu^T \cdot g(x) \quad (23)$$

Dakle KKT uslovi zavise od koeficijenata α ali takodje i od dualnih promenljivih (λ^*, μ^*) koje nisu date. Naš pristup povratku koeficijenata α će se zasnivati na zadovoljavanju KKT uslova, a zahtevaće i da pored koeficijenata tražimo vrednost multiplikatora (λ^*, μ^*) .

Definisaćemo rezidualne kao što je uradjeno u [16].

$$r_{stat} = \nabla_x L(\alpha, x^*, \lambda^*, \mu^*) \quad (24)$$

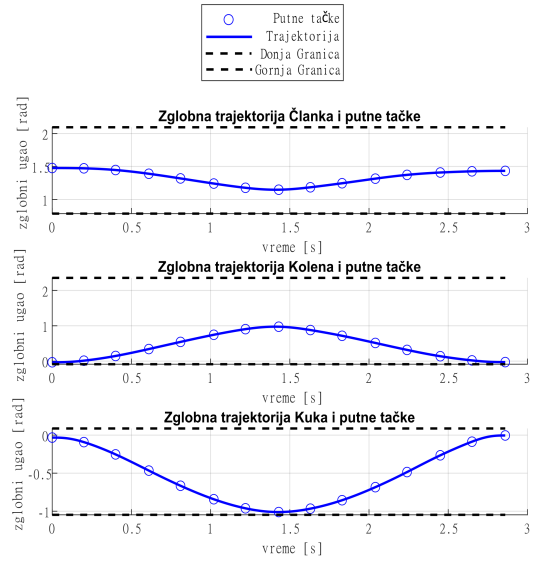
$$r_{comp} = \text{diag}(g(x^*))\mu^* \quad (25)$$

Zatim možemo formulisati želju za zadovoljavanjem KKT uslova kao optimizacioni problem pretrage za promenljivima $(\alpha, \lambda^*, \mu^*)$ takvim da su reziduali iz (24) i (25), koji odgovaraju uslovima (18) i (21), što bliži nuli a da istovremeno poštuju uslov (22) kao i uslov pozitivnosti i normalizacije koeficijenata α .

$$\begin{aligned} \min_{(\alpha, \lambda, \mu)} \quad & \|r_{stat}\|_2^2 + \|r_{comp}\|_2^2 \\ \text{s.t.} \quad & \alpha \geq 0 \\ & \mu \geq 0 \\ & \sum_{i=1}^{N_{CF}} \alpha_i = 1 \end{aligned} \quad (26)$$

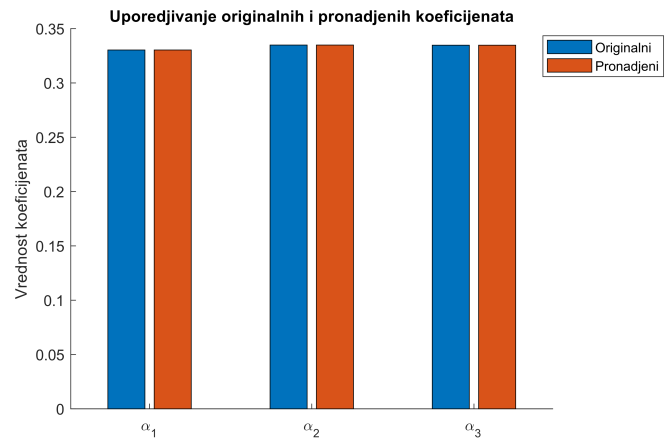
VI. Rezultati

Optimalne trajektorije generisane su po postupku opisanom u Deonici IV, sa $N = 15$ putnih tačaka, i sa koeficijentima $\alpha_1 = \frac{1}{3}$, $\alpha_2 = \frac{1}{3}$ i $\alpha_3 = \frac{1}{3}$, koji redom odgovaraju funkcijama zbira kvadrata zglobnih momenata, zbira kvadrata zglobnih ubrzanja, i zbira kvadrata globalnih snaga. Dobijene optimalne putne tačke, kao i odgovarajuće trajektorije dobijene interpolacijom opisanom u Deonici III, prikazane su na Slici 2.



Slika 2. Rezultati optimizacije

Rezultati inverzne optimizacije nad trajektorijama, odnosno putnim tačkama, generisanim sa prethodno pomenutim koeficijentima prikazani su na Slici 3. Na slici su upoređeni koeficijenti koji su iskorišćeni za proces direktne optimizacije sa pronađenim koeficijenti metodom inverzne optimizacije opisane u Deonici V.



Slika 3. Rezultati inverzne optimizacije

Ova dva prikazana grafika predstavljaju uzorak preko koga želimo da pokažemo da metoda radi u simulaciji. Metoda je isprobana za 3D rešetku vrednosti koeficijentata ($\alpha_1, \alpha_2, \alpha_3$) i pokazala je zadovoljavajuće rezultate na svim ispitanim vrednostima.

Generisanje optimalnih trajektorija u industrijskim zadacima potrebno je za procenu ergonomije tog pokreta kod ljudi, i samim tim određivanja nekih pravila izvođenja pokreta kako bi se sprečile ili umanjile mišićno-skeletalne povrede.

Takodje je od značaja mogućnost određivanja principa po kojima se kreću zdravi ljudi, i upoređivanja sa principima po kojima se kreću ljudi sa određenim patologijama, kako bi se mogle davati bolje smernice pacijentima prilikom rehabilitacije.

Budući radovi mogu primenjivati ovu metodu na pravim ljudskim podacima kako bi se pronašle odrednice kojima se ljudi vode prilikom generisanja sopstvenih pokreta. Ova metoda se takodje može primeniti i na drugim pokretima, međutim morao bi se postaviti drugačiji optimizacioni model, sa ograničenjima koja bi zavisila od samog pokreta.

Zahvalnica

Ovo istraživanje podržano je od strane Ministarstva Inostranih Poslova Republike Francuske, i Ministarstva Prosvete, Nauke i Tehnološkog Razvoja Republike Srbije, unutar programa za bilateralnu naučnu saradnju između Republike Francuske i Republike Srbije, projekat #17, HUMAN-COMAN. Ovo istraživanje takodje je podržano od strane Campus France, Francuske agencije za promociju visokog obrazovanja, i internacionalne mobilnosti.

Reference

- [1] Erik A Chumacero-Polanco and James Yang. „A review on human motion prediction in sit to stand and lifting tasks”. in: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. volume 50077. American Society of Mechanical Engineers. 2016, V01AT02A066.
- [2] Ali Leylavi Shoushtari. „What strategy central nervous system uses to perform a movement balanced? Biomechatronical simulation of human lifting”. in: Appl Bionics Biomech 10 (2013), pages 113–124.
- [3] Yujiang Xiang and others. „Human lifting simulation using a multi-objective optimization approach”. in: Multibody System Dynamics 23.4 (2010), pages 431–451.
- [4] Katja Mombaur, Anh Truong and Jean-Paul Laumond. „From human to humanoid locomotion—an inverse optimal control approach”. in: Autonomous robots 28.3 (2010), pages 369–383.
- [5] Sebastian Albrecht, Marion Leibold and Michael Ulbrich. „A bilevel optimization approach to obtain optimal cost functions for human arm movements”. in: Numerical Algebra, Control & Optimization 2.1 (2012), page 105.
- [6] Bastien Berret and others. „Evidence for composite cost functions in arm movement planning: an inverse optimal control approach”. in: PLoS Comput Biol 7.10 (2011), e1002183.
- [7] Arezou Keshavarz, Yang Wang and Stephen Boyd. „Imputing a convex objective function”. in: 2011 IEEE international symposium on intelligent control. IEEE. 2011, pages 613–619.
- [8] Peter Englert, Ngo Anh Vien and Marc Toussaint. „Inverse KKT: Learning cost functions of manipulation tasks from demonstrations”. in: The International Journal of Robotics Research 36.13-14 (2017), pages 1474–1488.
- [9] J Zico Kolter and Andrew Y Ng. „Task-space trajectories via cubic spline optimization”. in: 2009 IEEE International Conference on Robotics and Automation. IEEE. 2009, pages 1675–1682.
- [10] Raphaël Dumas and Janis Wojtusch. Estimation of the Body Segment Inertial Parameters for the Rigid Body Biomechanical Models Used in Motion Analysis. In: Müller B., Wolf S.(eds) Handbook of Human Motion. 2018.
- [11] Harvey Lipkin. „A note on Denavit-Hartenberg notation in robotics”. in: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. volume 47446. 2005, pages 921–926.
- [12] Wisama Khalil and Denis Creusot. „SYMORO+: a system for the symbolic modelling of robots”. in: Robotica 15 (1997), pages 153–161.
- [13] Thomas Robert, Julien Causse and Gérard Monnier. „Estimation of external contact loads using an inverse dynamics and optimization approach: general method and application to sit-to-stand maneuvers”. in: Journal of biomechanics 46.13 (2013), pages 2220–2227.
- [14] Sky McKinley and Megan Levine. „Cubic spline interpolation”. in: College of the Redwoods 45.1 (1998), pages 1049–1060.
- [15] Philippe Sardain and Guy Bessonnet. „Forces acting on a biped robot. Center of pressure-zero moment point”. in: IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 34.5 (2004), pages 630–637.
- [16] Adina Panchea. „Inverse optimal control for redundant systems of biological motion”. phdthesis. Université d’Orléans, 2015.

Workspace Analysis of a Collaborative Bi-manual Industrial Robotic System

Jovan Šumarac, Kosta Jovanović and Aleksandar Rodić

Abstract—Bi-manual manipulation has been a focus of extensive academic research and has found its uses in industry as well. The workspace assessment of a robot is one of the key parameters in robot consideration for commercial purposes. As well, it is essential for the research of a bi-manual robotic system that often tends to replace humans in bi-manual tasks or directly share the workspace with humans. The goal of this paper is to present a detailed workspace analysis of a dual-arm collaborative robot. The dual-arm collaborative robot has been developed at the Robotics Laboratory at "Mihajlo Pupin" Institute and it is briefly presented in the paper. The workspaces of particular robot arms on the dual-arm system, a shared workspace for bimanual operation, and a manipulability analysis are presented. The simulations have been performed in Matlab, whereas CoppeliaSim robot simulator has been used for the visualization of the results. The presented results are an essential point in consideration of optimal trajectory planning and bi-manual collaborative robot control.

Index Terms—Robot workspace; Bi-manual manipulation; Collaborative robots

I. INTRODUCTION

Bi-manual or dual-arm manipulation is increasingly applied in modern robotics, and it is gaining further momentum with the advances in collaborative robotics. Bi-manual robots' similarity to human body form together with the need to replace human auxiliary work, as well as their increased workspace and task range are some of the main reasons. Similarity to human form also means easier integration of bi-manual robots in different environments originally intended for human workers, as well as for tasks which require human robot physical interaction. The analysis of the complete expanded workspace of such a robot, both in terms of reachability and manipulability, is very important for defining and coordinating its tasks.

Since its humble beginnings in 1940s and 50s mostly for tele-operation tasks, the development and research of bi-manual robots slowed down and gave way to single arm robots. However it again gained popularity and advanced significantly since the 1990s [1]. The focus was not only on

Jovan Šumarac is with the Robotics Laboratory, "Mihajlo Pupin" Institute, University of Belgrade, Volgina 15, 11060 Belgrade, Serbia (e-mail: jovan.sumarac@pupin.rs)

Kosta Jovanović is with the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia (e-mail: kostaj@etf.bg.ac.rs)

Aleksandar Rodić is with the Robotics Laboratory, "Mihajlo Pupin" Institute, University of Belgrade, Volgina 15, 11060 Belgrade, Serbia (e-mail: aleksandar.rodic@pupin.rs)

scientific research, but also on producing commercial bi-manual robotic system.

Over the years many bi-manual robots were developed either for research or commercially. A detailed overview of the various scientific bi-manual platforms is given in [1]. Of those, the most interesting is Rollin' Justin [2]. It is a mobile robotic system and research platform that allows implementation of sophisticated control algorithms and dexterous manipulation. It is a powerful upper body humanoid robot with torso and two lightweight robot arms with four finger hands. Its workspace is quite large; its arm span is 3000 mm and the torso can move about 600 mm to the front and 300 mm to the back. However, a downside to this is that the robot is required to be mounted on a very stable base with a large footprint to prevent it from falling over.

In recent years, using collaborative bi-manual robots commercially has been a growing trend. In [3] a good overview of commercial bi-manual robots is given. The first such robot was called Baxter, presented in 2012 by Rethink Robotics. Its manipulator consists of a head, a torso and two arms with 7 degrees of freedom each. Its arm span is about 2600 mm and there is significant overlap between workspaces of both arms. Most of this common workspace is directly in front of the robot as it has a rotational limit at the shoulder [4]. ABB has also developed a commercial dual-arm collaborative robot called IRB 14000 Yumi. It can collaborate with a human and is intended for assembly of small parts. Its arms have relatively smaller ranges, about 560 mm each, resulting in a 1200 mm span and a smaller workspace[5]. Another smaller commercial bi-manual robot was also developed and presented by Epson in 2018, called the WorkSense W-01 with a similar arm range and workspace as Yumi [6].

The bi-manual robot analyzed in this paper was developed and made at the Robotics Laboratory at "Mihajlo Pupin" Institute. This paper presents the robot and an analysis of its workspace. The workspace is analyzed for each of the arms, and shared workspace is presented as well. Finally a manipulability analysis is performed. The simulation was done in Matlab and the results presented in CoppeliaSim robot simulator.

II. WORKSPACE ANALYSIS OF A BI-MANUAL ROBOT

The robot presented and analyzed in this paper was custom developed at the Robotics Laboratory [7]. The idea was to design and develop a cloud-enabled industrial human-size service robot. One of the main goals was to make it very

versatile and applicable in different tasks. The goal was also to make it the first-generation intelligent industrial service robot, meaning it possesses a significant level of artificial intelligence (such as perception, task planning, decision making, etc.).

The robot consists of several functional modules. Its base is a motorized cart that serves as a mobile platform and allows the transportation of the bi-manual robotic system. The mechanical robot torso has 4 degrees of freedom and the robot arms are mounted on it. The arms are two UR5 lightweight industrial manipulators with 6 degrees of freedom by Universal Robots.

Each robot arm has several modules at its end effector including force/torque sensors, wrist cameras and industrial grippers. Each arm is mounted on the torso at an angle of about 15 degrees which results in a larger shared workspace. Fig. 1 shows the robot in virtual reality in CoppeliaSim robot manipulator.

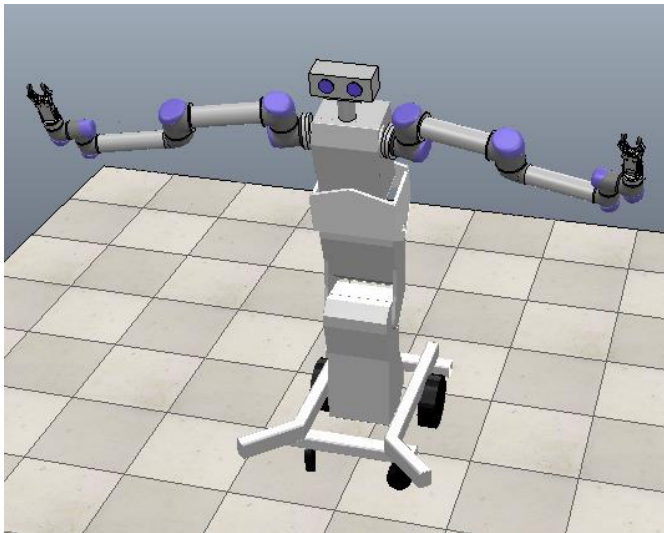


Fig. 1. The robotic system analyzed in this paper shown in a robot simulator

Fig. 1. shows the robot in a slightly bent forward torso configuration, which is common if the objects of interest are in front of the robot. It also shows the robot with a stylized head and eyes. This was done because the development of a vision and audio system is one of the goals for this robot as well.

The prototype of the mechanical torso currently developed at the Robotics Laboratory is shown in Fig. 2. The mechanical torso is of variable geometry. It can be in a standing position like in Fig. 2 which is its fully extended position. It can also be fully contracted e.g. in a bent position. In this paper the workspace analysis is performed for one, common configuration of the torso shown in Fig. 1. The results for the whole robot workspace for this torso configuration can easily be generalized for other torso positions.

A key issue in developing such a robotic system is defining and analyzing its workspace. Besides the obvious benefit of an increased workspace due to two robot arms, it is important to calculate the limits of the workspace in terms of

reachability as well as identify problem areas within the workspace in terms of manipulability. This analysis is later used for designing the robots environment as well as the kind of tasks it will perform.



Fig. 2. The prototype of the mechanical torso of the analyzed robot.

Workspace analysis starts with the analysis of the individual workspaces of the robot arms. The arm ranges are provided by the manufacturer [8]. The idea behind the algorithm is to generate a relatively large number of random configurations within joint limits for both of the arms. Forward kinematics is then calculated to get the Cartesian coordinates of the end effector, and save its position for later drawing in the robot simulator. After visualizing the workspace in form of point clouds for each arm, the arm span and the volume of the shared workspace are calculated. This procedure is repeated for different numbers of random configurations. This is important for defining the reachability of the robotic system, e.g. the points in space it can reach.

Another important aspect of workspace analysis is finding the areas of reduced manipulability. The method for calculating manipulability in this paper was first described in [9]. The so-called Yoshikawa manipulability measure [10] describes how close the robot is to a singular configuration – e.g. how close it is to losing one of its degrees of freedom. The manipulability index is calculated as:

$$m = \sqrt{\det(JJ^T)} \quad (1)$$

where J is the Jacobian matrix of the robot arm at a certain configuration. This measure is proportional to the velocity ellipsoid at a given configuration. The velocity ellipsoid indicates the ability of the robot to move in each of the 3

translational and 3 rotational directions. If m equals zero it means that one degree of freedom is lost and that the robot is at a singular configuration. The bigger the index is, the more manipulability robot has in its current position. This measure is based only on the kinematics of the mechanism and does not take into account mass and inertia. Still, it's a good method for finding problem areas within the workspace that should be avoided in actual tasks.

The manipulability index is then calculated for all of the random configurations generated earlier. A cutoff index value is defined, below which a robot configuration is considered poorly manipulable. Percentages of such configurations for different number of random positions are calculated as well.

III. SIMULATION AND RESULTS

The system was modeled and the simulation was done in Matlab. Its' Robotics Toolbox package offers a number of ready-made functions, including the computing of forward, differential and inverse kinematics, trajectory planning, robot 3D animation, etc. After calculating the point cloud of reachable points, the workspace in body planes is shown in Matlab, while the visualization of the whole workspace and robot environment is done in CoppeliaSim robot simulator.

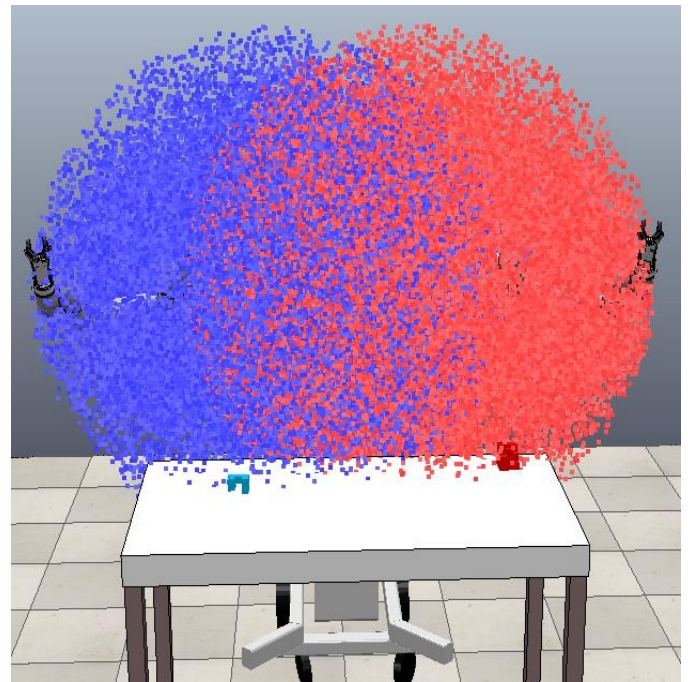


Fig. 4. Front view of the robot workspace in CoppeliaSim robot simulator.

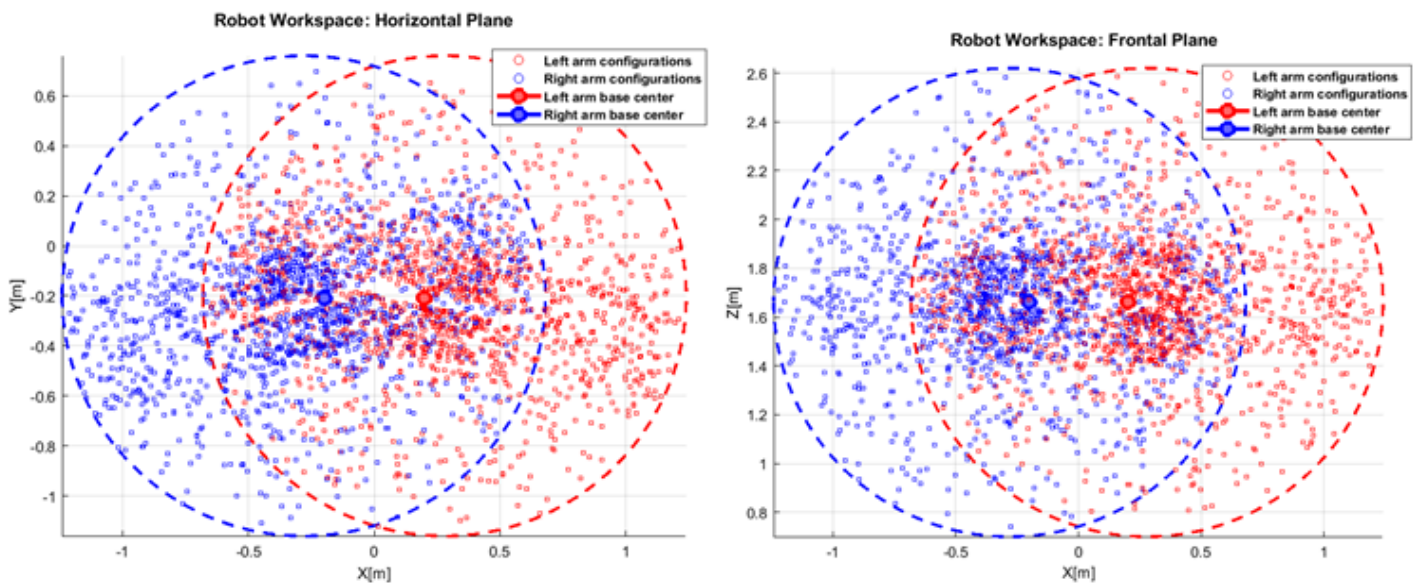


Fig. 3. Workspaces of the robot arms shown in horizontal (transverse) and frontal planes. The blue and red dashed lines mark the boundaries of the individual workspaces of the right and left robot arm workspaces, respectively.

Fig. 3 shows the robot workspace in horizontal and frontal body planes. The red points indicate the reachable points of the left arm while the blue ones are reachable for the right one. The individual arm workspaces are rounded with dashed lines so the shared workspace is easily identifiable.

Meanwhile, Fig. 4 shows the visualized workspace in robot simulator. The robot is in a slightly bent position. A table with two objects (two parts) for an assemblage task is shown in front of it as an example of a typical task and typical robot environment. As in previous figure, the reachable points for the left and right arm are shown in red and blue, respectively.

So, the visualized workspaces show the reachable area for the robot arms and for the whole robotic system as well. Since each arm has a range of 950 mm the arm span of the whole robot structure is around 2300 mm. It has to be pointed out that UR5 arms themselves have a range of 850 mm. However, adding end effector equipment such as force sensors, wrist cameras and grippers extends that range. For this simulation only the grippers were added to the end effector resulting in a 100 mm larger range. The workspace volumes of the individual arms (the red and blue cloud points in Fig. 4) are around 3.6m^3 . The volume of the shared workspace, visible in

Fig. 4 as the area where blue and red points intersect is around 1.12 m^3 . It is also clear from Fig. 4 that the objects on the table are clearly within the reachable area.

From Fig. 3 it is clear that the individual workspaces of the robot arms do not just intersect in the area in front of the robot, but also in the back, as well as above and below the top of the torso. This information can be useful for specific tasks.

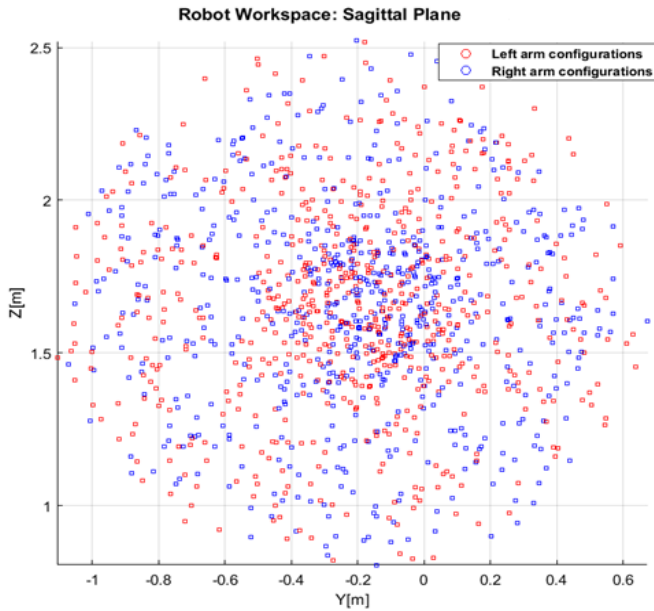


Fig. 5. Robot workspaces shown in sagittal plane.

Fig. 5 now shows the robot's sagittal plane. As its plane of symmetry, it contains similar number of configurations for both arms, equally and randomly distributed.

The robot structure itself is barely visible in Fig. 4. This is because of the relatively large number, 50000, of random configurations used for the simulation. Zooming into the point clouds in the robot simulator it can be seen that the points are equally distributed within the cloud (save for the areas in the torso which are not reachable). This is confirmed by Fig. 3 and Fig. 5 as well. These figures give the indication that all the areas within the workspace are equally reachable.

Manipulability analysis shows that that's not the case. Manipulability measure given with (1) is calculated for all the configurations used in mapping the workspaces. The key problem is identifying a key value of manipulability index, below which a configuration is considered poorly manipulable.

In [10] an index value of 10^{-5} is already considered quite poor, so a cutoff value of 10^{-4} is used to map problem areas within the workspace. Hence, the configurations with manipulability index of 10^{-4} or less are considered poorly manipulable (or "unmanipulable"). Configurations with the index between 10^{-4} and 10^{-3} are relatively manipulable and those with the index bigger than 10^{-3} are not considered problematic.

Fig. 6. shows the robot system with the poorly manipulable points drawn. The points were calculated for each of the robot

arms but were all drawn in the same color. The idea is to visualize them and identify problem areas to avoid for the whole robot in collaborative tasks.

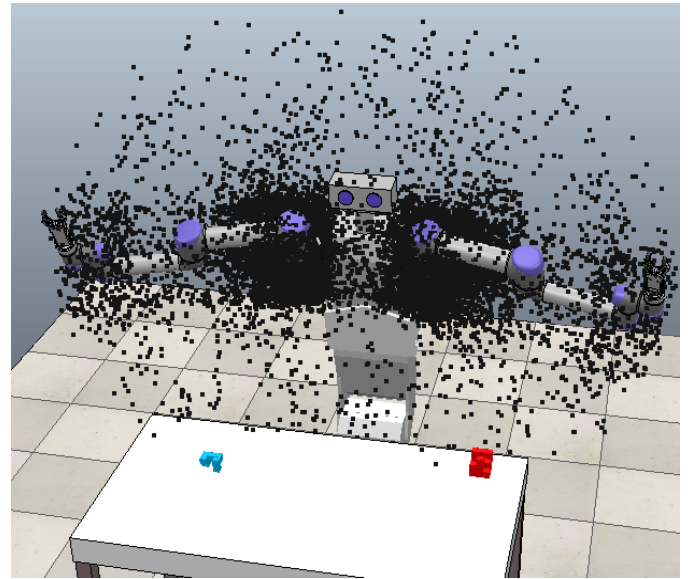


Fig. 6. Front view of the problem areas of the robot.

Clearly, there are clusters of problematic points around the bases of both arms and the first links as well as around the final links of the arms. On the other hand, there are no clusters in the area in front of the robot, closer to the table with the objects. Those are the least problematic areas which is important to know for practical tasks.

Fig. 6 shows only the points with the manipulability index equal to or less than 10^{-4} . If the relatively manipulable points were added too they would continue to cluster around the same locations. The areas which are in front of, behind, above and below the upper part of the torso, but which are a bit further from it, still remain the least affected by problematic points.

To corroborate this, and have a more detailed look into the structure of the problem areas it is good to again show the robot's body planes. This time they will be shown with unmanipulable configurations.

Fig. 7 shows the robots' horizontal and frontal planes. The problematic configurations are shown in the same color for both arms, to indicate problem areas for the whole robotic system, not just the individual arms. Again, it is evident that the least problematic areas are further away from the top of the torso in both directions.

These results again confirm the clustering of the problematic points around the top of the torso and to a lesser extent around the end effectors. The whole areas around the top part of the torso and the robot bases and first links form a clearly problematic region that should be avoided in tasks. However, this is not a practical problem for the robot. Since that region is very close to the location of the torso itself, many of those configurations are unreachable in any case, as the robot arms cannot be allowed to collide with the torso.

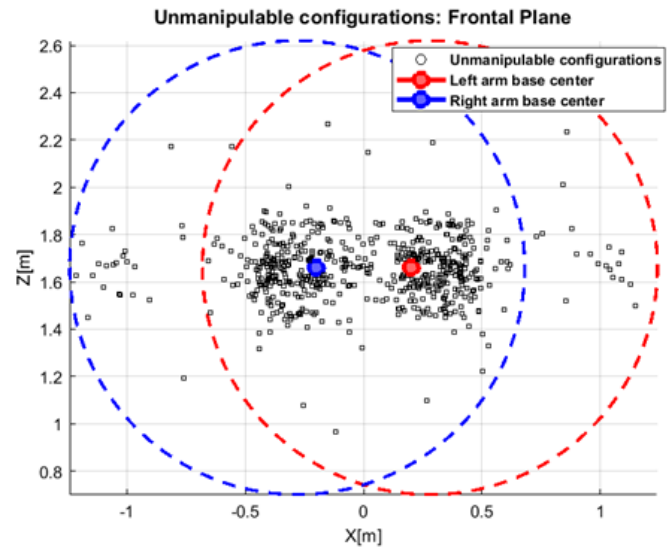
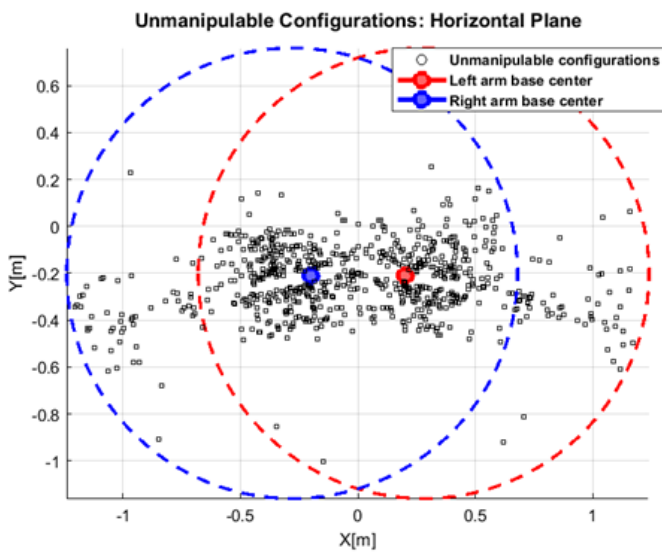


Fig. 7. Robot workspace with unmanipulable configurations shown in the horizontal and frontal plane of the robot's body.

It is interesting to analyze the number of poorly manipulable configurations, as well as the size of the manipulability index. The simulations were carried for a varying number of robot configurations, from 5000 to 50000 random positions.

The results are shown in Table I. They are shown for a single UR5 robot arm (the left one), since they are very similar to the other one.

TABLE I
MANIPULABILITY ANALYSIS

Number of configurations	Maximum manipulability index value	Minimum manipulability index value	Poorly manipulable configuration percentage
5000	0.1161	$8.472 \cdot 10^{-8}$	15.98
50000	0.1189	$1.124 \cdot 10^{-9}$	12.24
100000	0.1195	$5.710 \cdot 10^{-9}$	12.40
500000	0.1197	$1.338 \cdot 10^{-10}$	12.32

The final column shows the percentage of robot configurations with manipulability index less than 10^{-4} . It is the value below which a configuration is considered to have poor manipulability, as stated above. The percentage varies between approximately 16% for 5000 configurations to about 12.3% for a larger number of points.

The minimum value of the index is around 10^{-9} , which is pretty close to a singular configuration. The maximum value peaks at about 0.12, for the biggest number of robot configurations. While that indicates configurations with much more manipulability, the index is still not very high.

This shows that in practice, a seemingly large robot workspace can be significantly reduced by various limitations. Joint limits, self-collisions, singularities, and areas of reduced manipulability greatly impact its workspace and the way robot tasks can be defined and executed.

Another way of illustrating the manipulability in certain configurations is to show the robot with its velocity ellipsoids drawn. They are usually shown for the three translational

components of the velocity. The volume of the ellipsoid is proportional to the manipulability at a given configuration. The longer the ellipsoid is along one of its axes, the more the robot can move in that direction and vice versa. If the configuration is singular the ellipsoid collapses into a planar ellipse as it loses one degree of freedom. If it loses another degree of freedom it can further collapse into a line.

The ellipsoids are shown here for three characteristic robot configurations. For the first one the robot arms are assembling an object directly in front of it, equally distanced from both arms. The second configuration is quite similar but the object is in front of the left arm base. The third is the initial configuration of the robot with its arms completely spread.

The results are shown in Fig. 8, 9 and 10. They are plotted in Matlab. The torso and the arms are shown as a kinematic chain and the ellipsoids are plotted at the arms' end effectors. The colors are again red for the left arm and blue for the right one.

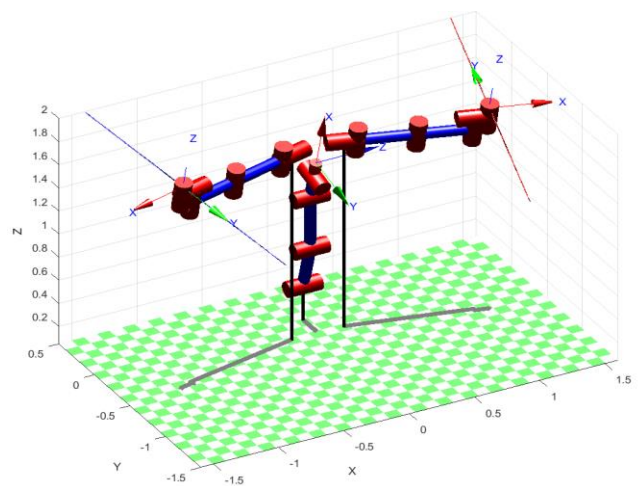


Fig. 8. Velocity ellipsoids in initial position.

The initial position is shown first and it is immediately clear that it is a singular configuration for the arms. Only lines are plotted meaning the robot loses two degrees of freedom and can only move in one translational direction.

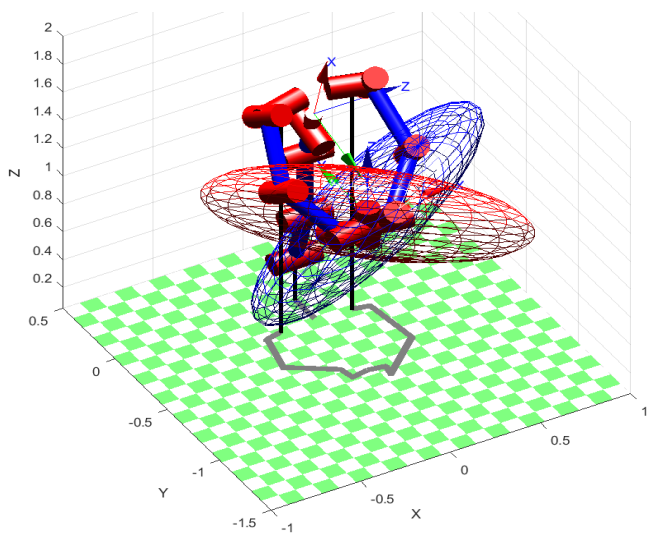


Fig. 9. Velocity ellipsoids in an assembly task in front of the torso

Fig. 9 shows the assembly position in front of the torso. The object is at an equal distance from both arms, and the ellipsoids are clearly much bigger, and equal to each other, indicating good manipulability.

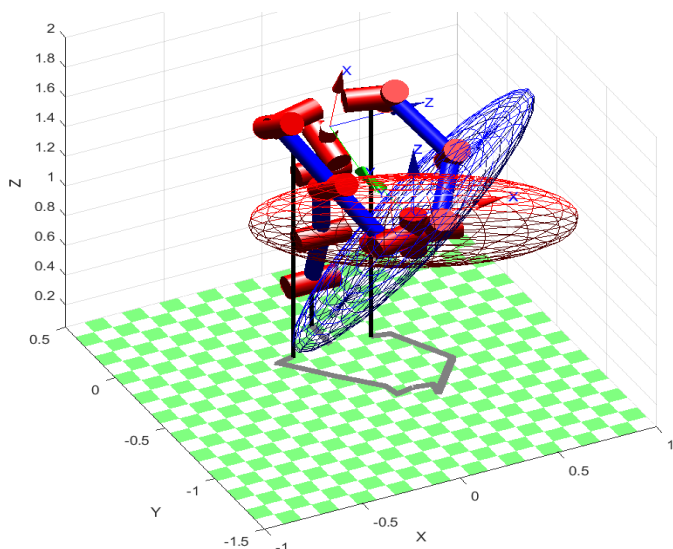


Fig. 10. Velocity ellipsoids in an assembly task in front of left arm base

Fig. 10 shows a similar task, this time in front of left arm basis. The shape of the left arm ellipsoid (red) remains similar while the right one is longer in one direction but quite shorter in another, which indicates a decreased manipulability for the right arm, which is expected.

IV. CONCLUSION

Simulation results have given a good overview of the workspace of a bi-manual robotic system. They have shown a large reachable area of the whole robot as well as a considerable shared workspace. Defining this workspace is very important as one of the early steps in developing a custom bi-manual robotic system. This information is

essential for configuring the robot's environments and for defining its various future tasks.

Manipulability analysis has identified problem areas within the reachable workspace as well as areas of good manipulability. This is important for defining the way the robot will perform its tasks and its optimal trajectories.

While a good indicator of the robot's manipulability, this analysis is not perfect. It only takes into account the kinematic properties of the robot, and it does not mean that all of the unmanipulable or relatively manipulable configurations will present a problem for the physical robot. Still, it does give a good overview of the areas that should be avoided.

This paper presented a starting work in the analysis of a custom bi-manual robotic system. A more detailed manipulability analysis, that takes into account the masses and inertias of the mechanism could be performed, and compared with previous results. The authors' future work will also consider various other aspects of bi-manual robotic systems such as different control strategies, trajectory planning, etc.

V. ACKNOWLEDGEMENTS

The results presented in the paper are obtained in the scope of the research projects: "Development and Experimental Performance Verification of Mobile Dual-Arms Robot for Collaborative Work with Humans", Science and Development Programme - Joint Funding of R&D Projects of the Republic of Serbia and the People's Republic of China, contract no. 401-00-00589/2018-09, 2018-2021 and national R&D project no. TR-35003, both supported by the Ministry of education, science and technology development of Republic Serbia.

REFERENCES

- [1] C. Smith, Y. Karayiannidis, L. Nalpantidis, X. Gratal, P. Qi, D. V. Dimarogonas, and D. Kragic, "Dual arm manipulation - a survey," *Robot. Auton. Syst.*, vol. 60, no. 10, pp. 1340–1353, 2012.
- [2] M. Fuchs et al., "Rollin' Justin - Design considerations and realization of a mobile platform for a humanoid upper body," 2009 IEEE International Conference on Robotics and Automation, 2009, pp. 4131–4137, doi: 10.1109/ROBOT.2009.5152464.
- [3] Z. Pilat, W. Klimasara, M. Pachuta, M. Słowikowski, "Some New Robotization Problems Related to the Introduction of Collaborative Robots into Industrial Practice" *J. Autom. Mob. Robot. Intell. Syst.* vol. 13, no. 4, pp. 91–97, 2020.
- [4] R.L. Williams II, "Baxter Humanoid Robot Kinematics", Internet Publication, April 2017, <https://www.ohio.edu/mechanical-faculty/williams/html/pdf/BaxterKinematics.pdf>.
- [5] ABB IRB 1400 Yumi Technical Data, May 2021, <https://new.abb.com/products/robotics/collaborative-robots/irb-14000-yumi/irb-14000-yumi-data>
- [6] EPSON Technical Data, May 2021, <https://www.epson.eu/robots/dual-arm>
- [7] A. Rodić, J. Šumarac, I. Stevanović, M. Jovanović, "Cloud-Enabled Bi-manual Collaborative Robot with Enhanced Versatility for Customized Production", Proceedings of the 30th International Conference on Robotics in Alpe-Adria-Danube Region, Futuroscope-Poitiers, France, pp. 240–249, June, 2021.
- [8] Universal Robots Technical Data, May 2021, <https://www.universal-robots.com/download/manuals-cb-series/service/service-manual-ur5-cb3-english/>
- [9] P. Corke, *Robotics, Vision and Control*, 2nd ed., Berlin, Germany, Springer, 2017.
- [10] T. Yoshikawa, "Manipulability of Robotic Mechanisms", *Int. J. Rob. Res.*, vol. 4, no. 2, pp 3-9, 1985.

Interface development dedicated to connecting CAD tools for 3D modeling of complex objects and industrial robot's controller

Uroš Ilić, Jovan Šumarac, Ilija Stevanović, Aleksandar Rodić

Abstract—This article covers the development of the interface dedicated to converting CAD models into some universal computer readable form – Excel spreadsheet. Main intention is transcription of complex models' geometry and coloring to numerical database. First the article will describe the process of saving CAD models in compatible form and its conversion to Microsoft Excel spreadsheets via a set of MATLAB's designated scripts and functions. Main focus is forming a set of significant points meant to guide the industrial robot's end effector to outline (with a laser pointer) the surface of the object that's in contact with the work desk. At the end, the given trajectory will be shown in MATLAB's plotter and correspondent Excel table.

Index Terms—CAD; MATLAB; Microsoft Excel; Path planning; Industrial robots

I. INTRODUCTION

MASS production as a leading example of XX century rapid development is slowly but consistently being replaced by mass customization. Its elementary feature is the use of flexible computer-aided manufacturing systems to produce custom output. Such systems combine the low unit costs of mass production processes with the flexibility of individual customization.

Even though integrating robots to manufacturing and assembly processes started long time ago, new problem opposes: How to program the robot in such way so production lines become more flexible in terms of fulfilling customer demands? For example, if it happens that there is a specific requirement on only one product, it is necessary to program the entire workstation in accordance with the requirement. After performing the task, it needs to be reprogrammed for the prior purpose. All this work is forming expenses, both the financial ones and also in form of human resources, i.e., the

Uroš Ilić is with the Faculty of Mechanical Engineering, University of Belgrade, 16 Kraljice Marije, 11020 Belgrade, Serbia (e-mail: ilicuros01@gmail.com)

Jovan Šumarac is with the Institute Mihajlo Pupin, University of Belgrade, 15 Volgina, 11020 Belgrade, Serbia (e-mail: jovan.sumarac@pupin.rs)

Ilija Stevanović is with the Institute Mihajlo Pupin, University of Belgrade, 15 Volgina, 11020 Belgrade, Serbia (e-mail: ilija.stevanovic@pupin.rs)

Aleksandar Rodić is with the Institute Mihajlo Pupin, University of Belgrade, 15 Volgina, 11020 Belgrade, Serbia (e-mail: aleksandar.rodic@pupin.rs)

engagement of engineers and/or programmers.

With the advent of factory digitalization and Industry 4.0, the whole world is moving towards factories with no or minimal number of workers. For this purpose, we strive for remote control of production systems, i.e., robots, both on production tasks and on assembly lines. With that in mind, it is necessary to develop such system where it is possible to control individual robots (manipulators). One of the conceptual solutions is forming a database within the cloud, which could be accessed simultaneously by the robot, which performs the assembly, and a programmer, who would work on updating the database itself.

II. OBJECTIVES OF CAD SYSTEMS AND ROBOTIC CONTROLLER INTEGRATION

Each of the designed parts will, firstly, be modeled in Solidworks® environment, and then exported with the appropriate extension. Such files are imported into MATLAB and, afterwards, exported to Microsoft Excel spreadsheets. This type of data storing is found convenient for later use by numerous programming languages such as C, C++, Python or MATLAB itself once again. For each of the imported models in MATLAB, a set of designated points in space with TCP's approach vector will be formed.

The final goal is to generate the trajectory of TCP with the given information, and thus solve the inverse kinematic problem of the manipulator. In other words, it is necessary to form a sequence of matrices of significant points' coordinates along with approach vector's x-y-z components, which will unambiguously determine the position of the robot's segments when moving, i.e., performing a certain technological operation.

All gathered information will be integrated into the database and the robot will, based on the data it receives from the camera, with the help of artificial intelligence, define which parts are in front of it, and in which position and orientation. Based on all of that, the robotic system will choose the appropriate technological operation of assembly or manufacturing process. Finally, with the given information, the robot can initialize, first the formation of the trajectory, and afterwards its realization.

III. USING SOLIDWORKS® SOFTWARE FOR DESIGNING COMPLEX OBJECTS

Solidworks® is a solid modeling computer-aided design (CAD) program that's suitable for designing numerous complex objects. In this particular case various mechanical components are modeled and used for development of the interface. Some of these objects are shown on Figure 1.

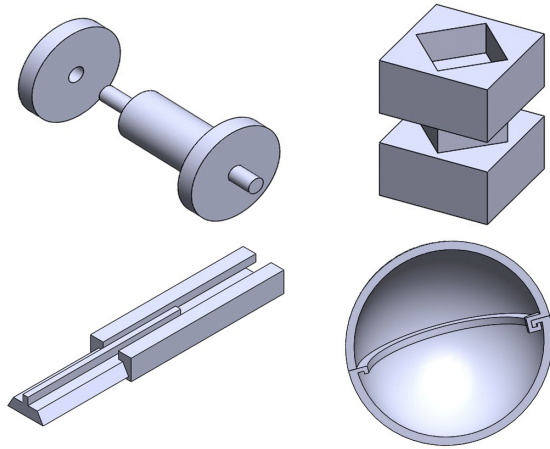


Fig. 1. Representation of typical complex objects found in the machine industry. These objects are chosen as they represent various kinds of assemblies, depending on interrelation of the parts.

The selected assemblies are some of the most common types in the machine industry and assembly processes. They are listed gradationally, so that simpler assemblies are described first. The simplicity of the assembly is observed from the aspect of the number of possible mutual positions that the parts can or must take to form an assembly.

A. Axle with bearings

This is one of the simplest mounting assemblies. In layman's terms, bearings can be found in infinitely many positions with the axle. This statement is justified by the axial symmetry of the parts that enter the assembly. It is necessary to achieve the appropriate positions immediately before the assembly process and perform a simple translation in order to form an assembly. Their model is shown in Fig. 1 in upper-left corner. Even though bearings need significant force to be mounted, this assembly is only used in the field of robot kinematics.

B. Simple puzzle

This form represents a more complex assembly than the previous one since only 4 mutual positions in the assembly are possible. They are achieved by a 90° rotation around the mounting axis. If the parts are not marked, the observer will not even be able to distinguish the different positions. Even though this type of the simplest puzzle is not typically found in the industry it's just a mere representation of assembly of prismatic shapes. It is shown in Fig. 1 in upper-right corner.

C. Linear guide rail

Classical example of a linear bearing, that is actually an assembly that can only be formed in a certain way. The guide and the slider must be brought to a precisely determined position and orientation in order for the assembly to be formed by a simple translation or inserting the slider into the guide. This assembly is shown on Fig. 1, bottom-left corner.

D. Spherical vessel

Unlike the other parts, it is not given in some of the standard views (isometry, trimetry ...). The given view (Fig. 1, bottom-right corner) is selected to show the grooves for closing and opening. This type of assembly is one of the most complex assemblies that can be found on the assembly line. It is necessary to bring both hemispheres to a position in which the tongue of the upper part coincides with the slit of the lower part. After connecting the hemispheres, it is necessary to rotate them by a certain angle and thus achieve the formation of the assembly, i.e., sealing the spherical vessel.

E. 3D puzzle pyramid

This type of assembly was taken into consideration as an example of the most complex technological assembly operation. The pyramid consists of 5 parts. It is necessary to assemble them in a specific order. An additional aggravating circumstance is the difficulty of distinguishing the parts of the pyramid itself. For this purpose, the outer sides of the parts are painted with various colors, while the inner sides are painted with the characteristic color of wood. The idea is to assemble the pyramid by a two-handed manipulator in some of the future works. Both assembled and disassembled pyramid is shown in Figure 2, respectively.

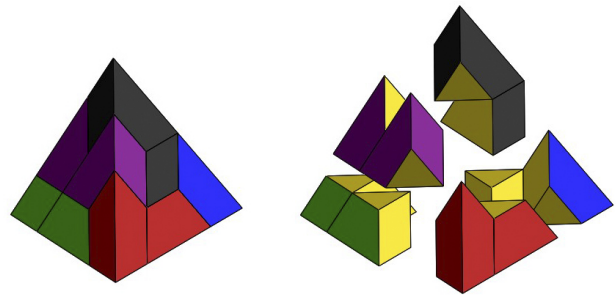


Fig. 2 – Comparison of assembled and disassembled pyramid. The complexity of parts' interrelationships can be seen on the figure.

IV. SURFACE TRIANGULATION, DATA EXPORT AND OBJECTS' DATABASE CREATION

In order for the parts and its CAD models to be readable by the robot's control unit, it is necessary that the geometry data of those parts is stored in numerical format. For this purpose, Microsoft Excel spreadsheet is chosen. Apart from being compatible with numerous programming languages, it is also easy to be reviewed by the user of the interface.

Of course, direct conversion from CAD models to Excel spreadsheets is not possible. Therefore, the MATLAB

software package will be used, where the procedures for the indirect conversion of parts from the Solidworks® environment into an Excel spreadsheet will be written. In order for all this to be feasible, it is necessary to export the CAD model in the appropriate format. Figure 3 represents the flowchart of this method.

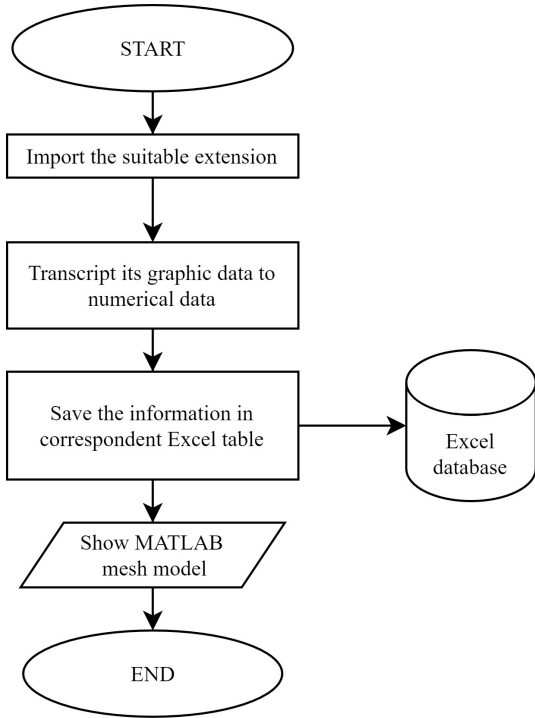


Fig. 3 – Flowchart representing the process of transferring the CAD model to the designated Excel spreadsheet with a mesh model’s representation in MATLAB plotter

Of all the possible extensions that Solidworks® supports, only two solutions are significant - STL format (.stl) and VRML format (.wrl). Although the selection of each of them entails certain advantages and disadvantages, that are presented in Table 1.

TABLE I
COMPARISON OF CHOSEN EXTENSIONS’ CHARACTERISTICS

Format (extension)	Advantages	Disadvantages
VRML (.wrl)	Ability to work with colors and their RGB vectors	Poorly researched topic; Lack of support on MATLAB forums
STL (.stl)	Relatively easy import to MATLAB; Very strong support within the MATLAB community	Able to store only geometry data, but not coloring of the faces

A. STL files

The abbreviation STL (Standard Triangle Language) represents a format suitable for 3D printing, i.e., stereolithographic methods in general. The format is characterized by triangulation of surfaces, i.e., division of surfaces into triangles. Figure 4 shows the process of forming a STL file.

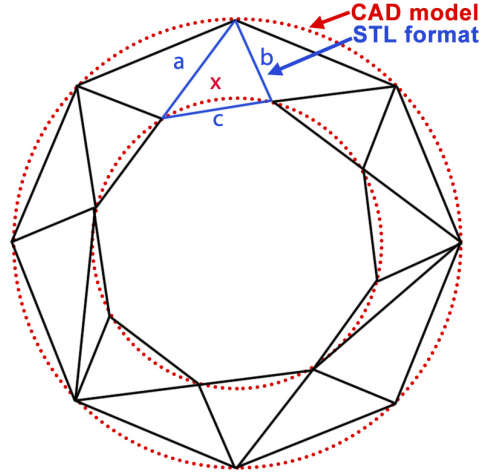


Fig. 4 – From the amorphous structure of the torus shape represented with dotted red line in the picture, a multitude of triangles is formed (one of which is marked in blue).

If it is amorphous, i.e., the more complex the surface, the greater the number of triangles in the network and vice versa. It is quite certain that simple geometric surfaces such as rectangles or squares will be divided into only two triangles, while circles and elliptical shapes will be divided into a finitely large number of triangles. Every STL file is characterized with connectivity matrix, coordinates of every point and each triangle’s normal vector. The connectivity matrix has dimensions $3 \times n$, where n represents number of triangles. This matrix defines what points form what triangle, respectively. Each triangle has its own corresponding normal vector represented with a unit vector.

B. VRML files

The VRML (Virtual Reality Modeling Language) format was created at the end of the last century with the intention of displaying three-dimensional vector graphics objects. The main emphasis was on the implementation of this format in internet communication, especially web presentations. These types of files (although their extension is *.wrl) are actually textually formatted and very easy to read by humans. Of course, by reading only a text file, one cannot create the body image that the file describes. Therefore, MATLAB will be used, which will read certain segments of the file, convert them into numerical values and form from them, first a mesh model, and then an Excel spreadsheet.

As mentioned in Table 1, VRML files have the ability to store information about faces’ coloring. Information about

coloring is stored in similar way like those for normal vectors. For each generated triangle a three-dimensional vector is formed that describes primary colors in additive color synthesis. Each component represents the amounts of red, green and blue, respectively.

C. Creating the database

After successfully exporting the models in suitable form, it's needed to import them in the MATLAB's environment. After loading the file's full name, the designated function makes the spreadsheet with the same name as the STL/VRML file. Given spreadsheet is formatted in such manner (merging cells, naming the titles, etc.) that is appropriate for following code to fill in the information about triangulation (connectivity matrix), coordinates, normal vectors and, regarding the VRML file, the coloring too.

When reading of the chosen format is done, designated script stores significant values in appropriate matrices and starts filling in the Excel spreadsheet. If chosen format is STL, the script uses modified version of downloaded function `stl_read()`, made by MATLAB community. On the other side, if it's needed to convert the VRML file, the script calls designated function `wrl_read()`, that, firstly convert's the VRML file to text file and extracts desired strings of text that contains information that's needed to fill in the Excel spreadsheet. These strings are converted into numerical form and saved as matrices, that, as in previous case, fill in the database.

In Figure 5 a typical Excel spreadsheet with CAD model's info is represented. The spreadsheet is cut in half over the J-column in order to fit the formatting of this article.

	A	B	C	D	E	F	G	H	I	J
1	Triangles					Vertices (coordinates)				
2	Number	Vertices (number of vertex)				Number	X	Y	Z	
3	1	7	2	9		1	0	-84	0	
4	2	9	2	5		2	0	0	0	
5	3	4	6	9		3	8.083	-70	22.862	
6	4	5	2	4		4	8.083	-42	22.862	
7	5	4	2	1		5	8.083	-14	22.862	
8	6	4	1	3		6	24.249	-42	0	
9	7	2	7	1		7	24.249	-14	0	
10	8	1	7	6		8	32.332	-56	22.862	
11	9	1	6	10		9	32.332	-28	22.862	
12	10	9	6	7		10	48.497	-56	0	
	J	K	L	M	N	O	P	Q	R	S
1	Normal vectors (unit vectors)					Triangles' colors				
2	Number	X	Y	Z		Number	R	G	B	
3	1	0.471405	0.816497	0.333333		1	1	0.784314	0	
4	2	0.471405	0.816497	0.333333		2	1	0.784314	0	
5	3	0.471405	-0.816497	0.333333		3	1	0.784314	0	
6	4	-0.942809	0	0.333333		4	1	0.784314	0	
7	5	-0.942809	0	0.333333		5	1	0.784314	0	
8	6	-0.942809	0	0.333333		6	1	0.784314	0	
9	7	0	0	-1		7	1	0.784314	0	
10	8	0	0	-1		8	1	0.784314	0	
11	9	0	0	-1		9	1	0.784314	0	
12	10	0.942809	0	-0.333333		10	1	0.784314	0	

Fig. 5 – The appearance of the Excel spreadsheet that originated from model's VRML file. This screenshot shows data only for first 10 triangles, even though the number of triangles can be measured in thousands for amorphous surfaces.

Here, a user can see from what vertices the triangles are format, e.g., the vertices numerated as 7,2 and 9 form the first triangle. Their coordinates in local coordinate system can be read in the neighboring table. Components of the normal unit vector for the first triangle can be read from cells L3 to N3 and its RGB coloring vector can be found in cells Q3:S3. Note that this spreadsheet came from conversion of the VRML file, since it has information about the coloring. The spreadsheet formed from STL files is practically the same without the last (fourth) table describing the colors.

Note that the tables always start from the same cell, no matter what model is being converted. Only change is number of rows in the Excel file. This feature enables making the universal MATLAB function for reading Excel spreadsheets and storing the values in corresponding matrices.

V. GENERATING MESH MODELS FROM 3D DATABASE

As a verification of the stored data, it's needed to read the spreadsheet and visualize that numerical information with a mesh model. If and only if the mesh model made by MATLAB plotter is same as initial CAD model, the code can be verified.

Firstly, the code calls the Excel file with desired name and starts reading the data from the tables. For every table read, it stores the values in corresponding matrix. After that, an already built-in functions form a three-dimensional mesh from the connectivity list (first table in the spreadsheet) and corresponding coordinates of given vertices (second table in the spreadsheet). Afterwards, each normal unit vector is drawn with the origin in triangle's centroid. If we're working with the STL files, all triangles will be painted with the same color. On the other side, if the user is converting the VRML file, each triangle will be colored with their correspondent RGB vector. Figure 6 shows the CAD model of an axle on the left and the result of reading the spreadsheet on the right. This is the result of converting the STL files.

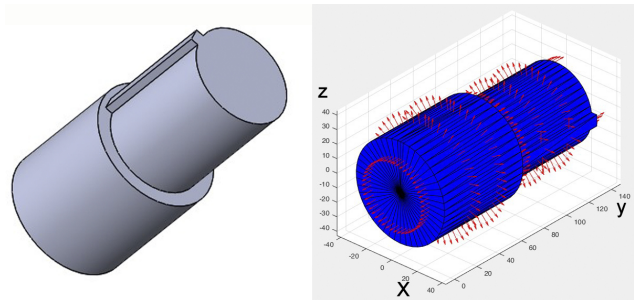


Fig. 6 – The result of the conversion of the CAD model to Excel spreadsheet and back again to MATLAB's mesh model. Coloring of the mesh model can be changed to any color, but whole model must be colored uniformly

On the other side, if we chose to export the CAD model as a VRML file, it will store its original coloring and represent it in its original colors in MATLAB plotter. Figure 7 shows a mesh model with surface vectors of 3D puzzle pyramid's red part in its original coloring. As it was said earlier the parts' outer surface is colored differently, but their inner sides depict wood's natural coloring.

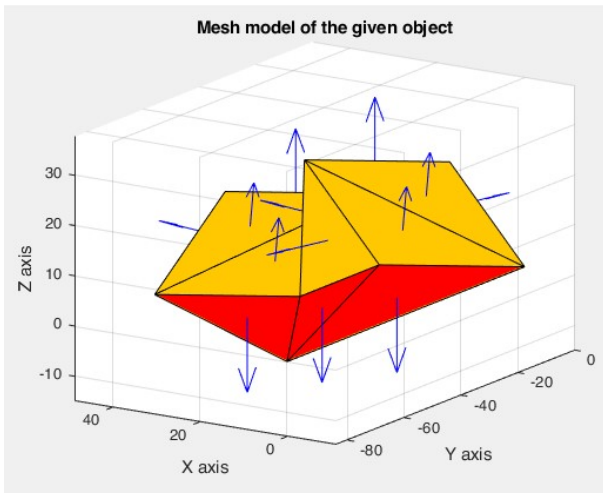


Fig. 7 – 3D puzzle pyramid’s corresponding mesh model of the red part shown in MATLAB plotter. Outer face is colored with red and inner ones are with color of wood, just like we colored them in Solidworks®. Blue arrows pointing outwards represent normal unit vectors of each triangle.

VI. DEFINING THE CONTOUR OF THE OBJECT AND CORRESPONDING SIGNIFICANT POINTS

The procedure for defining the path that will enable the TCP to outline the surface lying on the desk is given on the figure 7 in form of the flow chart. This flowchart represents the second part of the whole process – the extraction of the numerical data of modeled parts.

The first step is to acquire the information of the object on the work desk. The camera will recognize the object lying on the desk and its position and orientation. Let’s say that camera recognized the object on Figure 7. With the help of artificial intelligence, desired object’s information will be found in the database. With the knowledge of position of the object in global, i.e., robot’s coordinate system, object’s local coordinate system will be transformed. Hence the vertices in contact with the desk will change. Therefore, we’ll know the exact vertices lying on the desk, and their coordinates in global coordinate system.

The problem occurs if the shape of the face is oval or circle-shaped. Since the triangulation transforms the circle into a polygon, the center of the circle becomes the vertex that’s included in all triangles. Even though it has its z-coordinate equal to zero, we must eliminate it too in order to form a closed polygon.

With the points extracted from the Excel spreadsheet, the 2D polygon of the surface is made. On Figure 8 the polygon is drawn with red lines with normal vectors pointing outside. The face represented on the Figure is actually the outer side of the puzzle’s part – the red side.

When the polygon representing the contour of the object lying on the work desk is constructed, we advance to forming a trajectory of the tip of the end-effector of the robot. As said earlier, the laser pointer will be attached to the end-effector that will outline the contour of the object. In order to do such a thing, we must make another polygon with desired offset from the edges of the initial polygon.

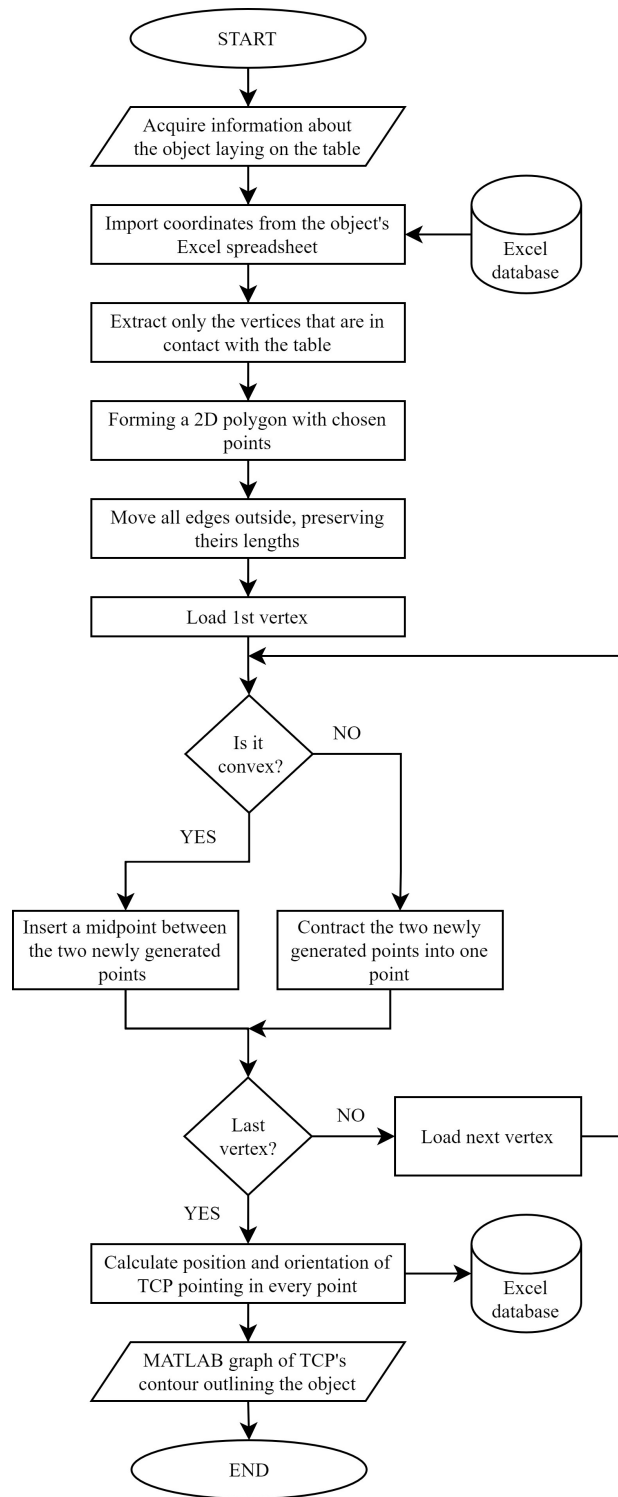


Fig. 7 – A flowchart depicting the process of the creation of significant points of TCP’s path outlining the lying surface of the given object

The newly formed polygon is actually a modified enlargement of the surface’s contour with rounded edges. It is constructed in a *for*-loop that goes over every point and checks if it’s a convex or concave vertex. The whole

process of creating the rounded enlarged polygon from the initial one can be described with the following steps:

- First, every edge is shifted outwards for desired distance. Now, instead of closed polygon, we have a set of disconnected lines, but with preserved lengths.
- For every vertex of the contour, algorithm checks whether they're convex or concave
- If they are convex, a new middle point is added, so that end effector outlines the convex vertex in a radius
- If, on the other hand, the vertex is concave, two significant points are replaced with one and therefore radius is made again in order to avoid collision.

After this part of the algorithm completes the whole contour and the last vertex is formed, the potential trajectory of the TCP is plotted in 2D diagram, as represented in Figure 8.

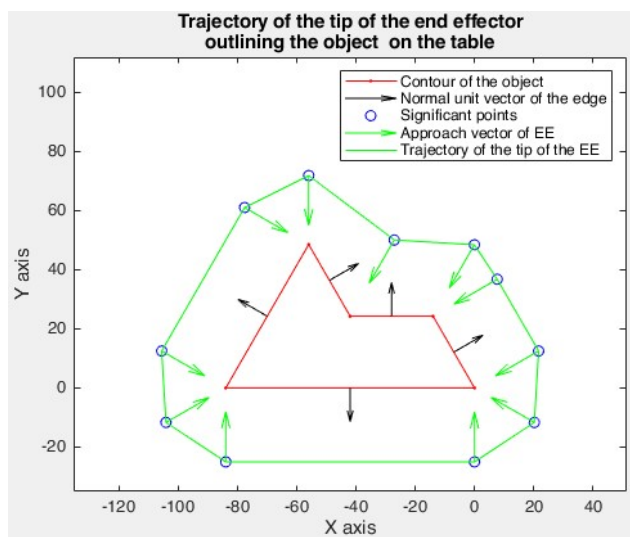


Fig. 8 – Newly generated trajectory of the end effector. It can be seen that one of the vertices is concave, hence a collapsing of neighboring points occurred. It can also be seen that the edges have been shifted outwards in the direction of normal unit vector.

In every, newly generated, significant point, the end effector's approach vector is computed. Approach vector will always have direction from the significant point to the correspondent point within the contour. Every point's position and approach vector's components are then stored in an Excel spreadsheet, as shown on Figure 9.

	A	B	C	D	E	F	G
1	Points	TCP Position			Approach vector		
2		X	Y	Z	X	Y	Z
3	1	0.00000	-25.00000	43.30127	0.00000	0.50000	0.86603
4	2	20.20032	-11.66273	43.30127	-0.43301	0.25000	0.86603
5	3	21.65070	12.49989	43.30127	-0.43301	-0.25000	0.86603
6	4	7.65070	36.74889	43.30127	-0.43301	-0.25000	0.86603
7	5	-0.04962	48.41162	43.30127	-0.25000	-0.43301	0.86603
8	6	-27.11194	50.03582	43.30127	-0.25000	-0.43301	0.86603
9	7	-56.00010	71.82244	43.30127	0.00000	-0.50000	0.86603
10	8	-77.65059	60.99708	43.30127	0.43301	-0.25000	0.86603
11	9	-105.65059	12.50008	43.30127	0.43301	-0.25000	0.86603
12	10	-104.20031	-11.66261	43.30127	0.43301	0.25000	0.86603
13	11	-84.00000	-25.00000	43.30127	0.00000	0.50000	0.86603

Fig. 9 – The look of the Excel spreadsheet containing the information about TCP's position (left part) and end-effector's approach vector outlining the object's surface on the table. The Z coordinates are the function of the desired angle of the laser's beam.

VII. CONCLUSION

Although the given example is a simple problem, it represents the new way of transcribing and eventually storing the numerical representation of complex geometry objects. This way of storing the data (in Excel spreadsheets) is chosen because it is the best way for a human eye to examine the data. This data can be stored in any file that is suitable for reading as long as the programmer knows how to store/look/extract the information.

It is clear that this process doesn't give homogeneous transformational matrices for every significant point. The idea is to store information in that way that can be easily retrieved by robot's controller or any other computing unit meant for simulation or robot control.

ACKNOWLEDGMENT

The results presented in the paper are obtained in the scope of the research projects: „Development and Experimental Performance Verification of Mobile Dual-Arms Robot for Collaborative Work with Humans“, Science and Development Program – Joint Funding of R&D Projects of the Republic of Serbia and the People's Republic of China, contract no. 401-00-00589/2018-09, 2018-2021 and national R&D project no. TR-35003, both supported by the Ministry of education, science and technology development of Republic Serbia.

REFERENCES

- [1] Peddie, Jon, *The history of visual magic in computers: how beautiful images are made in CAD, 3D, VR and AR*, 1st ed, London, Springer, 2013
- [2] Tseng, M.M.; Jiao, J. (2001) "Mass Customization", in: *Handbook of Industrial Engineering, Technology and Operation Management* New York, NY: Wiley, 2001, 978-0-471-33057-8
- [3] J. J. Uicker, G. R. Pennock, and J. E. Shigley, *Theory of Machines and Mechanisms*, New York, Oxford University Press, 2003
- [4] Paul Festa and John Borland, "Is a 3D web more than just empty promises?". CNET News.com, May 19, 2005.
- [5] P. Corke, *Robotics vision and control. Fundamental algorithms in MATLAB®*, 3rd ed, London, Springer, 2011

A Mobile Robot Visual Perception System based on Deep Learning Approach

Aleksandar Jokić, Lazar Đokić, Milica Petrović and Zoran Miljković

Abstract—In this paper, we present the novel mobile robot perception system based on a deep learning framework. The hardware subsystem consists of an Nvidia Jetson Nano development board integrated with two parallelly positioned Basler daA1600-60uc cameras, while the software subsystem is based on the convolutional neural networks utilized for semantic segmentation of the environment scene. A Fully Convolutional neural Network (FCN) based on the ResNet18 backbone architecture is utilized to provide accurate information about machine tool models and background position in the image. FCN model is trained on our custom-developed dataset of a laboratory model of manufacturing environment and implemented on mobile robot RAICO (Robot with Artificial Intelligence based COgnition).

Index Terms—Deep learning; Perception System; Mobile robot; Semantic Segmentation.

I. INTRODUCTION

Modern mobile robot sensors (e.g., cameras or lidars) provide a rich amount of data about the current state of the environment. However, the way the data is interpreted and transferred into useful information has been an active area of research in the last two decades. Deep learning, or more precisely, Convolutional Neural Networks (CNNs), represent one of the most promising methodologies that can enable mobile robots to understand and interact with their environment in a more sophisticated manner [1]. The main disadvantage of CNNs is the requirement for a substantial amount of computation power for real-time implementation. Fortunately, several modern single-board computers or hardware accelerators provide enough computing power to deploy low-weight CNNs for real-time implementation.

The authors [2] developed the visual perception system

Aleksandar Jokić, PhD student, University of Belgrade - Faculty of Mechanical Engineering, Department of Production Engineering, Laboratory for industrial robotics and artificial intelligence (ROBOTICS&AI), Kraljice Marije 16, 11120 Belgrade 35, The Republic of Serbia (ajokic@mas.bg.ac.rs).

Lazar Đokić, PhD student, University of Belgrade - Faculty of Mechanical Engineering, Department of Production Engineering, Laboratory for industrial robotics and artificial intelligence (ROBOTICS&AI), Kraljice Marije 16, 11120 Belgrade 35, The Republic of Serbia (ldjokic@mas.bg.ac.rs).

Dr. Milica Petrović, Assistant Professor, University of Belgrade - Faculty of Mechanical Engineering, Department of Production Engineering, Laboratory for industrial robotics and artificial intelligence (ROBOTICS&AI), Kraljice Marije 16, 11120 Belgrade 35, The Republic of Serbia (mmpetrovic@mas.bg.ac.rs).

Dr. Zoran Miljković, Full Professor, University of Belgrade - Faculty of Mechanical Engineering, Department of Production Engineering, Laboratory for industrial robotics and artificial intelligence (ROBOTICS&AI), Kraljice Marije 16, 11120 Belgrade 35, The Republic of Serbia (zmiljkovic@mas.bg.ac.rs).

utilized for navigation in the indoor environment based on CNNs for scene classification. They deployed shallow CNN to achieve real-time mobile robot navigation within the environment in both dynamic and static conditions. The development of the CNN model capable of determining the 3D physical properties of objects in the scene is presented in [3]. The authors propose using the learned properties to predict the outcome of the dynamic events in the environment. This type of perception system can be beneficial for mobile robots employed in highly dynamic environments. The authors of [4] developed a complex cleaning mobile robot perception system with two submodules, one based on Bayesian filtering of data from 2D lidar, 3D lidar, and RGB-D camera used for human detection and tracking, and the second one for obstacle and dirt detection based on two RGB-D cameras and 3D lidar sensor. In [5], the authors developed a perception system based on a monocular camera for data acquisition and SURF point feature extraction method integrated with neural extended Kalman filter for simultaneous localization and mapping of mobile robot's position and orientation. Another promising methodology for developing perception systems for mobile robots is a cooperative perception [6], where the perception of one autonomous agent depends on the perception of other nearby agents. Camera information obtained by one mobile robot (agent) can be propagated to the others if their relative pose is estimated well. Moreover, the authors implemented the camera models that share information regarding pedestrian detection. The mobile robot developed with the aim to be applied for manufacturing purposes was proposed in [7]. Safe indoor navigation is provided by the perception system based on a 2D camera, 3D camera, laser scanner, ultrasonic sensors, and internal measurement unit. On the one side, safe navigation is provided by a laser scanner and ultrasonic sensors, while a 3D camera is used for the correction step in the Kalman filter state estimation. Moreover, the 2D camera is utilized for adaptive path planning by detecting lines on the ground.

This work presents the novel semantic segmentation-based mobile robot perception system implemented on our own developed mobile robot RAICO. The modified ResNet18 backbone architecture is integrated with RAICO's sensory subsystem, which contains two parallelly positioned Basler daA1600-60uc cameras. The proposed perception system provides RAICO with the ability to safely navigate the laboratory model of the manufacturing environment by knowing the position of machine tools in the image plane.

The paper is structured as follows. Section two is devoted to a thorough explanation of the considered CNN model and its training procedure. The third Section describes the experimental results and perception system evaluation while concluding remarks are presented in the fourth Section.

II. MOBILE ROBOT PERCEPTION SYSTEM

The mobile robot perception system is developed with two parallelly mounted Basler daA1600-60uc cameras facing downwards with an inclination angle of 30° and a baseline of 12.5 cm. The combination of mentioned cameras with *Evetar M118B0418W* lenses (4 mm focal length) provides a large angle-of-view of the scene, approximately $W \times H = 84^\circ \times 68^\circ$. The cameras are connected via USB3.0 to the Jetson Nano development board for image acquisition. The whole perception system is positioned on the top of the mobile robot RAICO (Fig. 1).

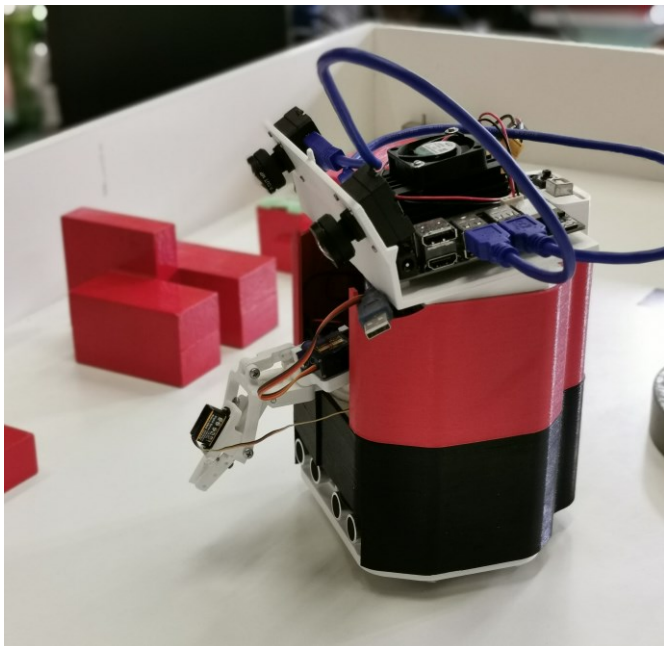


Fig. 1. Mobile robot RAICO with assembled perception system

The main component of the perception system is the Fully Convolutional neural Network (FCN) used for semantic segmentation. To enable the processing in near real-time, the selected backbone is ResNet18 based network. The network is trained by utilizing our custom-developed dataset for semantic segmentation. Image data is acquired within the laboratory model of a manufacturing environment, while the segmentation masks are hand-labeled. The dataset consists of densely labeled images with four machine tool classes and the background class. The dataset contains 125 images divided for training and testing in the 80/20 ratio. The sample of the dataset is shown in Fig. 2.

Hard data augmentation is carried out on the images used for training to improve neural network generalization. Mobile robot RAICO uses real-world noise-prone cameras that can significantly impact the accuracy of neural networks [8].

Therefore, the first augmentation is performed with Gaussian noise added to the images. Two Gaussian noise levels have been introduced for all the images used for training. The first level contains the noise with zero mean and variance in the range of 0.002-0.004, and the second one has a variance of 0.002-0.011 (sample of image with Gaussian noise is shown in Fig. 3).

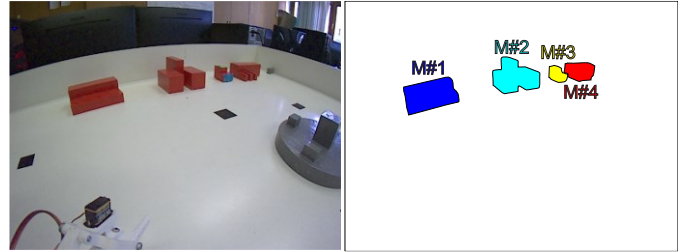


Fig. 2. Sample of the custom-developed dataset

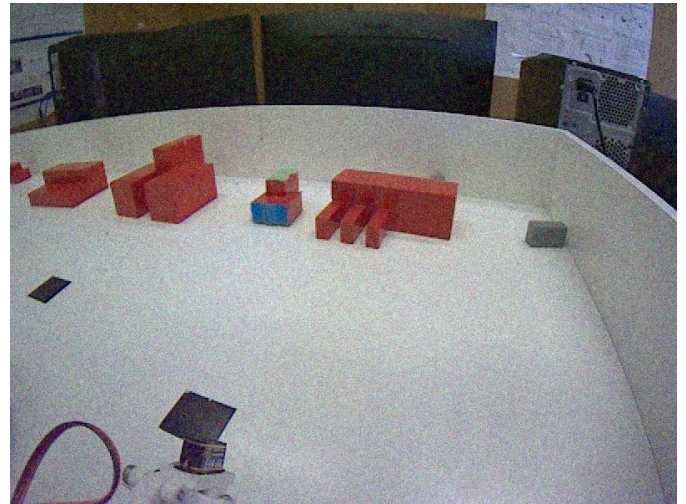


Fig. 3. Image with Gaussian noise

Besides the Gaussian noise procedures, we also applied three data augmentation procedures: (i) horizontal flips (10% of the images), (ii) random crops with a scale of 0.7 (10% of the images), and (iii) the complete image pixel intensities change in the range of 0.8-1.2 (10% of images); this procedure results in images with different illumination intensity levels, which can realistically occur during the different parts of the day. After data augmentation is done, the considered dataset contains 370 images used for training neural networks.

The details about utilized architecture are presented in Fig. 4. Different blocks of layers are presented with rectangles of different colors, while the parameters for those layers are presented within the rectangle. W represents the weight matrix dimensions, S is the stride value, and P represents the padding value. Convolution, BatchNormalization, and ReLU layers are presented with blue blocks. The green block presents the MaxPooling layer, while the convolution and BatchNormalization block is presented with the brown

rectangle. Finally, the adding layer in combination with the ReLU activation layer is presented with orange. Input images have $800 \times 600 \times 3$ resolution, while the output semantic mask has the dimension of 19×25 . The probabilities of the class prediction are calculated by utilizing the Softmax activation function (1), while the utilized loss function is Cross-entropy (2).

$$s_i = \frac{e^{y_i}}{\sum_{i=1}^N e^{y_i}} \quad (1)$$

$$\ell(\mathbf{s}, \mathbf{c}) = -\sum_i^N c_i \log(s_i) \quad (2)$$

Where \mathbf{y} represents the output vector of the neural network, i is the current element of the output vector, N is a total number of classes (and the number of elements in the output vector), s_i is the output of the softmax function for each element, \mathbf{c} represents one-hot vector for the correct class of the current input vector, and ℓ represents the loss function value.

The training is carried out by PyTorch v1.6.0 with Stochastic gradient descent and the momentum of 0.9. The initial learning rate is $\eta=0.01$ with the changing schedule defined with (3):

$$\eta^{\text{new}} = \eta^{\text{old}} \cdot \left(1 - \frac{\text{current_epoch}}{\text{max_epoch}}\right)^{0.9} \quad (3)$$

It is important to note that `current_epoch` is enumerated from 0 to `(max_epoch - 1)`. For the experimental research presented in this paper, the maximum number of epochs is 30, while the mini-batch size is 4. Lastly, the regularization technique is utilized with a weight decay of 0.0001. Training is performed on Nvidia RTX 1660 GPU with 6GB of RAM.

Since the Nvidia Jetson nano is an edge device with limited processing power (NVIDIA 128-core Maxwell GPU), the whole FCN network with encoder and decoder parts could not be implemented in real-time. Therefore, the authors propose to maintain the output of the backbone network and directly calculate the semantic mask with the output resolution instead of deconvolving that information to acquire prediction with the same resolution as input. Having that in mind, the output mask is considerably smaller in resolution than an input image. However, the achieved accuracy is entirely satisfactory.

III. EXPERIMENTAL RESULTS

We have trained the FCN-ResNet18 model on our custom-developed dataset for semantic segmentation of laboratory model of a manufacturing environment. Moreover, the trained model is implemented in the perception system on the mobile robot RAICO. Two metrics utilized to analyze the generalization performance of the FCN-ResNet18 model are Global accuracy and Intersection over Union (IoU). The training results for each class, as well as for the whole dataset,

are presented in Table I.

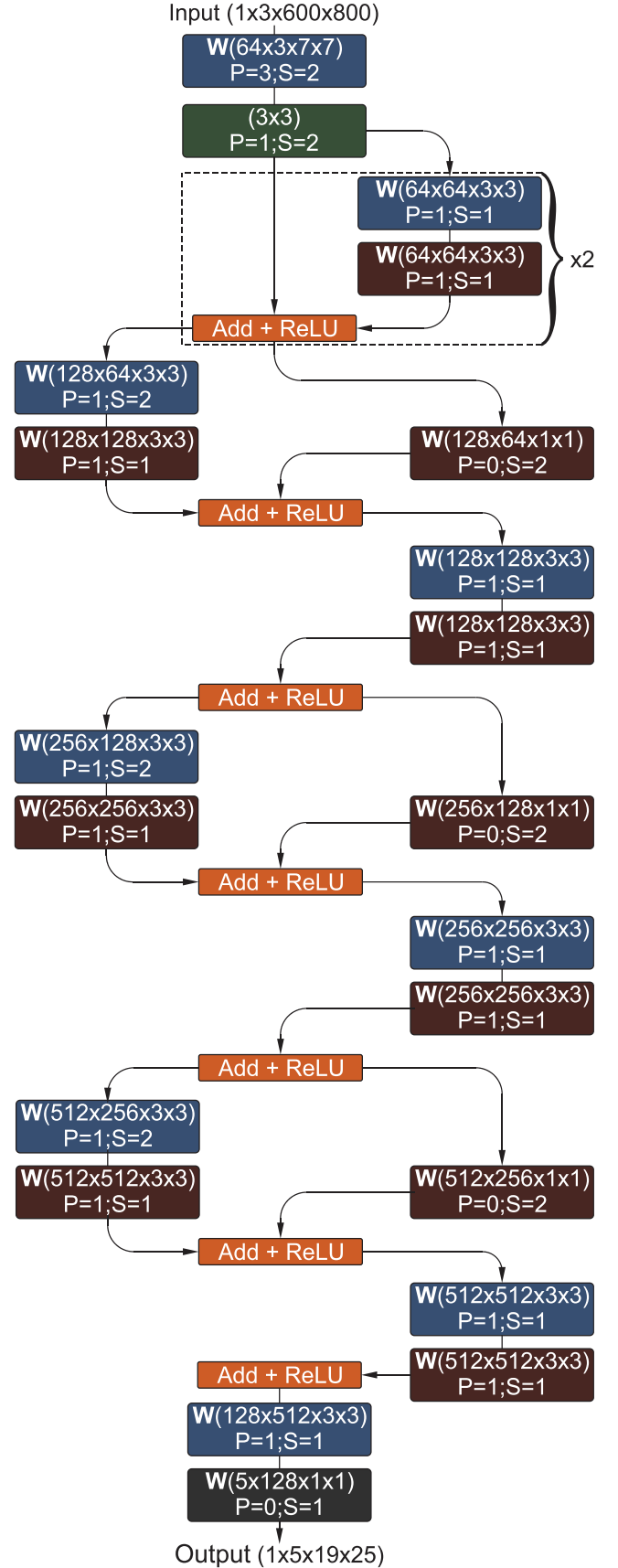


Fig. 4. Architecture of the FCN-ResNet18 model

TABLE I
THE EXPERIMENTAL RESULTS OF THE SEMANTIC SEGMENTATION MODEL

Accuracy measures	Background	M#1	M#2	M#3	M#4
Global per-class accuracy [%]	96.8	68.9	88.6	91.4	94.4
Per-class IoU	96.1	58.2	69.4	45.9	58.6
Mean global accuracy =96.0		Mean IoU = 65.6			

As shown in Fig. 5, there is a significant class imbalance in the considered dataset, as in most semantic segmentation datasets. The dominant class in the images is the background.

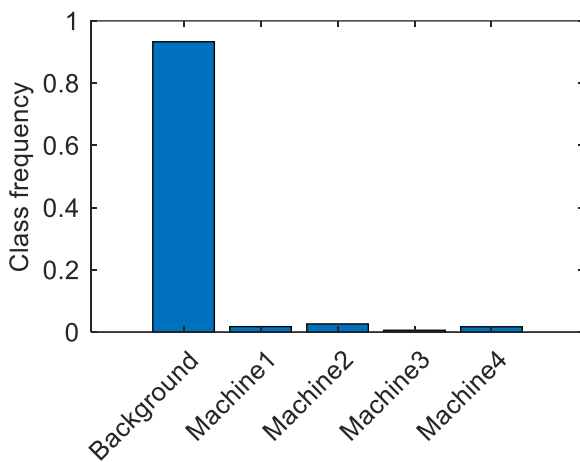


Fig. 5. Class frequency in the custom dataset

Having that in mind, the highest accuracy is achieved for the class with most samples, even though the authors have added the class weights that are inversely proportional to the class frequencies. Moreover, the worst results (for IoU metric) are achieved for Machine3 (M#3) since it is the smallest machine and therefore occupies the smallest percentage of the scene. Interestingly, global accuracy for M#1 is the smallest compared to all the other classes. The authors further investigated this occurrence and presented the overlay view of two test images and their semantic masks generated by the FCN network (Fig. 6). As it can be seen, the network misclassified half of the M#1 in the first image in Fig. 6. Furthermore, in the image with occlusions, part of the M#1 is misclassified and labeled as M#4.

Achieved mean global accuracy is 96.0%, which is a promising result; however, the mean IoU measure of 65.6 is much more representative of the actual generalization capabilities of the FCN model.

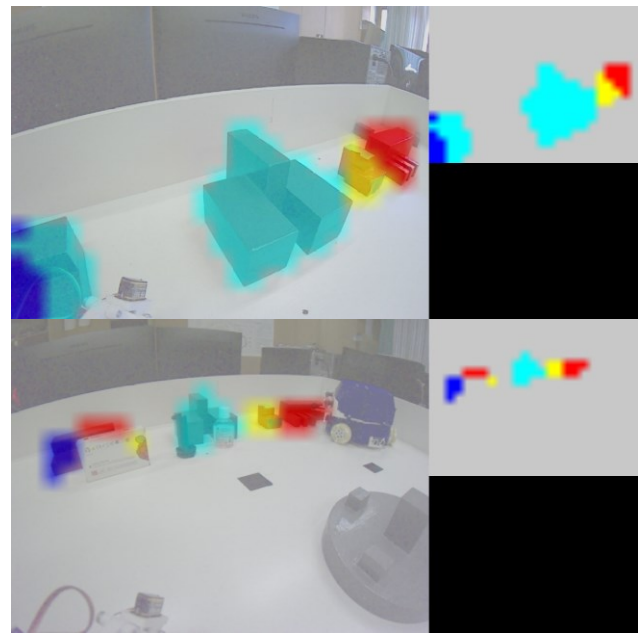


Fig. 6. Test images overlaid with semantic maps

To further test the accuracy of the trained network, the model is implemented on mobile robot RAICO and tested online by the real-time acquisition of images and their semantic segmentation. Fig. 7 presents few images acquired and segmented by the FCN model in a real-world scenario. To increase the effectiveness of the FCN model, it is transformed to an ONNX format and optimized by utilizing Nvidia TensorRT.

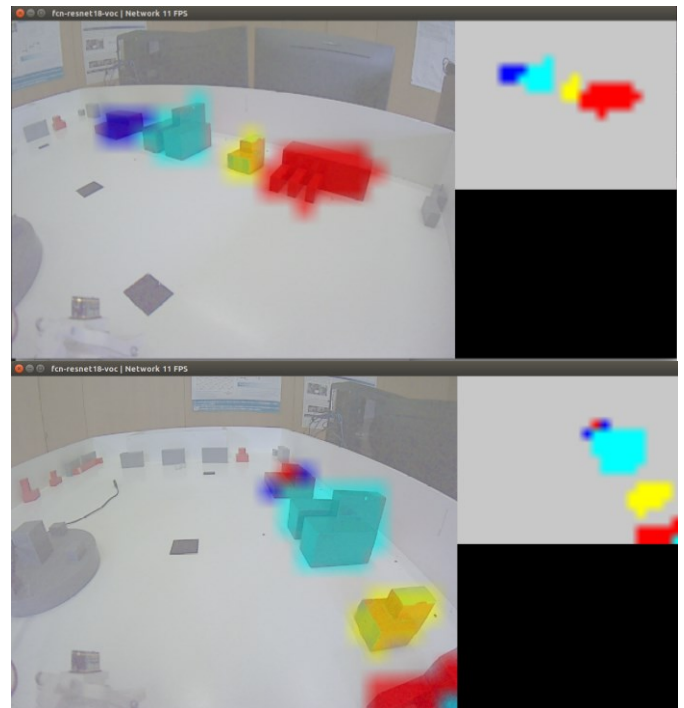


Fig. 7. Testing of implemented FCN model

From Fig. 7, it can be seen that the considered FCN model achieves acceptable accurate segmentation results, with minor errors on machines that are either far away, occluded by other machines, or only partially visible. Furthermore, the model is implemented with 11FPS, which is acceptable for a mobile robot with low-velocity profiles.

IV. CONCLUSION

This paper proposes the new perception system of mobile robot RAICO based on a Fully Convolutional neural Network with ResNet18 backbone architecture. Training of the neural network model is carried out on a custom-developed dataset for semantic segmentation of the laboratory model of the manufacturing environment. The perception system is integrated with the Nvidia Jetson Nano development board and two Basler dart cameras and configured as a standalone device. After the training procedure is completed, the model is implemented on the mobile robot RAICO, with the achieved accuracy measures of 65.6 for mean IoU and 96.0 for the global accuracy. The implemented system works in a near real-time manner achieving approximately 11FPS. Future research directions could include creating a larger dataset with more classes of manufacturing entities, as well as developing a novel, faster architecture for semantic segmentation capable of running real-time on Jetson nano.

ACKNOWLEDGMENT

This work has been financially supported by the Ministry of Education, Science and Technological Development through the project "Integrated research in macro, micro, and nano

mechanical engineering – Deep learning of intelligent manufacturing systems in production engineering" (contract No. 451-03-9/2021-14/200105), and by the Science Fund of the Republic of Serbia, grant No. 6523109, AI – MISSION 4.0, 2020 – 2022.

REFERENCES

- [1] J. Shabbir and T. Anwer, "A survey of deep learning techniques for mobile robot applications," *arXiv Prepr. arXiv1803.07608*, 2018.
- [2] T. Ran, L. Yuan, and J. B. Zhang, "Scene perception based visual navigation of mobile robot in indoor environment," *ISA Trans.*, vol. 109, pp. 389–400, 2021.
- [3] J. Wu, I. Yildirim, J. J. Lim, B. Freeman, and J. Tenenbaum, "Galileo: Perceiving physical object properties by integrating a physics engine with deep learning," *Proc. Advances in neural information processing systems (NIPS 2015)*, vol. 28, pp. 127–135, 2015.
- [4] Z. Yan, S. Schreiberhuber, G. Halmetschlager, T. Duckett, M. Vincze, and N. Bellotto, "Robot perception of static and dynamic objects with an autonomous floor scrubber," *Intell. Serv. Robot.*, vol. 13, pp. 403–417, 2020.
- [5] Z. Miljković, N. Vuković, M. Mitić, and B. Babić, "New hybrid vision-based control approach for automated guided vehicles," *Int. J. Adv. Manuf. Technol.*, vol. 66, no. 1–4, pp. 231–249, 2013.
- [6] S. Sridhar and A. Eskandarian, "Cooperative perception in autonomous ground vehicles using a mobile-robot testbed," *IET Intell. Transp. Syst.*, vol. 13, no. 10, pp. 1545–1556, 2019.
- [7] J. Qian, B. Zi, D. Wang, Y. Ma, and D. Zhang, "The design and development of an omni-directional mobile robot oriented to an intelligent manufacturing system," *Sensors*, vol. 17, no. 9, Article Number: 2073, 2017.
- [8] D. M. Chan and L. D. Riek, "Object proposal algorithms in the wild: Are they generalizable to robot perception?," *Proc. 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2601–2607, 2019.

Development of applicative interface for connecting optical 3D scanner and robot controller of the UR-5 industrial robot arm

Emilija Stanković, School of Electrical Engineering, University of Belgrade
Ilija Stevanović, Institute Mihajlo Pupin, Robotics department
Aleksandar Rodić, Institute Mihajlo Pupin, Robotics department

Abstract— This paper is focused on the implementation, development and testing of an interface for connecting an automated 3D scanner and an industrial robot. Scanning is based on the use of Structured Light scanning software and the accompanying equipment. A rotating platform is constructed, and it is powered by an Arduino-controlled stepper motor. Program in addition to rotating platform by optimal number of degrees, also communicates with the scanning software - via the COM port to synchronize the movement of the rotating platform and the scan. 3D image is obtained by connecting several captured frames during the rotation of the object. The result is point cloud in space, post-processing is performed, and selected points form a robot trajectory. Simulation is performed by MATLAB Robotics Toolbox.

Index Terms— Structured Light scanning, Robotics System Toolbox, UR-5, Point cloud

I. INTRODUCTION

Spatial visualization through realistic 3D modeling is the process of obtaining a mathematical representation of the three-dimensional surface of an object. 3D scanning is widely used in all spheres of industry because it provides fast data collection, and it guarantees a high quality. 3D scanners have the ability to capture in detail points, lines and surfaces of objects, as well as texture and size.

3D scanning is transforming nearly every industry, construction, manufacturing, marketing, medical industry, forensics, reverse engineering in areas of automotive, aerospace and shipbuilding. It has a multitude of applications in robotic vision and guidance, some of them being AI applications, art, design, interior visualization, media, machine control, site layout, flight and wind tunnel testing, prototyping and 3D printing, quality control, and scanning of power equipment assembling of a substation.^[1]

Emilija Stanković is with the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia (e-mail: emilijazstankovic@gmail.com)

Ilija Stevanović is with the Institute Mihajlo Pupin, University of Belgrade, 15 Volgina, 11020 Belgrade, Serbia (e-mail: ilija.stevanovic@pupin.rs)

Aleksandar Rodić is with the Institute Mihajlo Pupin, University of Belgrade, 15 Volgina, 11020 Belgrade, Serbia (e-mail: aleksandar.rodic@pupin.rs)

With that in mind, incorporating 3D scanning technology and robotic manipulators has the potential to facilitate all spheres of industry, from simple things like manufacturing to complicated operations like robotic surgery. Different approaches to using 3D scanners as robotic vision systems have been proposed in the literature.^{[2][3][4]} In industry, robotic systems are taught by guidance to perform specific tasks on object. When changing the object that the manipulator needs to interact with it is necessary to repeat the process of teaching guidance every time. With the introduction of 3D scanning this problem has been avoided and flexibility precision and repeatability are increased. Nowadays, robotic surgery occupies an important place when it comes to the advancement of technology and is being applied more and more every day. Some of the fields are Microsurgery (Micro-manipulation), Skin harvesting (Surface tracking), Neurosurgery Interventional radiology (Constrained targeting).^[5]

II. DEVELOPMENT GOALS

Collaborative robotic arms are used in almost all spheres of industry because in addition to classic industrial precision, they also provide safety and protection, so that people can move freely in a collaborative environment.

Direction in which this project is developing, in its primary focus has demonstration that will with the help of new 3D scanning technologies, enable robots to perform functions with great precision. This represents the something that man cannot perform with his free hand. An example of such functions is robotic surgery.

III. 3D SCANNER HARDWARE DESCRIPTION

The system is primarily composed of Logitech C615 web camera (maximum resolution 1920 x 1080 pixels, and range from 30fps to 5fps), Projector - Acer K132 (resolution 1280 x 800 and 500 ANSI Lumens), Structured Light scanning software and calibration pattern and calibration panel. The angle between the walls of the calibration panel is 90 degrees and the size of the calibration pattern depends on the size of the object being scanned.

Additional hardware components required for construction of automatic 360° scanning system for rotating platform, Kinco 2S42Q-0240 Step motor, Kinco 2M412 stepper motor driver, DC Power supply, Arduino UNO, HDMI and USB cables. Algorithm code in Arduino software allows communication

between Structured Light scanning software and rotating platform. The optimal angle of the camera in relation to the projector is 22-25°. For objects larger than 350mm the camera is placed to the right of the projector.

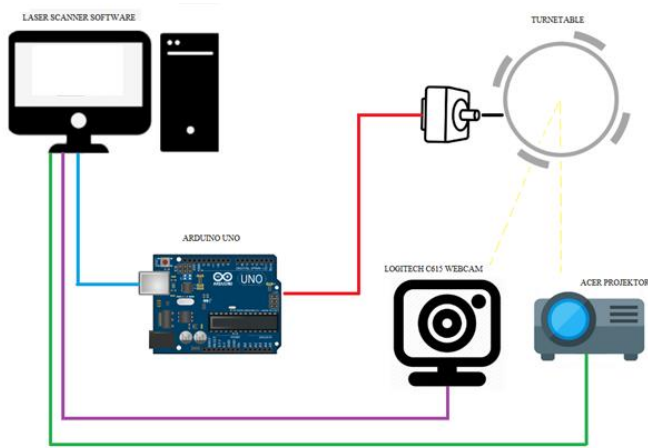


Fig. 1. Hardware configuration of the automatic 3D scanner

IV. DESCRIPTION OF THE APPLICATION INTERFACE

The calibration process consists of two key parts, camera calibration and projector calibration. In order to obtain a projection matrix from the world coordinates to the image in the process of calibrating the camera, the calibration pattern must have at least six points that are clearly visible.^[6]

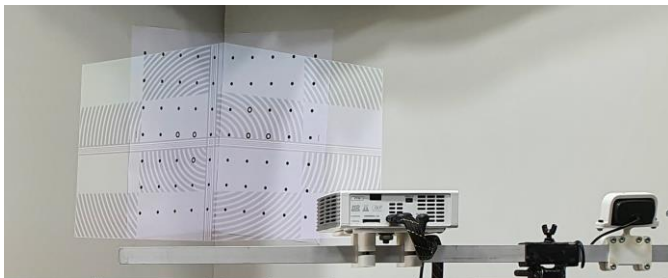


Fig. 2. Calibration process

The projector illuminates a striped pattern, so that software can determine the relative position of the camera and projector. The calibration template has many points, so that in the calibration process the projector can estimate all the unknown coefficients, unambiguously.

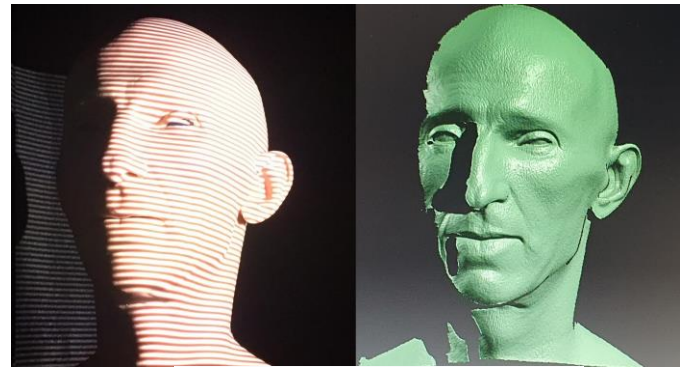


Fig. 3. Scan process (left) and result of one scan (right)

The software has the ability to communicate via the com port by sending numbers and letters. In advanced settings, it is possible to personalize the message that is sent and received.

The commands that the software receives are (Commands):

- Q – AddToList
- I – StartScan
- U – ModeSL
- P – StopLaserScan

The characters that the software sends are (Messages):

- 0 – StartSLScan
- H – GrabImage
- C – ModeSL

An algorithm implemented in Arduino that allows synchronizing movement of the rotating platform and the scanning software:

Serial.available() function returns number of bytes(characters) available for reading from serial port. Input is placed in variable *c* and the program waits for *c* to have a value of 0 which represents that the scanning has started. After that, the next character that arrives from the serial port is entered in the variable *p*, as long as *p* does not get the value of H, which means that the frame is captured, it is reloaded. When *p* is equal to H, the algorithm waits for *msAfterPictureTaken*. The value of this time content depends on the complexity of the scan, the image quality, and whether it is necessary to collect the texture color. A Q character is then sent to the serial port, which is a command to the software to save the current scan in a List. Program then waits for *msAfterSave*, and the platform on which the object is located rotates by Rotation angle. The Rotation angle value is determined before the scan begins and depends on the complexity of the object. In order for the platform to stabilize after the change of position, it is necessary to wait for *msAfterRotation*, the *cnt* counter is incremented by one and the U command is sent to the software to prepare for further scanning, to adjust all camera and projector parameters. If the value of the *cnt* counter is less than the number *n* obtained when 360 is divided by the Rotation angle, full rotation is not completed and scanning is continued by sending an I character to the serial port indicating the command for the software to start scanning. If a full rotation is performed, the command P is sent the scanning process is stopped and thus the process of collecting individual scans is completed.

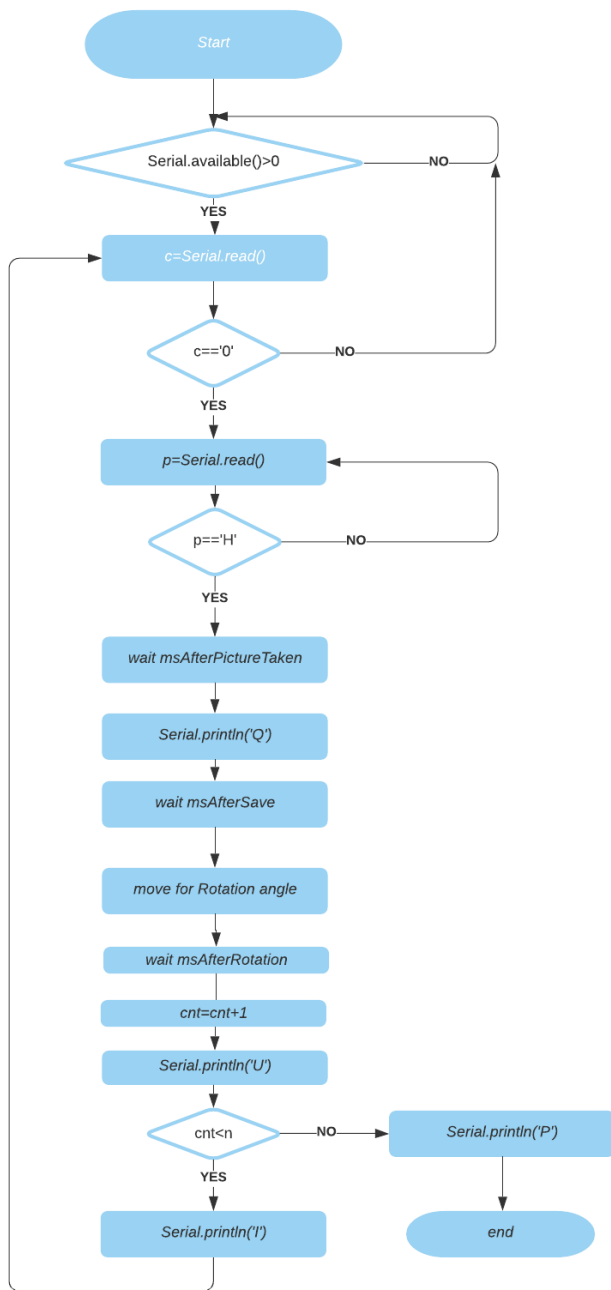


Fig. 4. Flowchart of the algorithm for synchronizing movement of the rotating platform and the scanning software

The duration of the scanning process depends on the quality of the scanned object required for proper application, so it needs to be determined experimentally. After collecting N scans, they are saved and exported to MeshLab to be aligned and then fused.



Fig.5. Screenshots of the successive scans

The result of the scan is the obj file format – it represents 3D geometry of the object. File contains:

- v-vertices 3xN matrix contains x, y, z coordinates of points.
- vt- texture vertices optional if texture capture is enabled during scanning.
- vn- vertex normal.
- f- faces (face is any of the individual flat surfaces of a solid object)

For files of this complexity to be used in the simulation in MATLAB, it is necessary to simplify the scan result and for that the MeshLab software package was used.

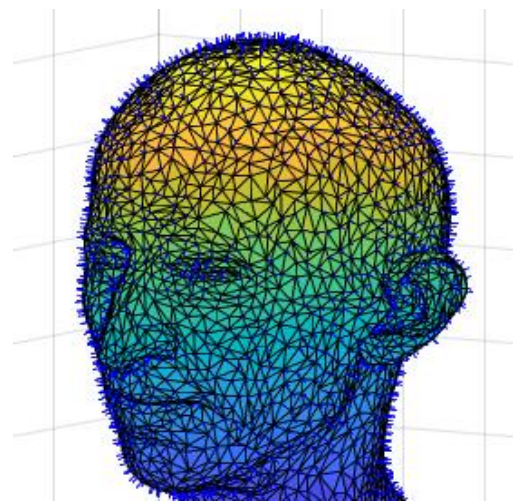


Fig.6. Mesh representation – of simplified scan in MATLAB

The position vectors consist of the x, y and z coordinates of each vertex. Let n_α be the normal vector derived from the mesh file, the components of the rotation matrix of the end effector are obtained as follows [8]:

$$\hat{a} = -\frac{n_\alpha}{\|n_\alpha\|}, \hat{n} = \hat{a} \times \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \hat{o} = \hat{a} \times \hat{n} \quad (1)$$

$$R_{EE} = [\hat{n} \ \hat{o} \ \hat{a}] = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (2)$$

From there we can derive values that represent the orientation of each vertex:

$$\begin{aligned} \psi &= \text{atan2}(r_{21}, r_{11}) \\ \theta &= \text{atan2}(-r_{31}, \cos(\psi)r_{11} + \sin(\psi)r_{21}) \\ \varphi &= \text{atan2}(\sin(\psi)r_{13} - \cos(\psi)r_{23}, -\sin(\psi)r_{12} + \cos(\psi)r_{22}) \end{aligned} \quad (3)$$

V. CONNECTION OF SCANNER AND ROBOT CONTROLLER

Universal robots have the ability to communicate over TCP/IP protocols (TCP socket connection over Ethernet). A PC is a server while a robot is a client. It is possible to establish communication by writing scripts in different programming languages, but as the simulation was performed in MATLAB the communication is also implemented in MATLAB. MATLAB has built-in functions for creating servers and clients. It is necessary to know the IP address and port number. Server waits for client connection. The robot needs to send the message that the server expects in order to get the first desired waypoint as feedback.

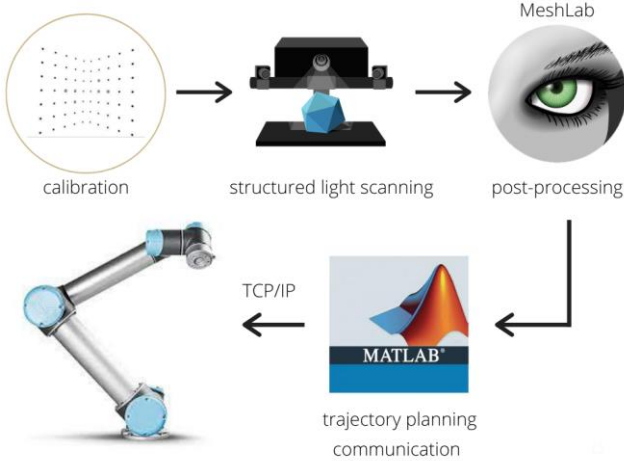


Fig.7. High-level description of the integrated system - 3D scanner and industrial robot UR-5

The message the robot receives consists of the number of data sent and that data, so that at any time it is possible to check whether the message has been sent in its entirety (format in PolyScope: `receive_data:= $[6,0,0,0,0,0,0]$`).

Data sent from server must be in bracket, each data must be followed by a comma unless it is the last, at the end there is 'n'. The robot has three main ways of calculating how to move from Waypoint to Waypoint which is a nonlinear movement "MoveJ", a linear movement, "MoveL" and a

circular movement (MoveC) which is under a Process move "MoveP".^[9]

The Universal Robot is controlled on Script Level, URScript is the robot programming language (PolyScope software).

An example of the functions used^[10]:

- *MoveL(*pose*, a=acceleration, v=*speed*, r=blend radius)* move to position (linear in tool-space), a[m/s²], v[m/s], r[m].
- *get_forward_kin()* returns tool pose-forward kinematic transformation (joint space to tool space) of current joint positions.
- *get_inverse_kin(x)* returns joint position- Inverse kinematic transformation (tool space to joint space). Solution closest to current joint positions is returned.
- *socket_open(server, port)* Open ethernet communication.
- *socket_read_asciifloat(number)* Reads a number of ascii float from the TCP/IP connected.
- *socket_send_string(str)* Sends a string to the server Sends the string through the socket in ASCII coding

Robot sends current tool positions (fun. *socket_send_string*) in the format: p[x,y,z,ψ,θ,φ]. Functions for reading current positions of the final device are written in MATLAB.

VI. DEMONSTRATION EXAMPLE

Experimental verification of the system was performed by scanning the model of Nikola Tesla's head.

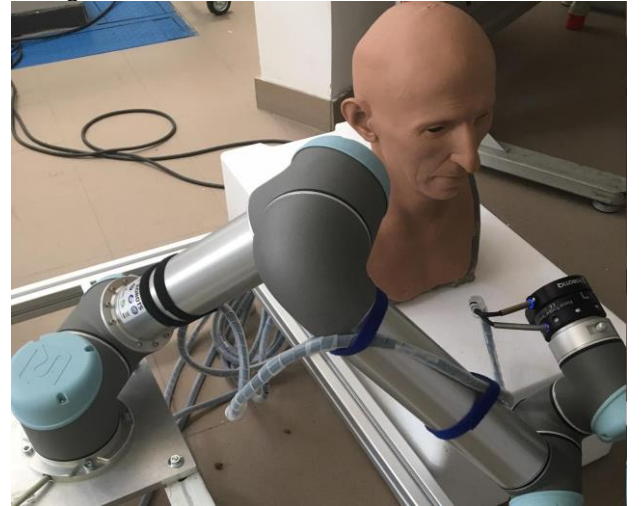


Fig8. The interaction of the robotic arm and the model of Nikola Tesla's head

In order to better simulate the movement of the robotic arm, the points obtained as a cross section of the point clouds and horizontal plane passing through the axis of symmetry were chosen as a demonstration example.

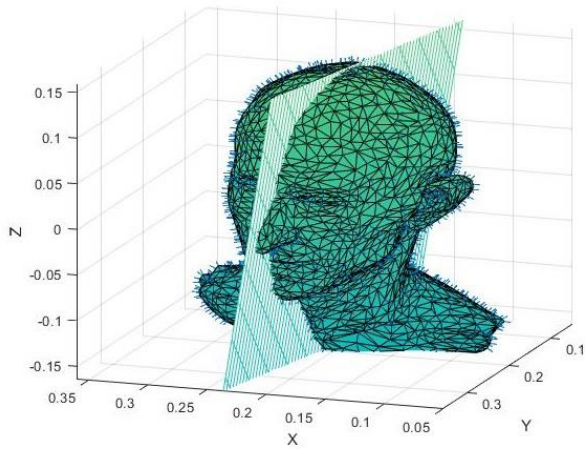


Fig.9. Intersection of planes and mesh representations

The distance of each point from the plane was calculated and based on that, points with a smaller distance from the threshold were selected. A value of 5mm was chosen for the threshold and *Figure10.* shows which points meet this criterion.

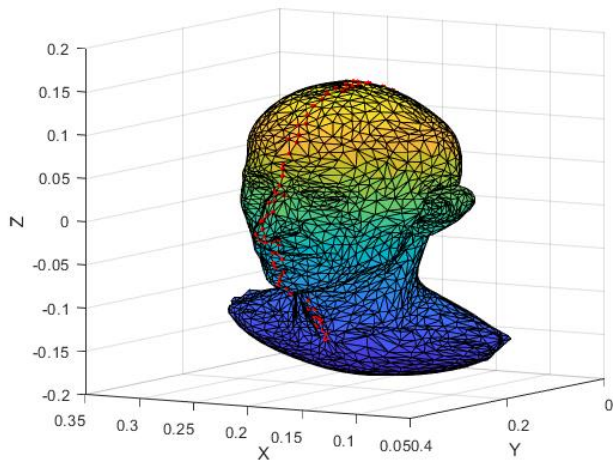


Fig10. Selected points

The robot from the initial position crosses the contour of the face at a constant speed of one cm per second, continuously without stopping.

A laser is placed on the end effector for easier visualization. In order to find a point on the line(X) containing the normal vector(*n*) that passes through the original point(*P*) on the scanned head model and is *d=1* cm away from the original point, the following system of equations is solved:

$$\vec{n} = \begin{bmatrix} l \\ m \\ s \end{bmatrix} \quad \begin{matrix} P = (x_1, y_1, z_1) \\ X = (x, y, z) \end{matrix} \quad (4)$$

$$t = \frac{x - x_1}{l} = \frac{y - y_1}{m} = \frac{z - z_1}{s} \quad (5)$$

$$d = \sqrt{(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2} \quad (6)$$

The result of the displacement of the profile points by 1 cm in the direction of the normal is shown in the *Figure11.*

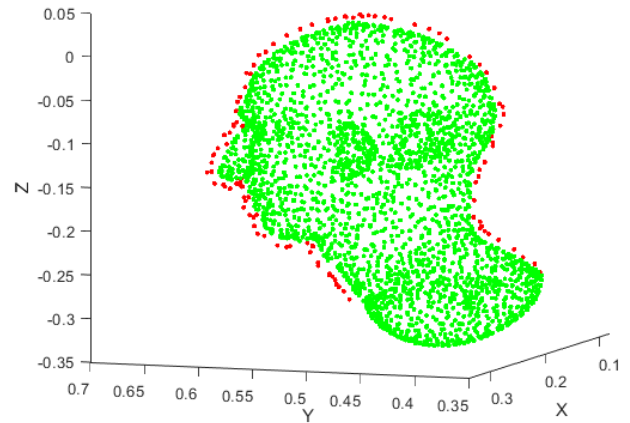


Fig.11. New selected points

For the purposes of simulation, the UR5 robot was imported from the Robotics System Toolbox, as rigid Body Tree.

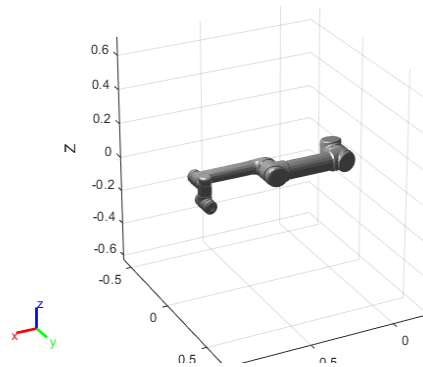


Fig.12. Robot UR5 in home configuration position

By calling the *inverseKinematics* function, an inverse kinematics object is formed which, based on the desired position and orientation of an end-effector computes the joint configuration that leads to it.^[11] The numerical optimization used in the calculation of inverse kinematics is the Broyden – Fletcher – Goldfarb – Shanno algorithm (BFGS):

Taking into account the starting point(*q₀*) and the initial Hessian matrix(*H₀*) for each iteration(*k*), the search direction(*d_k*) is calculated:

$$d_k = -H_k g_k \quad (7)$$

If the gradient(*g_k*) is zero, the search stops. If not, step size (*α_k*) is calculated with line search method, and new point is obtained:

$$x_{k+1} = x_k + \alpha_k d_k \quad (8)$$

Then the new Hessian matrix is recalculated, and the stop criterion is examined. If it is not fulfilled, it moves to the next iteration.^[12]

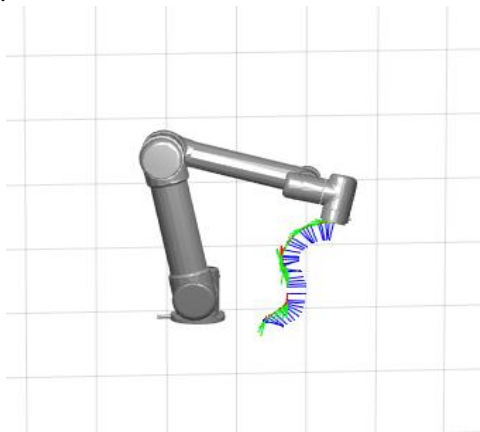


Fig.13. Simulation of trajectory monitoring

Motion trajectory was implemented as Joint space and Task space trajectory. In addition to interpolating the positions between each waypoint, it is necessary to implement interpolation between orientations. This is achieved by interpolation between quaternions owing to the fact that if we interpolated the Euler angles the solution would not be unique. The way this is implemented in MATLAB is the SLERP (Spherical Linear Interpolation) method.

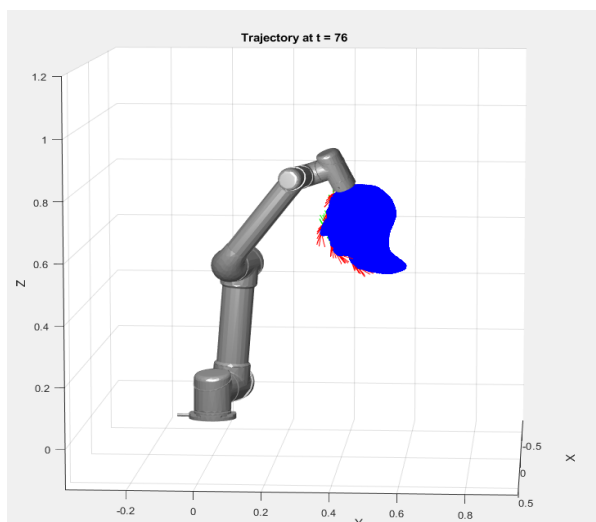


Fig.14. Simulation of trajectory monitoring

Another way to obtain the internal angles of the robot from the external coordinates of position and orientation is to implement an analytical procedure for solving inverted kinematics.^[13] As there are several configurations that can produce a certain position and orientation, one of the solutions is to choose the configuration that makes the smallest change in the position of the joints compared to the previous point. This method is much slower, so numerical calculation is applied with the help of libraries and built-in functions in the MATLAB Robotics Toolbox.

VII. CONCLUSION

Automatic 3D object scanning facilitates processes not only in robotics but also in other important spheres of life. Collaborative robots and their integration with such systems provide a new form of technological advancement that aims to improve the way people live and work. The results of the scanning of Nikola Tesla's head met the criteria of precision. The goal of this work, the construction of automated scanning and connection with the universal robot UR5 is achieved. Automated scanning is implemented using simple hardware for the motorized movement, and low cost or open-source software. Moreover, the process of filtering and collecting point clouds that form the robot trajectory is described. Experimental verification of the system was performed and communication is established.

ACKNOWLEDGMENT

The results presented in the paper are obtained in the scope of the research projects: „Development and Experimental Performance Verification of Mobile Dual-Arms Robot for Collaborative Work with Humans“, Science and Development Programme – Joint Funding of R&D Projects of the Republic of Serbia and the People's Republic of China, contract no. 401-00-00589/2018-09, 2018-2021 and national R&D project no. TR-35003, both supported by the Ministry of education, science and technology development of Republic Serbia.

REFERENCES

- [1] O. Vozisova, S. Eroshenko, E.Koksharova, A. Khalyasmaa and S. Dmitrir (2016, March), “Application of 3D Scanning and Printing Technologies in Electric Power Industry,” Ekaterinburg, Russia. In 2016 IEEE International Conference on Industrial Technology (ICIT) (pp. 892-897). IEEE
- [2] S. Koceski, N. Koceska, P. B. Zobel and F. Durante, "Characterization and modeling of a 3D scanner for mobile robot navigation," 2009 17th Mediterranean Conference on Control and Automation, 2009
- [3] E. Digor, A. Birk, A. Nüchter “Exploration Strategies for a Robot with a Continuously Rotating 3D Scanner”, Simulation, Modeling, and Programming for Autonomous Robots, 2010, Volume 6472
- [4] Ogorodnikova, O., Olchanski, D. “On a 3D scanning robot system design problem”, Periodica Polytechnica Mechanical Engineering, 51(1), pp. 39–44, <https://doi.org/10.3311/pp.me.2007-1.06>
- [5] Etienne Dombre „Introduction to Surgical Robotics“, LIRMM, Montpellier 2009.
- [6] Josep Forest i Collado, “New methods for triangulation-based shape acquisition using laser scanners,” *Girona, maig 2004*.
- [7] Du Q. Huynh, “Calibration of a Structured Light System: A Projective Approach,” Centre for Sensor Signal and Information Processing Signal Processing Research Institute Technology Park, The Levels, SA 5095.
- [8] Jelena Z. Vidaković, Vladimir M. Kvrđić, Mihailo P. Lazarević, Zoran Z. Dimić, Stefan M. Mitrović (2017), “Procedure for Definition of End-effector Orientation in Planar Surfaces Robot Applications,” *Belgrade Tehnika*, 72(6), 845-851.
- [9] UNIVERSAL ROBOTS User Manual, Version 3.3.3.
- [10] The URScript Programming Language Manual.
- [11] Peter Corke, Robotics Toolbox for MATLAB, Release 9.10, February 2015. <https://petercorke.com/toolboxes/robotics-toolbox/>
- [12] Atikah Ramli, Ibrahim Jusoh, Mohd Rivaie Mohd Ali (2017), “A combination of Broyden-Fletcher-Goldfarb-Shanno (BFGS) and n-th section method for solving small-scale unconstrained optimization”, *Malaysian Journal of Fundamental and Applied Sciences*, 13(4), 717-720.
- [13] Ryan Keating, “UR5 Inverse Kinematics”, Johns Hopkins University.

Fusion of Camera-Acquired Data and CAD 3D Models of Objects in Forming a Visual Feedback Loop for Industrial Robots

Miloš Nenadović, Uroš Ilić, and Aleksandar Rodić

Abstract—In this paper, we present a computer vision system for object detection and spatial position and orientation recognition. To solve the problem, we separated the process in several stages: in the first stage the system uses principal component analysis (PCA) method in optimal conditions to detect objects in the image. Then, after the object is extracted, we projected it in the appropriate eigenspace, produced by singular value decomposition (SVD) of the set of images of the rotated 3D CAD model. The closest match is then processed by correlation between it and the real-object image in log-polar space. The result is combined with information from other cameras to derive the approximate position and object orientation using multiple-view geometry.

Key words—Object detection; Singular Value Decomposition; Log-Polar; Multiple-view geometry.

I. INTRODUCTION

The problem of object detection in images is one of the oldest in computer vision [1] and has over the years been solved by various methods, among which are neural networks and principal component analysis (PCA). Convenient in the sense that they only require a set of training data to extract key features from objects, they provide vastly accurate classification results. However, as the accuracy rises so does the required memory, which is also the case as the complexity of objects rises.

In industrial robotics we require the robot arm to handle objects of varying complexities, ranging from simple boxes to machine parts. In these cases, a camera is a good tool to use to scan the scene and retrieve information about objects. The problem arises when we need to do more than classify an object. Assuming we are only provided with several cameras, we aim to get the most information about the scene which would be the position and orientation of objects.

We will aim to decrease the amount of images necessary to train the neural network, or in the case of principal component analysis [2] to eliminate the need for them altogether and instead only use the 3D CAD models provided. The CAD models are used to obtain reference images of their respective objects in different positions [3]. From these images we

calculate the histogram of oriented gradients (HOG) and use them for singular value decomposition (SVD) based on the idea that these features are enough to determine, at least the approximate, orientation of an object. Following this, we will use correlation of gradient data in log-polar space which is rotation and scale invariant. Additionally, we will analyze the speed of the algorithm to determine whether it can be used for moving objects. We will test the system on a set of simple objects (cube, cuboid, cylinder and pyramid) of different colors (red, blue, green, brown, orange, black). In future work, we will attempt to use the system on more complex objects such as machine parts or 3D wooden puzzle pieces.

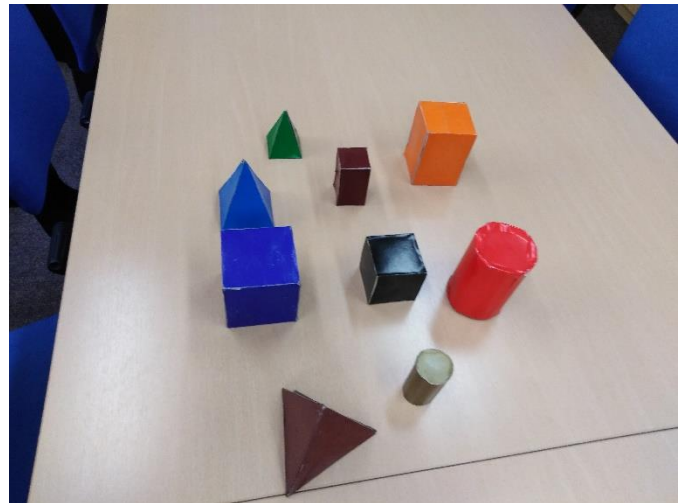


Fig. 1. Scene setup with simple objects of multiple colors on a table.

II. THE ROLE OF ROBOT VISION AND 3D CAD MODELING IN TRAJECTORY PLANNING

To understand the need for proper object position and orientation recognition, we look at an example of a robot grapppler arm. We assume the objects are still and placed on a table in various positions. To successfully grab an object, the robot arm must be placed so that the object does not slip from its grasp. Additionally, a robot may need to navigate between objects and in such cases the knowledge of their position and

Miloš Nenadović is with the Faculty of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia (e-mail: neosmilos@gmail.com).

Uroš Ilić is with the Faculty of Mechanical Engineering, University of Belgrade, 16 Kraljice Marije, 11120 Belgrade, Serbia (e-mail: ilicuros01@gmail.com).

Aleksandar Rodić is with the Institute Mihajlo Pupin, 7 Sazonova, 11000 Belgrade, Serbia (e-mail: aleksandar.rodic@pupin.rs).

orientation is crucial [1]. Camera images can be used to determine the approximate position of an object using multiple-view geometry, after knowing the exact or approximate positions of the cameras. In this case we assume camera positions are unknown except for their general relative position i.e. ‘front’, ‘left’ and ‘above’. Thus, we used reference points to determine the approximate camera positions before using them to determine the object positions.

III. CONFIGURING THE 3D ROBOT VISION SYSTEM

Although some results are possible without calculating the positions of the cameras, we strive to achieve the most accurate representation of the scene and so require this as the first step. Given that no prior information is known about the position and angle at which the cameras observe the scene, we use a set of reference points on the scene to determine them. The size, position and distance between the reference points is known. We search the images for these points, assign the appropriate labels for them, and then, using the known data we reconstruct the camera position and orientation.

IV. FUSION OF 3D CAD MODEL DATA AND CAMERA IMAGES

The CAD models are used to construct a set of images from various views by applying rotation transformations to them. For the SVD algorithm we use rotations around each of the axes in increments of 15 degrees, whereas for the log-polar correlation we use 5-degree increments, however for only two axes. This is because one of the axes contributes solely to 2D image rotation.

As for the vision data, as mentioned before, we use a set of three cameras to observe the scene. Each camera image is processed on its own in parallel with the others and upon completion updates the other two. In this way, should any image be wrongly paired, the image with the higher correlation coefficient will take precedence and change the initially estimated view-image to another one based on the camera positions.

In this way we achieve a feedback loop by utilizing CAD model data to interpret the information acquired by cameras and then comparing them to the real thing.

V. RECOGNITION OF OBJECT SHAPE AND SPATIAL ORIENTATION USING IMAGES AND 3D CAD MODELS

A. Object detection using PCA

Before the object orientation algorithm can begin, we must first detect objects in the image. One of the goals we set ourselves was to detect the objects without using real-object images, but rather just the models. For this purpose, we created 936 training images for each of the models, depicting different views that we used as the training set.

The PCA method for object detection relies on using a training set to find principal components (PCs) to use to reconstruct images. Assuming that the best image reconstruction can be done only when using the PCs created from an image set of the same type [2], based on the accuracy of the reconstruction, we can classify an image as one of the objects or part of the background.

As mentioned in [2] we use vectorized reference intensity images to acquire a projection matrix P , and μ as the mean of the set. When given a vectorized intensity input image u we get the reconstructed image r through:

$$r = P'P(u - \mu) + \mu. \quad (1)$$

The reconstructed image is then compared to the original image and their difference d is expressed as the 2-norm between the two image vectors, as in

$$d = \|r - u\|. \quad (2)$$

To test this method, we supplied an image of the blue pyramid object as the input image and then used different training sets containing reference model-images and looked at differences d they produced.

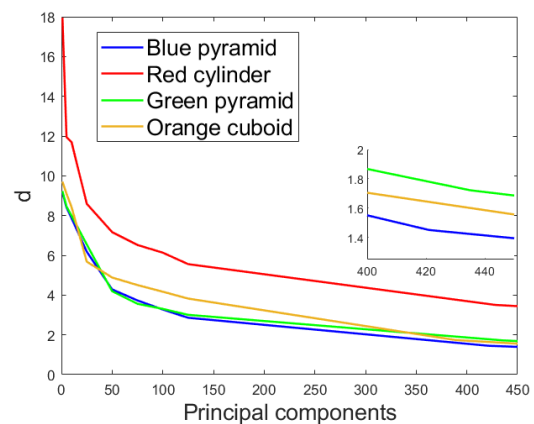


Fig. 2. Image reconstruction of the blue pyramid image using principal components obtained from four different model data sets (blue pyramid, red cylinder, green pyramid, and orange cuboid).

As shown in Fig. 2, the best reconstruction is achieved by using the appropriate training data set. We used a set of fixed numbers for the number of PCs used to reconstruct the image, except for the final one, which was the number of PCs that conserved 90% of the singular energy α , similar to how it was done in [3].

One of the problems this method encountered was the situation when, instead of an object, the input image was that of the background. Given its simplicity in comparison to an image containing an object the difference d was even lower than those of the desired object. Thus, we found that it was necessary to have a minimum number of real-object images and background images to create an additional classifier, or for the parts of the image with just the objects to be extracted before applying PCA.

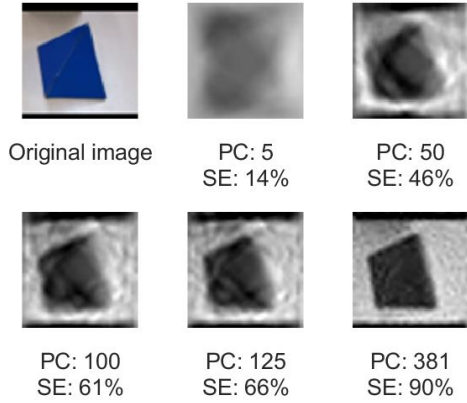


Fig. 3. Image reconstruction using principal components obtained from the blue pyramid model data set, with PC showing the number of principal components used and SE representing the retained percentage of singular energy.

As shown in Fig. 3 increasing the number of PCs also increases the accuracy of the reconstruction. However, doing so beyond a certain point would reduce efficiency instead as the increase of accuracy does not justify the decrease in computation speed and added memory.

B. Finding the closest match using the SVD algorithm

After extracting the object from the entire image, we proceed to use singular value decomposition to find the closest match among a set of rotated model images. As an object's orientation can be represented by its edges, we use gradient analysis to find features. The histogram of oriented gradients provides us with the necessary data t (column matrix of size $n \times 1$) that we place as the columns of a matrix $T = [t_1 \ t_2 \ \dots \ t_N]$, which the SVD algorithm is applied to.

$$T = U\Sigma V^T \quad (3)$$

From (3) we take the matrix U and use the first k ($1 \leq k \leq N$) columns representing eigenvectors of the largest k eigenvalues in S , sorted in descending order [2]. Like so, we derive the matrix U' from U . Multiplying with vectorized HOG features of input image data u we can project that data to the eigenspace, while reducing dimensionality. This is also applied to all columns of the matrix T creating a matrix T' of reduced size $k \times N$. We then compare the projected data of the input image u with the other projections t' from reference image HOG features (columns of T'), as in

$$d_i = \|t_i - U'^T u\|. \quad (4)$$

Index i in (4) denotes the index of the projection being compared with the projected input image. We store the results in an array which we sort at the end of the algorithm, and the index with the minimum difference points to the reference model image that would be the best pair to the input image.

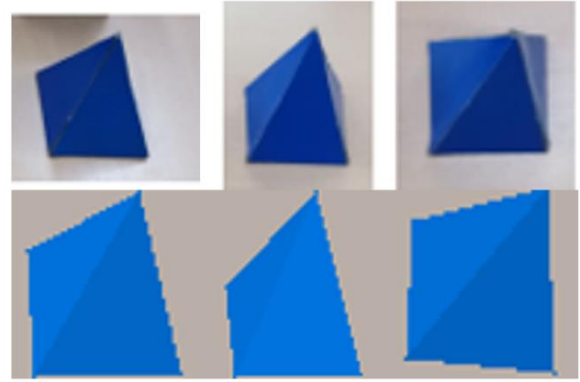


Fig. 4. Result of SVD comparison between extracted object images (top row) from three views (front, left and above) and a set of rotated model images (bottom row).

We can see in Fig. 4 that the algorithm provides two good matches and one bad match. Additionally, we can see that the sides of the pyramid in bad lighting were omitted from the approximations. However, as the algorithm's main purpose is just to narrow the search range for the subsequent parts, having even two out of three bad matches wouldn't pose a problem.

In future work, we will focus on improving the quality of the SVD method to include more accurate estimates both in optimal conditions, as was done in this paper, and when there is occlusion as the total time required for the match to be found greatly depends on the initial estimate.

C. Finding the closest match using log-polar correlation

Aside from the SVD algorithm used to narrow down the possible choices for object positions, we use intensity image correlation in log-polar space to deduce the best pairing of the real-object-image and view-image of the model. Log-polar space is used as it is both rotation and scale invariant [5]. Correlation is used to find the pair with the highest correlation coefficient and determine the rotation angle which is proportional to the vertical shift.



Fig. 5. Model images from multiple viewpoints (first row) and the log-polar transformations of the paired intensity images (second row)

One thing that can be seen from Fig. 5 is that all the log-polar transformations are quite unique. As such, it is not required to store all the information from those images. Rather, as they are computed from the center of the object and images encompasses mainly the object in question, we can take an area of interest that is the second half of the log-polar image along the ρ -axis. This area contains the entire object outline as well as the endpoints of the object edges within the outline.

Problems that may occur in this stage are related to the center used for computing the log-polar transformation. Because we were dealing with objects without occlusion and that are distinct from the background, we could use simple methods of color segmentation to find all the object pixels and then determine the center as a “center of mass”.

However, in cases where occlusion is present, or the object is more complex, or the object detection algorithm does not perfectly capture the object, finding the exact center becomes a problem. As mentioned before, log-polar correlation is both rotation and scale invariant, but it is sensitive when it comes to the center from where it is calculated. In case the estimated center of the object is shifted from where it is located on a template image then correlation may not yield adequate results, depending on how much the center is shifted. In [5] it is shown that the best correlation coefficient is achieved when the location of the object center matches that of its template image.

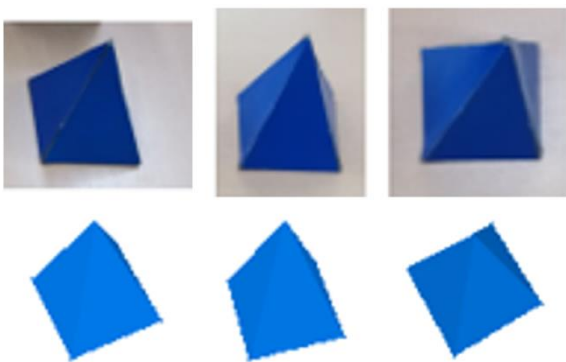


Fig. 6. Real-object images (first row) and best pairs from log-polar correlation without angle correction (second row).

The results shown in Fig. 6 show that the log-polar correlation method finds accurate matches for the objects in question. This means that it is safe to use the less-accurate SVD method first to provide an initial object orientation estimate.

VI. EXPERIMENTAL RESULTS AND DISCUSSION

The algorithm was run under the assumption of real-time work, meaning that the number of additional angles around the ones found by the SVD method, which would be checked by the log-polar correlation, had to be lowered to two sets of 4 angles on each side of the central one found by SVD (total of 9 angles). With such modifications the algorithm ran at around 60 ms for single-object orientation detection. This is excluding the time for object detection which took around 50 ms. We will attempt to further bring down the required computation time for this task in future work.

Additionally, one of the possible alterations that we tried was lowering the number of angles that would be checked to only one, the one provided by the SVD method. This lowered the computation time from 60 to around 10 ms.

The possibility of larger errors occurring due to SVD was taken into account and thus after every iteration (determining the 6D position of every object in the scene), the results are carried over to the next iteration until a plateau is hit. Based on

the experimental results, we have determined that the algorithm is capable of operating in real-time.

The entire algorithm we proposed is an iterative one and can be summarized in several steps:

1. We use PCA to detect objects in the image for each of the cameras. This step can be skipped after the first run in cases where the scene is static.
2. We calculate an input image’s HOG features and send them to the SVD algorithm to estimate the object’s orientation. This is done for every camera image, resulting in three estimates.
3. The estimates are then turned to intensity images, transformed into log-polar space, and processed by 2D correlation. The resulting coefficients are compared, and the best result is sent further.
4. Knowing the transformation matrices between cameras, we use the estimate to obtain images that the other cameras should see. These images are treated as the new input images and sent to Step 1.

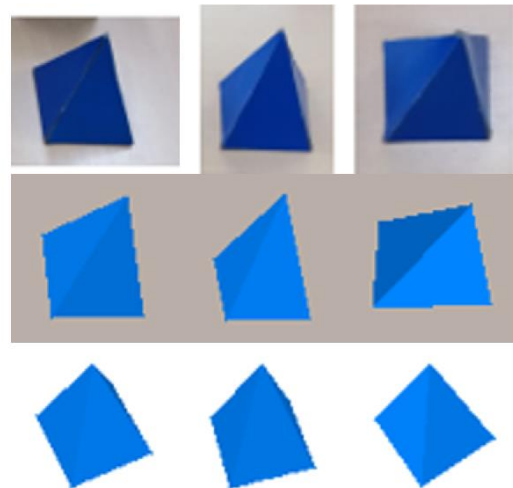


Fig. 7. Iteration 1 of the algorithm. Real-object images (first row), SVD estimated object view images (second row) and log-polar correlation results without angle correction (third row).

Fig. 7 shows that even in the first step of the algorithm we can obtain accurate results, similar to ones in Fig 6 and that the algorithm would end in the second iteration. The total number of iterations needed for convergence varies depending on the initial orientation estimations as well as the number of camera images that we can work with. In cases where one or two of the cameras cannot see an object, that number would exponentially rise.

VII. CONCLUSION

In this paper, we have presented a method of determining the 6D position of an object based on singular value decomposition and correlation of gradients in log-polar space. PCA allowed us to accurately detect an object, while using the same method with a set of HOG features allowed us to give an initial estimate of the object’s orientation. Afterwards, we have taken the

gradients of the grayscale images of the estimates and transformed them to log-polar space. The properties of the log-polar space, scale and rotation invariance, have allowed us to lower the number of matches (number of reference images) to compare, reducing computation time. Moreover, we have taken only part of the log-polar images, further reducing computation time. Finally, using iterations assures operation in real-time, while the results on Fig. 6 show that the algorithm will converge with accurate results.

ACKNOWLEDGMENT

The results presented in the paper are obtained in the scope of the research projects: "Development and Experimental Performance Verification of Mobile Dual-Arms Robot for Collaborative Work with Humans", Science and Development Program – Joint Funding of R&D Projects of the Republic of Serbia and the People's Republic of China, contract no. 401-00-00589/2018-09, 2018-2021 and national R&D project no.

TR-35003, both supported by the Ministry of education, science and technology development of the Republic of Serbia.

REFERENCES

- [1] Labbé, Y., Carpentier, J., Aubry, M., & Sivic, J. (2020, August). "CosyPose: Consistent multi-view multi-object 6D pose estimation". In *European Conference on Computer Vision* (pp. 574-591). Springer, Cham.
- [2] Luis Malagón-Borja, Olac Fuentes, "Object detection using image reconstruction with PCA", *Image and Vision Computing*, 27 (1-2), 2-9
- [3] Huang, S. C., Huang, W. L., Lu, Y. C., Tsai, M. H., Lin, I. C., Lau, Y. C., & Liu, H. H. (2019, January). "Efficient Recognition and 6D Pose Tracking of Markerless Objects with RGB-D and Motion Sensors on Mobile Devices". In *VISIGRAPP (1: GRAPP)* (pp. 375-382).
- [4] H. S. Prasantha, H. L. Shashidhara and K. N. Balasubramanya Murthy, "Image Compression Using SVD," *International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)*, 2007, pp. 143-145
- [5] R. Matungka, Y. F. Zheng and R. L. Ewing, "2D invariant object recognition using Log-Polar transform," *2008 7th World Congress on Intelligent Control and Automation*, 2008, pp. 223-228

Influence of muscle co-contraction indicators for different task conditions

Marija Radmilović, Djordje Urukalo, Miloš Petrović, Filip Bečanović and Kosta Jovanović

Abstract— In this research paper, arm co-contraction indicators are examined in different scenarios such as load variation, hand velocity variation, and in tasks with different precision. The experimental results show the relationship between muscle co-contraction and increase of load, velocity, or precision. According to the results, the differences in the muscle co-contraction related with gender and the age of the participants for the same task are evident. The results of the analysis for each task are in the align with the results presented in the previous research. The results of this research have made a significant contribution in analyzing human stiffness and its implementation in the human-like motion of robots.

Index Terms — muscle, co-contraction, musculoskeletal stiffness, biceps, triceps, anterior deltoid, posterior deltoid

Marija Radmilović is with the Mihailo Pupin Institute, University of Belgrade, 15 Volgina, 11060, Belgrade, Serbia, (e-mail: marija.radmilovic@pupin.rs).

Djordje Urukalo is with the Mihailo Pupin Institute, University of Belgrade, 15 Volgina, 11060, Belgrade, Serbia, (e-mail: djordje.urukalo@pupin.rs).

Miloš Petrović is with the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia (e-mail: petrovic.milos@etf.bg.ac.rs)

Filip Bečanović is a joint Ph.D. student with the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia, and the University of Paris-Est Créteil, Laboratory of Images, Signals and Intelligent Systems, 120 rue Paul Armandot, 94400 Vitry sur Seine, (email: becaphilippe@gmail.com)

Kosta Jovanović is with the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia (e-mail: kostaj@etf.bg.ac.rs)

Distribution of Control Tasks to Smart Devices in Industrial Control Systems: a Case Study

Zivana Jakovljevic, *Member, IEEE* and Dusan Nedeljkovic

Abstract—Cyber Physical Systems (CPS) and Internet of Things (IoT) open the way for new generation of Industrial Control Systems (ICS) characterized by high flexibility, modularity and reconfigurability necessary within Industry 4.0. Inevitable shift from centralized to distributed control systems is underway, but the changes are not as rapid as expected. One of the limiting factors is the lack of engineering techniques for distributed control systems design, simulation and verification. In this paper we analyze recently proposed techniques for distributed control systems development using an example of a simple transport system consisting of two CPS – smart conveyor belt and smart cylinder. In particular we consider the methods based on Control Interpreted Petri Nets (CIPN), Supervisory Control Theory (SCT) and IEC 61499 standard.

Index Terms—Distributed Control; Industrial Control Systems; Cyber Physical Systems; Industry 4.0.

I. INTRODUCTION

Industry 4.0 and introduction of Cyber Physical Systems (CPS) at manufacturing shop floor lead to significant changes in Industrial Control Systems (ICS) [1]. The demand for highly flexible automation and reconfigurable manufacturing induced by fluctuating market needs and high product variety on one [2], and the development of CPS based systems as the leading enabling technology on the other hand [1] represent the main drivers of these changes. CPS based smart devices with integrated computational and communication capabilities open up new possibilities in terms of ICS modularity, flexibility and reconfigurability. It is expected that with the full extent implementation of CPS at manufacturing shop floor the traditional automation hierarchy standardized through IEC 62264 will be replaced with truly distributed control systems where the control tasks will be carried out through interoperability of networked smart devices – Fig. 1. It is expected that all elements of automation hierarchy will remain, but in terms of functional hierarchy distributed over network without the pyramidal structure of the corresponding devices [3].

CPS are already readily employed at manufacturing shop floors in different automation tasks primary as smart sensors and actuators, and strict automation hierarchy is already

Zivana Jakovljevic is with the University of Belgrade – Faculty of Mechanical Engineering, 16 Kraljice Marije, 11000 Belgrade, Serbia (email: zjakovljevic@mas.bg.ac.rs).

Dusan Nedeljkovic is with the University of Belgrade – Faculty of Mechanical Engineering, 16 Kraljice Marije, 11000 Belgrade, Serbia (email: dnedeljkovic@mas.bg.ac.rs).

broken down as there exist communication of different devices intra and over non-adjacent levels of automation pyramid. Nevertheless, as a rule, smart sensor and actuators are integrated in ICS in traditional manner – they are connected to the central controller (e.g., Programmable Logic Controller - PLC) that carries out the control task. In this way, computational capabilities of CPS, their modularity and ability to make manufacturing systems adaptable to new products are not fully exploited.

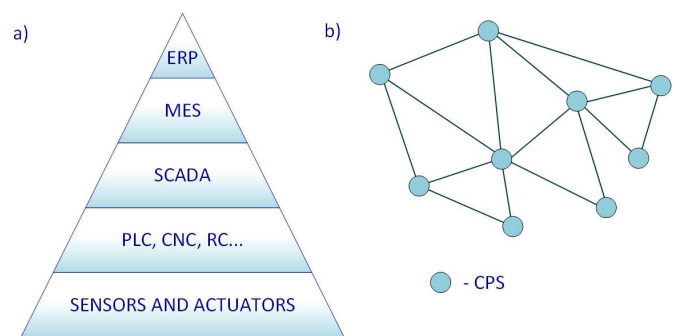


Fig. 1. Change of control paradigm in Industry 4.0: a) IEC 62264 automation hierarchy; b) Distributed control

There are several reasons for this. One is the inertia of control engineers community to implement new trends and their rationale to keep the existing techniques for ICS design, that were practically tested and proven in a myriad of real-world examples. Yet, due to inability of traditional ICS to deal with high product variety, there is a trend to perform a number of processes in manufacturing manually which represents a step backwards with respect to process sophistication; in this context, modularity of control system and distribution of control tasks are paramount for automation in Industry 4.0. The other reason is the lack of well-proven formally-consequent engineering techniques for the desing of distributed control systems, i.e., for the distribution of control tasks to smart devices [4]. Finally, when CPS are employed at manufacturing shop floor (regardless if centrally or distributed) cybersecurity related issues on communication links emerge.

Recently, a number of techniques for distribution of control tasks to smart devices and for dealing with cybersecurity within ICS have been proposed. Within this paper we will illustrate some of them using an example of simple system for parts transport that is presented in Section II. In particular, we will consider the method for control system distribution that is based on Control Interpreted Petri Nets (CIPN) that we proposed in [5], as well as the method based on Supervisory Control Theory [6] – Section III. Furthermore, in Section IV

we will show how the distributed control system can be simulated using IEC 61499 standard [7], and in Section V we will consider the possible effects of cyber-attacks on discrete event system using the methodology that we presented in [8]. Finally, Section VI gives some concluding remarks.

II. DESCRIPTION OF THE SYSTEM AND CONTROL TASK

The system that we consider in this paper is used for parts transport and consists of: 1) smart conveyer belt and 2) smart cylinder (Fig. 2). Conveyor belt transfers parts to position II and it is actuated by step motor M that is started/stopped using signal m . In addition, the belt contains sensor s that detects the presence of parts in removing position II, as well as start switch st that is used for the start of the system operation. Both sensors and actuator are connected to the belt's local controller denoted as LC_1 . Smart cylinder, on the other hand, besides the actuator, contains a monostable 5/2 dual control valve controlled by signal ap ($ap = 1$ for cylinder advancing and $ap = 0$ for cylinder retracting), as well as limit switches $a1/a0$ for detecting final advanced/retracted position. Pneumatic and sensing devices are augmented with local controller LC_2 that controls the smart cylinder. The allocation of sensing and actuation signals to smart devices' local controllers is presented in Table I.

TABLE I
SENSORS AND ACTUATORS SIGNALS MAPPING TO LOCAL CONTROLLERS

Device	Loc. Cont.	Signal	Description
Smart conveyer	LC_1	m	Motor actuation
		s	Sensor for part detection
		st	Start switch
Smart cylinder	LC_2	ap	Cylinder actuation
		$a1$	Advanced position sensor
		$a0$	Retracted position sensor

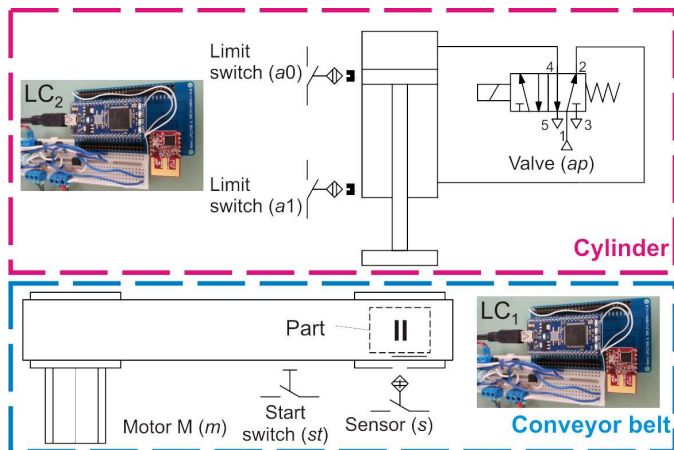


Fig. 2. Graphical representation of the system used in case study

The system should function as follows. After pressing the start switch, conveyor belt starts motion; when the part comes to the conveyor belt, it is transferred to position II where the sensor s is activated. After activation of signal s , conveyor belt stops and the cylinder removes the part from the belt (it advances and immediately retracts to home position). Once

cylinder reaches the retracted position, the conveyor starts moving again to transfer new part to position II and the cycle repeats.

There are a number of different methods for formal description of the controller that would ensure the described system behavior when it is controlled using one (centralized) controller, e.g., PLC. Most frequently employed technique that represents *de facto* standard in ICS design are CIPN and the derived formalisms of GRAFCET (Graphe Fonctionnel de Commande des Étapes et Transitions - Functional Graph of Control by Steps and Transitions) and IEC 61131-3 Sequential Function Chart (SFC). CIPN represent bipartite graphs containing transitions represented by bars and places represented by circles [9]. Within CIPN each transition has associated condition as a Boolean function of sensory signals (although actuator signals can be used as well), whereas the actions which change the values of control system outputs are allocated to the CIPN places. The state of CIPN is represented by token(s) assigned to places, and during system evolution the token passes from previous to the next place when the transition between them fires, i.e., when the associated condition is true. Using CIPN formalism the desired performance of the system from Fig. 2 can be described using CIPN form Fig. 3; this CIPN can be easily transferred to SFC or other embedded devices programming languages and implemented in centralized PLC or some other device (e.g., microcontroller).

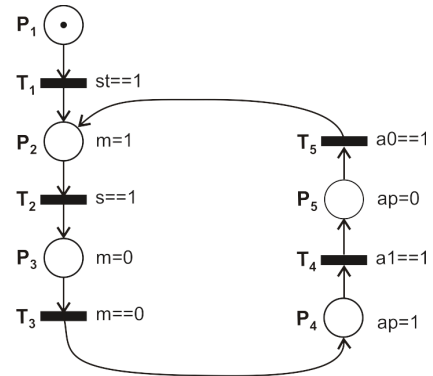


Fig. 3. CIPN describing the desired behavior of the system from Fig. 2.

III. DISTRIBUTION OF CONTROL TASKS TO SMART DEVICES

Opposite to centralized control systems design, generally, there is a lack of methods for the design of distributed controllers [10]. Existing approaches can be classified as *bottom-up* and *top-down*. Using *bottom-up* methods the behavior of each device within the control network is described using Petri nets [11], Supervisory Control Theory (SCT) [12], IEC 61499 [7] or similar formalisms, and the behavior of the system as a whole is obtained through their composition. These methods are characterized by low backwards compatibility since their implementation necessitates completely new mindset in system designers. Furthermore, they are usually based on trial and error and as such are time consuming and require significant verification [4]. *Top-down* approaches, on the other hand, describe the system as if centrally controlled and, using predefined methodology, distribute control tasks to smart devices. These

methods are characterized by high backwards compatibility and can be easily embraced by engineers in everyday practice.

A *top-down* approach from [5] is based on CIPN. Global CIPN describing the behavior of the system as a whole and mapping of sensors and actuators to local controllers with direct access to these devices are at the input of this method. Following the set of rules that consider the allocation of sensing and actuation signals to transitions and places within global CIPN on one hand, and physical mapping of sensors and actuators to local controllers LC_i on the other, this method generates a separate $CIPN_i$ to be employed at each LC_i . Basically, this approach extracts from global CIPN into $CIPN_i$ the places that contain actuating signals allocated to LC_i along with preceding transitions, and introduces places with sending commands to compensate missing links. The details regarding the method can be found in [5].

Following this approach, for the CIPN from Fig. 3 and allocation of sensing and actuation signals from Table I, $CIPN_1$ and $CIPN_2$ presented in Fig. 4 are obtained. Within these CIPNs, the transitions and places that were extracted from global CIPN (Fig. 3) are denoted in parentheses. The actions associated to places within $CIPN_1$ and $CIPN_2$ contain only actuating signals mapped to corresponding devices (conveyor belt and cylinder, respectively). On the other hand the transitions can contain the signals allocated to the other device, as denoted red in T_4^1 , T_{init}^2 and T_4^2 . These signals are transferred between local controllers using selected communication protocol, and protocol agnostic sending commands are denoted green in corresponding places in Fig. 4. These commands are the result of the application of the procedure from [5].

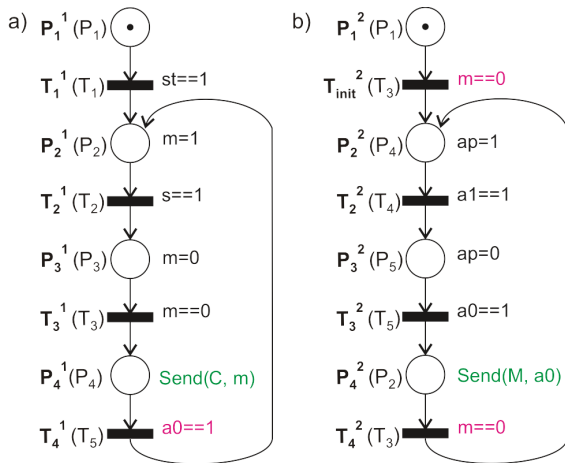


Fig. 4. CIPN representation of the system control distributed to smart devices: a) $CIPN_1$ representing control tasks distributed to smart conveyor belt; b) $CIPN_2$ representing control tasks distributed to smart cylinder;

For example, while at P_4^1 smart conveyor belt sends information about motor stopping to smart cylinder which receives it in transition T_{init}^2 or T_4^2 depending on the current state; in this way T_3 from Fig. 3 is performed. Similarly, while at P_4^2 smart cylinder sends information about getting to retracted position to smart conveyor belt, which the latter receives at transition T_4^1 , delivering the T_5 from Fig. 3. Once CIPNs for local controllers are generated, their implementation is straightforward as in the case of centralized controllers.

An example of *bottom-up* approach for distribution of control tasks will be illustrated using SCT formalism [6]. The models of the local controllers – supervisors for smart devices can be presented by Finite State Machines (FSM) in which the transition between states occurs on the events that represent the change of certain (actuation or sensory) signal. FSMs representing the desired behavior of LC_1 and LC_2 are given in Fig. 5. In particular, S_B represents smart conveyor belt (Fig. 5a) and S_C (Fig. 5b) smart cylinder cyber part. Note that in smart conveyor belt FSM model motor activation signal ($m = 1$) is denoted by mp , whereas deactivation signal ($m = 0$) is denoted mm . Similarly, for smart cylinder $ap = 1$ is denoted by ap and $ap = 0$ by am . The signals that are transferred from a local controller are marked green, and the signals that are received are marked red.

As can be observed from the Fig. 5, the design of local controllers using this formalism is not straightforward and requires to simultaneously take care about the behavior of both devices and communication of signals between them which can be extremely tedious and error prone when larger number of local controllers is used. The equivalence of S_B and S_C to $CIPN_1$ and $CIPN_2$ from Fig. 4 can be observed.

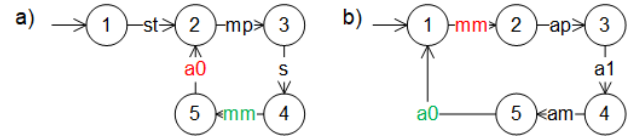


Fig. 5. Case study from Fig. 2 – FSMs representing: a) Conveyor belt local controller – S_B ; b) Cylinder local controller – S_C .

To verify that the system as a whole will have the desired behavior after implementation of the developed controllers, using SCT formalism, the physical part of the system should be modeled as well. FSMs from Fig. 6 - G_B and G_C represent physical parts of smart conveyor belt and smart cylinder, respectively.

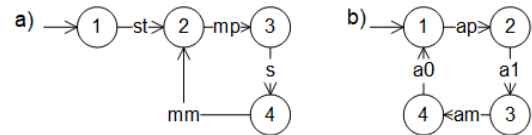


Fig. 6. Case study from Fig. 2 – FSMs representing: a) Conveyor belt physical device - G_B ; b) Cylinder physical device - G_C .

Conjoint operation of all four FSMs from Figures 5 and 6 is presented by FSM G from Fig. 7, where x, y, z and v in state notation (x, y, z, v) represent the states of S_C, S_B, G_C and G_B , respectively. FSM G , that represents the behavior of the CPS from Fig. 2, is obtained using the following FSM operation:

$$G = (S_B \parallel S_C) \times (G_B \parallel G_C) \quad (1)$$

where \parallel denotes FSM parallel composition, and \times product operator [6]¹. Comparing CIPN from Fig. 3 which represents the desired behavior of the system and FSM from Fig. 6 that represents the conjoint behavior of smart devices with distributed control tasks, the equivalence can be observed. Nevertheless, it should be noted that the implementation of SCT formalism assumes that CIPNs are not used during

¹ All automata operations are carried out using DESUMA software [13]

system representation, and in this paper we use it for the comparison only.

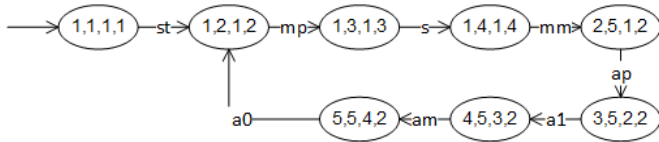


Fig. 7. Case study from Fig. 2 – behavior of the system represented using single FSM – G

IV. MODELING AND SIMULATION OF DISTRIBUTED CONTROL SYSTEMS USING IEC 61499

Once the control task is distributed to smart devices' local controllers, it is beneficial to further verify it, preferably using simulation. IEC 61499: Function blocks [7] represents an international standard intended for distributed control system modeling and simulation. Using this standard the functionality of a system's hardware or software component is encapsulated using function blocks (FBs) that introduce object oriented paradigm into industrial control systems programming [14, 15]. FBs represent classes whose instances can be used for task execution in concrete applications. Using this formalism, CPS are modeled and simulated through interaction between their physical and cyber parts, each represented by separate FB. The blocks are integrated through real-time interaction that is modeled using events and data flows. The functionality of an FB is event driven and it is defined by its Execution Control Chart (ECC) that specifies the behavior of the modeled component when certain events in the system occur. They are in the form similar to CIPN.

IEC 61499 formalism models the behavior of the system through a network of FBs called application. Within application each smart device is introduced using corresponding FBs and multiple devices are interconnected using certain events and data. Fig. 8 represents IEC 61499 application for the system from Fig. 2². Within this application function blocks ConveyCyber and ConveyPhys model cyber and physical parts of the smart conveyor, whereas CylCyber and CylPhys model corresponding parts of smart cylinder. Each FB contains head (upper part) that contains input (left side) and output (right side) events and body that contains input (left side) and output (right side) data. FB refreshes input/output data on the occurrence of the corresponding event, and their interconnections are modeled with connectors - red for event and blue for data flow.

Fig. 9 presents the ECCs for FBs modeling the behavior of the smart conveyor. At the system start, ConveyPhys transfers to S1, activates start switch (action *Start*) and issues **CNF** event to invoke the change of corresponding ConveyPhys output and ConveyCyber input; after that, it waits for **REQ** event and receipt of signal for motor activation (*mot = true*) from ConveyCyber when it passes to S2. Within S2 it activates sensor (action *Sensor_act*), issues **CNF** event to change corresponding input in ConveyCyber and waits for **REQ** events with the desired input data from ConveyCyber to return to S1.

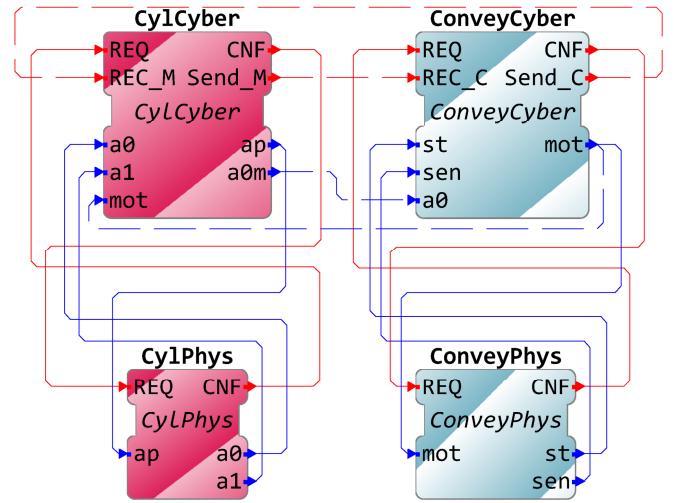


Fig. 8. IEC 61499 application that models the behavior of the system from Fig. 2.

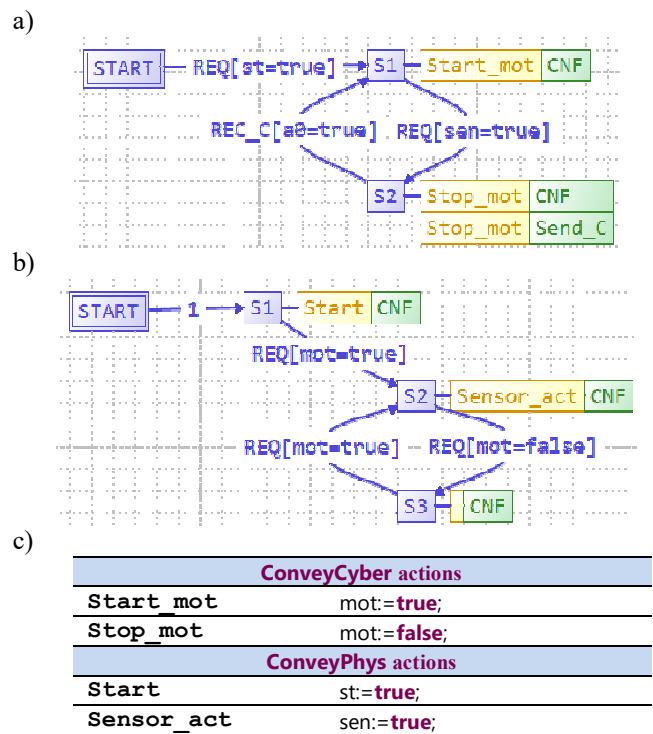


Fig. 9. ECCs for conveyor belt: a) cyber part of the conveyor belt - ConveyCyber ; b) physical part of the conveyor belt ConveyPhys; c) actions definition.

Simultaneously, at the beginning, ConveyCyber waits for **REQ** event (**CNF** event from ConveyPhys – Fig. 8) and the value *st = true* from ConveyPhys; on the receipt of this data it transfers to S1 where it carries out *Start_mot* action and invokes **CNF** event to inform the ConveyPhys (which is at that moment in S1 or S3) about signal change. Afterwards it waits in S1 information from ConveyPhys that sensor is activated to transfer to S2. In S2 it invokes the action for motor stopping and informs ConveyPhys and CylCyber about this change using events **CNF** and **Send_C**. Finally, after receipt of information that cylinder removed the part from the

² For IEC 61499 modeling and simulation 4diac software [16] is used.

belt from CylCyber through event **REC_C** and data $a0 = true$, it returns to S1.

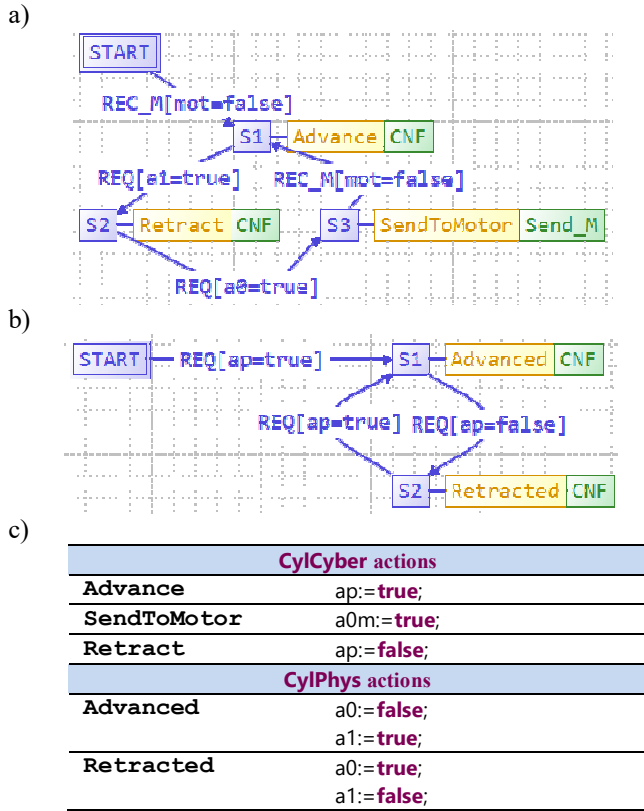


Fig. 10. ECCs for cylinder: a) cyber part of the cylinder - CylCyber ; b) physical part of the cylinder CylPhys; c) actions definition.

Similarly (Fig. 10), at the beginning CylCyber waits for the information from ConveyCyber that motor stopped (**REC_M** event that is connected to **Send_C** from ConveyCyber – Fig. 8), moves to S1, invokes cylinder advancing (action *Advance*), and upon receipt of the event and data referring to the completion of CylPhys advancing, issues commands for its retraction. When it gets the information that CylPhys retracted, CylCyber issues **Send_M** event to inform CylCyber that motor can start motion, and waits for the new signal from ConveyCyber that motor stopped (**REC_M** event connected to **Send_C**) to return to S1 and prepare for the new cycle. Finally, CylPhys moves between states corresponding to advanced and retracted positions in accordance with the events and data received from CylCyber.

Comparing these ECCs with CIPNs and FSMs from Figures 4, 5 and 6, the equivalence can be observed.

To simulate the behavior of the control system defined by application (Fig. 8) function blocks are allocated to different resources and the communication is introduced through publish and subscribe FBs as presented in Fig. 11 using an example of smart cylinder.

V. SECURITY RELATED ISSUES

As can be observed from previous sections, the performance of distributed control system is communication intensive, thus openings up the space for malicious cyber-attacks by various adversaries and raising cyber security related issues. A number of different attack scenarios can be identified and in case of discrete event systems such as one at hand two kinds of attacks can be singled out: 1) event removal and 2) event insertion. The adversaries can issue these attacks in various combinations always having the common goal to remain stealthy and to have negative effect on the system performance. Considering the possible consequences of cyber-attacks on ICS that can be not only of economic nature but also safety related (catastrophic damages of the equipment or even threats to human health and lives), it is crucial to analyze possible attack points, scenarios and consequences at early phases of the system design. Within this paper we will briefly illustrate an approach for modeling attacks scenarios using SCT based methodology that we proposed in [8]. We will consider the examples of removal and insertion attacks on $a0$ signal transmitted from smart cylinder to conveyor belt.

Following $a0$ removal attack denoted by $a0r$, smart conveyor belt cyber part will remain in the state at which it waits for $a0$ to progress with functioning, whereas smart cylinder cyber and physical parts will continue functioning as if attack did not occur. Described evolution of the system can be modeled using FSMs under attack S_B^r , S_C^r and G_C^r presented in Fig. 12a-c. In these automata the evolution of the system elements under attack is denoted in red lines. The details of the model generation can be found in [8], and the performance of the system as a whole is represented by automaton:

$$G^r = (S_B^r \parallel S_C^r) \times (G_B \parallel G_C^r) \quad (2)$$

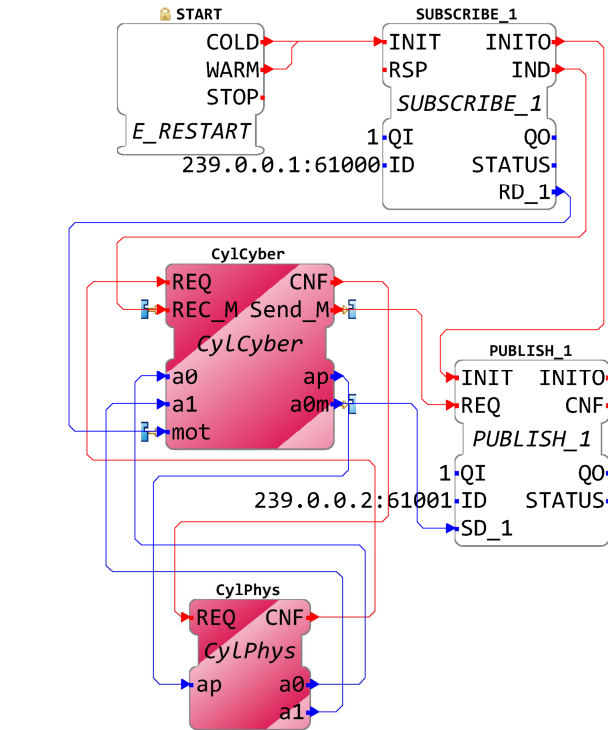


Fig. 11. Cylinder resource with introduced Publish and Subscribe function blocks for simulation of the system performance.

which is presented in Fig. 12d. From this figure it can be observed that on the $a0$ removal attack the system stops (it enters the state marked red) and no damage is expected.

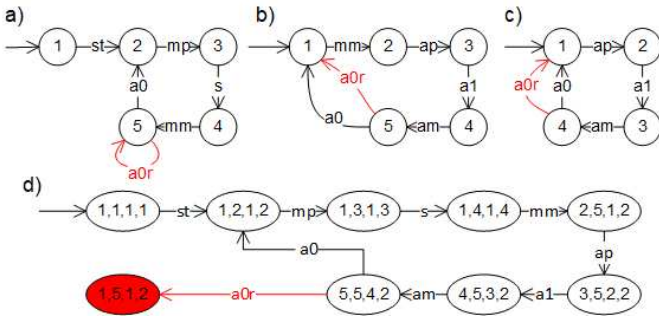


Fig. 12. Model of the system under $a0$ removal attack: a) S_B^r – model of conveyor cyber part; b) S_C^r – model of cylinder cyber part; c) G_C^r – model of cylinder physical part; d) model of the whole system behavior under attack.

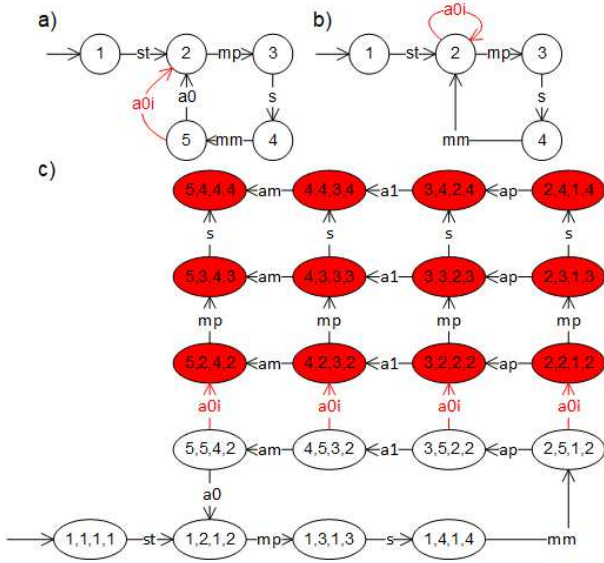


Fig. 13. Model of the system under $a0$ insertion attack: a) S_B^i – model of conveyor cyber part; b) G_B^i – model of conveyor physical part; c) model of the whole system behavior under attack.

When insertion attack is considered, it should be noted that to remain stealthy, the adversary can issue the $a0$ insertion attack only while S_B is in state 5 [8]. Following insertion attack $a0i$, the conveyor belt supervisor will transfer from state 5 to state 2 as if real $a0$ was received as presented in red line in FSM S_B^i in Fig. 13a. Simultaneously, conveyor physical part will remain in the state 2 as given in automaton G_B^i from Fig. 13b (the details regarding the formalism for generation of these automata can be found in [8]). The behavior of the whole system under $a0$ insertion attack is modeled through:

$$G^i = (S_B^i || S_C) \times (G_B^i || G_C) \quad (3)$$

presented in Fig. 13c where the possible evolutions of the system after attack are denoted using red states. These scenarios include starting the motor before the part is removed from the conveyor belt and can lead to the falling of the part from the transporter and its damage.

VI. CONCLUSION

Within this paper we have summarized and illustrated using a case study our recent research results in the area of CPS based distributed control systems design, verification and cyber protection. In particular, we have presented the application of the *top-down* approach for distributed control system design that is based on CIPN and that is characterized by excellent backwards compatibility. The convenience of this technique is supported through illustrating the application of a *bottom-up* technique based on SCT that requires completely new approach to ICS modeling. Furthermore, we have shown how the developed distributed control system can be simulated using standard IEC 61499. Finally, since the deployment of CPS and IoT at manufacturing shop floor leads to significant cyber security issues, we have shown how the SCT based framework can be applied for the analysis of cyber threats in our case study.

ACKNOWLEDGMENT

This research was supported by the Science Fund of the Republic of Serbia, grant No. 6523109, AI-MISSION 4.0

REFERENCES

- [1] H. Kagermann, W. Wahlster, and J. Helbig, "Recommendations for Implementing Strategic Initiative INDUSTRIE 4.0," Acatech, Germany, 2013 [Online]. Available: <http://www.acatech.de>
- [2] H. ElMaraghy, "Smart changeable manufacturing systems," *Proc Manuf*, vol. 28, pp. 3-9, 2019.
- [3] Reference Architecture Model Industrie 4.0 (RAMI4.0), 2015. [Online]. Available: <http://www.zvei.org>
- [4] J. Otto, B. Vogel-Heuser, and O. Niggemann, "Automatic parameter estimation for reusable software components of modular and reconfigurable cyber-physical production systems in the domain of discrete manufacturing," *IEEE Trans. Ind. Informat.*, vol. 14, no. 1, pp. 275-282, Jan. 2018.
- [5] Z. Jakovljevic, V. Lesi, S. Mitrovic, and M. Pajic, "Distributing sequential control for manufacturing automation systems," *IEEE Trans. Contr Syst Tech.*, vol. 28, no. 4, pp. 1586-1594, July 2020.
- [6] P. Ramadge and W. Murray Wonham, "The control of discrete event systems," *Proc. IEEE*, vol. 77, no. 1, pp. 81-98, 1989.
- [7] *Function blocks – Part 1: Architecture*, IEC 61499-1:2012.
- [8] Z. Jakovljevic, V. Lesi, and M. Pajic, "Attacks on Distributed Sequential Control in Manufacturing Automation," *IEEE Trans. Ind. Informat.*, vol. 17, no. 2, pp. 775-786, Feb. 2021.
- [9] R. David and H. Alla, *Discrete, Continuous, and Hybrid Petri Nets*, 2nd ed. Berlin, Germany: Springer, 2010.
- [10] R. Vrabčić, D. Kozjek, A. Malus, V. Zaletelj, and P. Butala, "Distributed control with rationally bounded agents in cyber-physical production systems," *CIRP Ann.*, vol. 67, no. 1, pp. 507-510, 2018.
- [11] L. Wang, C. Mahulea, and M. Silva, "Distributed model predictive control of timed continuous Petri nets," in *Proc. IEEE Conf. Decis. Control*, pp. 6317-6322, Dec. 2013.
- [12] M. Wakaiki, P. Tabuada, and J. P. Hespanha, "Supervisory control of discrete-event systems under attacks," *Dynamic Games and Appl.*, vol. 9, pp. 965-983, Dec. 2019.
- [13] L. Ricker, S. Lafortune, and S. Genc, "Desuma: A tool integrating gddes and umdes," in *Proceedings - Eighth International Workshop on Discrete Event Systems, WODES 2006*, pp. 392-393, 2006.
- [14] J. H. Christensen, T. Strasser, A. Valentini, V. Vyatkin, and A. Zoitl, "The IEC 61499 Function Block Standard: Software Tools and Runtime Platforms," in *ISA Automation Week*, 2012.
- [15] V. Vyatkin, *IEC 61499 Function Blocks for Embedded and Distributed Control Systems Design*, ISA, 2012.
- [16] <https://www.eclipse.org/4diac/> Accessed: 2020-04-24

Application of the Angular Dependency of the Zero Moment Point

Tilen Brecej¹ and Tadej Petrič²

Abstract—In this paper the widely used stability parameter called the zero moment point (ZMP) is redefined as the angle around the center of mass (COM) of the investigated system. With this redefinition the ZMP is expressed in a more general way which enables its application in a wider range of situations. The angular definition of the ZMP was validated with motion measurements of a person performing different movements recorded with the OptiTrack camera system. The skeleton of the filmed person was reconstructed with the Motive software and a body model was used to reconstruct its COM, which was further used to calculate the ZMP. The stability analysis of the recorded motion measurements presented in this article shows on real-world examples of human motion that the angular redefinition of the ZMP provides a general, reliable and simple-to-apply way of determining the stability of a system.

I. INTRODUCTION

Since the beginnings of motion control researchers have investigated stability conditions and defined different parameters that reveal whether a systems is stable or not and enable the calculation of possible motions in different situations. The research in the field of robot stability has deepened with the development of robotic systems capable of performing complex motion tasks such as for example humanoid robots performing human-like motion. Achieving stability of humanoid robots can be due to their relatively small feet in comparison to their relatively large body sizes very challenging. This is why also simple tasks such as for example walking, demand for constant stability verification and prediction in order to enable their accomplishment [1], [2], [3], [4]. But as the progress in this field is advancing with an incredible speed, robots became recently also capable of running, jumping and even skiing [5], [6], [7].

A system is capable of performing the desired tasks only if the forces, acting from the support polygon, allow it, which means that they make the system dynamically or statically stable. If the system is supported only by the forces acting from the ground, the support polygon lies within the boundaries of the contact of the system with the ground. In the case of a humanoid supported only by its feet the support polygon extends from its heel to its toes and between the outer edges of its left and its right foot.

One of the parameters defining the system's stability is the zero moment point (ZMP), defined as the location on the ground, where all the forces and torques, acting on the system, can be replaced by only one force [8], [9]. If the ZMP lies within the support polygon it coincides with the center of pressure and in this case the system is stable. On

the other hand, if the ZMP lies outside the support polygon it can not exist as there are no support mechanism outside the support polygon and therefore it is usually referred to as a fictitious ZMP. In this case the system is not stable and if the system is a humanoid it will flip either over its toes or its heel.

But different systems may have different support mechanism that do not necessarily act on the ground. A humanoid may for example support itself with his arms, that may apply support forces at different heights, or with its bottom, in the case when it is sited. In such scenarios the support polygon does not lie only on the ground but it extends to different heights, as shown in Fig. 1, and therefore the standard

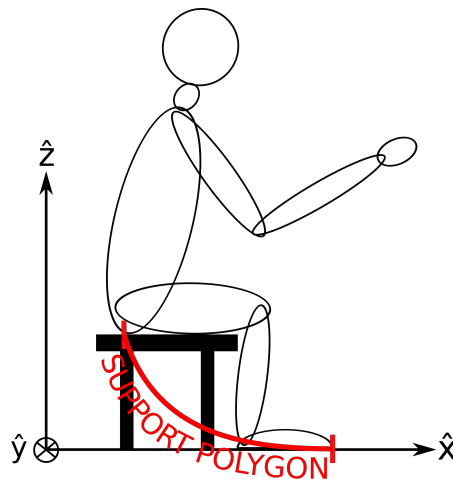


Fig. 1: The support polygon of a humanoid sited on a bench.

definition of the support polygon and the ZMP located on the ground can not be used. This is why in this article the ZMP is expressed in a more general way as the angle around the center of mass (COM) of the investigated system. Such definition can be applied to all systems, also those supported by different mechanisms at different heights, even below the ground or above the system itself.

In this article the stability of a humanoid, which can be either a person or a humanoid robot, is described, but the derived stability conditions can be applied to any system.

II. THE ZERO MOMENT POINT

A. The Standard Cartesian Definition

One of the most widely used stability parameters, the ZMP, can be expressed from the torque balance equation that takes into account the gravitational force acting on each segment of the humanoid, the forces accelerating each segment of the

¹Tilen Brecej and ²Tadej Petrič are with the Jožef Stefan Institute, Jamova c. 39, 1000 Ljubljana, Slovenia {tilen.brecej, tadej.petric}@ijs.si

humanoid in the desired direction, along with the torques produced by the rotations of each humanoid segment and all the external torques and forces present. By the definition the horizontal torques in the ZMP must be zero and only torques in the vertical direction can exist, but they are in usual circumstances balanced by the friction forces [8], [9].

The coordinate system defining the space in which the humanoid is positioned is oriented such that the sagittal plane of the humanoid lies in the $x-z$ plane, the lateral plane of the humanoid lies in the $y-z$ plane, while the ground lies in the $x-y$ plane, as shown in Fig. 1. This way the x and y components of the torque in the ZMP must be equal to zero, while its z component can be non-zero. For a thorough explanation of the torque balance equation and the derivation of the ZMP see [10].

The torque balance equation is a general equation that takes into account multiple effects that do in numerous circumstances not exist or can be neglected. This is why the position of the ZMP on the ground is usually obtained with some simplified models, one of them being the linear inverted pendulum model [11]. The latter assumes that the investigated system is symmetric with respect to the sagittal plane, it can be approximated by its COM, there are no external forces and torques present, the system does not rotate around any axis and that its COM lies at $y = 0$ and moves only in the x direction. In this case the location on the ground where the ZMP lies can be obtained as

$$x_{zmp} = x_{com} + \frac{z_{com}}{g} \ddot{x}_{com}, \quad (1)$$

where x_{com} and z_{com} are the x and z coordinates of the COM of the investigated system, respectively, \ddot{x}_{com} is its acceleration in the x direction, while g is the gravitational acceleration.

But the Cartesian definition of the ZMP may become useless in practice if the ZMP is located too far away from the investigated system, as the system may not have suitable mechanisms to support itself at such distant locations. Furthermore, if the system uses other support mechanisms that are not applying support forces on the ground, which may be in the case of a humanoid its arms and hands, the support polygon does not extend only on the ground and therefore its stability can not be verified using only the ground location of the ZMP. This is why a more general expression of the ZMP was developed.

B. The Angular Definition

From the torque balance equation it can be shown that taking into account the same assumptions as for the derivation of the linear inverted pendulum model, with the exceptions that now the ZMP does not need to lie on the ground any more and that the COM of the system can be accelerated also in the z direction, the ZMP can lie anywhere on the ZMP line defined as

$$z_{zmp} = \tan^{-1}(\varphi_{zmp})(x_{com} - x_{zmp}) + z_{com}, \quad (2)$$

where z_{zmp} and x_{zmp} are the z and x coordinates on the ZMP line, respectively, φ_{zmp} is the ZMP angle defined as

$$\varphi_{zmp} = -\arctan 2(\ddot{x}_{com}, (\ddot{z}_{com} - g)) \quad (3)$$

and \ddot{z}_{com} is the vertical acceleration of the COM. φ_{zmp} is located between the vertical line, passing through the COM of the investigated system and the ZMP line, which is in the case, when the assumptions made for the derivation of (2) hold true, passing through the COM of the investigated system, as shown in Fig. 2. With the angular redefinition

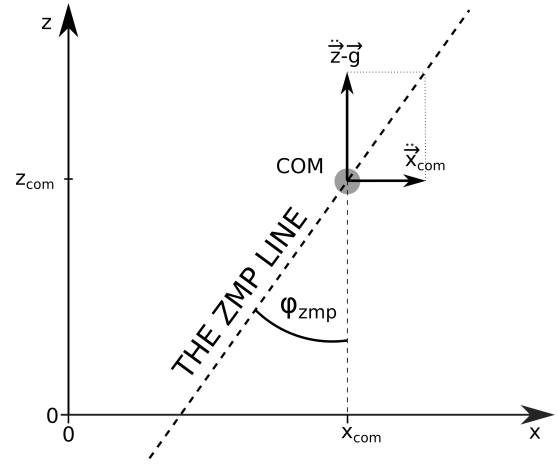


Fig. 2: The angular definition of the ZMP. φ_{zmp} is the angle between the ZMP line and the vertical line, passing through the COM of the investigated system.

the ZMP does not need to lie on the ground any more but it can lie at any height also below the ground, where $z_{zmp} < 0$, anywhere between the ground and the COM of the investigated system, where $0 \leq z_{zmp} \leq z_{com}$, or above the COM, where $z_{zmp} > z_{com}$. For a detailed derivation of the angular definition of the ZMP see [10]. To obtain the ZMP angle in the lateral ($y-z$) plane of the humanoid, the COM coordinates and accelerations in the x direction from (2) and (3) must be substituted with the corresponding values in the y direction.

The stability condition stating that the ZMP must lie within the support polygon can be expressed with the ZMP angle and the angles of the edges of the support polygon in the following way. For the sagittal plane of the humanoid, which is supported by only one foot, or by both feet, positioned in parallel one next to each other, this condition can be expressed as

$$\varphi_{heel} \leq \varphi_{zmp} \leq \varphi_{toes}, \quad (4)$$

where

$$\varphi_{heel, toes} = \arctan \left(\frac{x_{heel, toes} - x_{com}}{z_{com}} \right), \quad (5)$$

as shown in Fig. 3 If the humanoid is supported on the ground with both feet, that are not parallel one to the other, φ_{heel} refers to the heel of the back foot, while φ_{toes} refers to the toes of the front foot. If, on the other hand, the humanoid

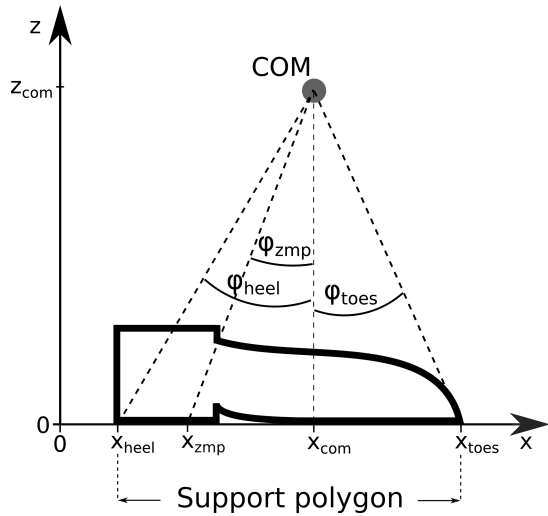


Fig. 3: The angles of the edges of the heel and the toes of the foot of the humanoid with respect to its COM, as defined by (5).

is oriented in the opposite direction, φ_{heel} and φ_{toes} must be interchanged. If the condition (4) does not hold true, the humanoid is not stable and it will flip over its toes or its heel.

III. VALIDATION OF THE ANGULAR DEFINITION OF THE ZMP

The angular definition of the ZMP was validated on two measurements of human motion filmed with a set of 13 OptiTrack cameras [12], emitting infrared light and detecting its reflection from 37 reflective markers, positioned on the filmed person. The markers were placed on predefined positions on the human body in order to enable Motive 2.2.0 [13] (the OptiTrack optical motion capture software) to reconstruct the human skeleton. Knowing the approximate positions of the human joints, estimated by the Motive software, and using a human body model [14], [15], [16], which enables the reconstruction of the sizes and masses of each body segment and the distances of their corresponding centers of mass (COMs) from the adjacent joints, the reconstruction of the total COM of the filmed person was possible. The ZMP line and the ZMP angle from (2) and (3), respectively, were then obtained from the reconstructed location and acceleration of the total COM of the person for each recorded frame.

In Fig. 4 you can see the frame sequence outtake from the first filmed motion. The top row shows the lateral plane (front view), while the bottom row shows the sagittal plane (side view) of the filmed person. The frames positioned one above the other were obtained at the same time and therefore represent the same capture. On the first two captures, obtained at times 2.7 s and 4.1 s, the filmed person was swinging from his right side to his left side, on the second two captures, obtained at times 20.6 s and 24.9 s, the person was bowing forth and back, while at the last capture, obtained at time

37.6 s, the person was standing on only one leg, bowing forth, with the other leg lifted up in the air and with his hands extended sideways.

The y and the x coordinates of the ZMP and the edges of the support polygon of the recorded person during the first motion measurement, presented in Fig. 4, where the person was supported only by his feet, are shown in Fig. 5 as a function of time. On the plot on top it can be seen that in the y direction the person was unstable only for small amounts of time during the measurement, when the ZMP is located outside the support polygon. During this measurement the person was not standing on both feet all the time but was also stepping to only one foot while swinging from one side to the other. This can be seen as a sudden narrowing of the support polygon in the y direction, such as for example at time 1.2 s, when the person started to stand only on his left foot, and a sudden widening of the support polygon, when the person was supported again by both feet, such as for example at time 2.0 s. The deviation of the ZMP outside the support polygon occurred during the swinging at around 1.9 s and 15.6 s, when the person was supported only by his left and only by his right foot, respectively, just before he landed to the other foot. This instability arised because just before the landing on the second foot, the person was falling down towards the ground and could not control his motion. Another instability can be seen at time around 14.1 s when the person was standing only on his right foot and was caused by fast and sudden movements of the person while catching balance. As the ZMP was calculated only from the reconstructed position and acceleration of the total COM of the body, the torques produced by the rotations of different body links were not taken into account, which can for fast and sudden movements influence the location of the ZMP. On the other hand, in the x direction the person was stable all the time, as the ZMP location was within the boundaries of the support polygon throughout the whole measurement, as shown in the bottom plot of Fig. 5.

Fig. 6 shows the same support polygon as Fig. 5, but with the ZMP and the edges of the support polygon expressed as angles around the COM of the measured person, defined by (3) and (5), respectively. The angular results are similar to the results expressed with Cartesian coordinates on the ground, but the lines representing the edges of the angular support polygon are more curved than the corresponding lines in the Cartesian coordinate system, as the angular results are not obtained relative to a fixed coordinate system but relative to the COM of the moving person. The angular values of the ZMP are, on the other hand, subjected to smaller variations in time than the corresponding Cartesian values.

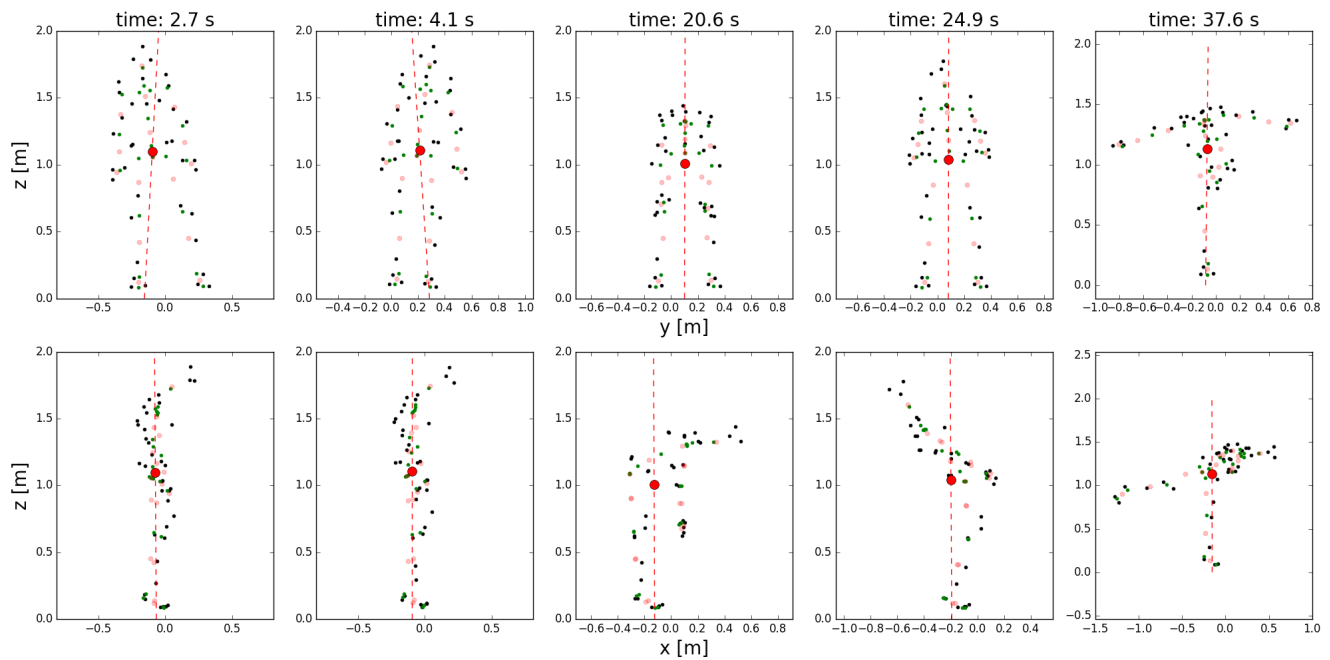


Fig. 4: Outtakes of the recorded and reconstructed data of the first filmed motion. The black dots represent the locations of the markers positioned on the filmed person, the green dots represent the reconstructed joint positions by the Motive software, the pale red dots represent the reconstructed locations of the COMs of each body link, the big red dot represents the location of the total COM of the filmed person and the red dotted line is the ZMP line. For the explanation of the body postures in each frame see the text.

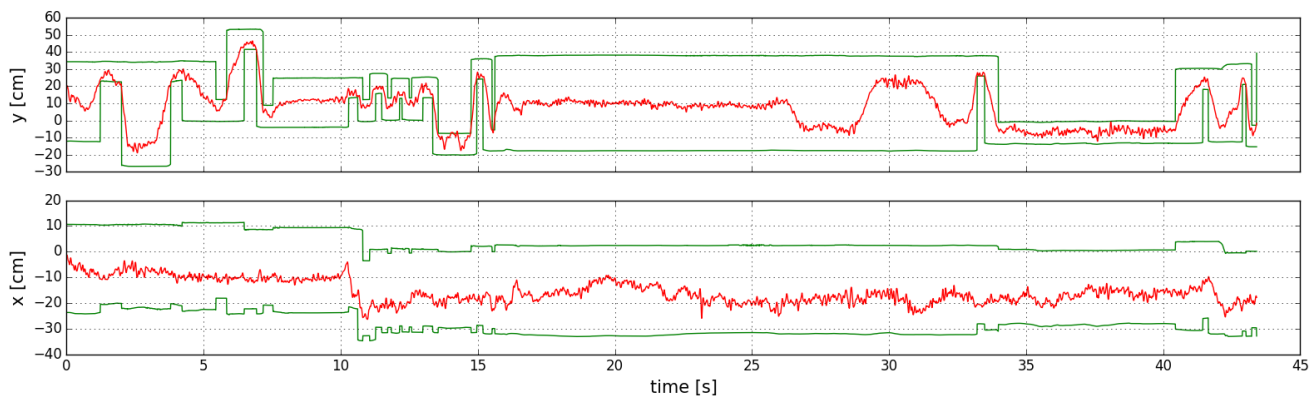


Fig. 5: The y (top) and x (bottom) coordinates of the ZMP (red lines) and the edges of the support polygon (green lines) for the first motion measurement expressed on the ground.

In the second motion filmed with the OptiTrack cameras the person is sitting down on a bench and standing up. Fig. 7 shows the frame sequence outtake of this motion.

In the first capture at time 0.2 s the person was stepping in front of the bench, in the second capture at 1.6 s the person was sitting down with his hands positioned on the bench, in the third capture at time 5.4 s the person was seated and in the fourth and fifth captures at 8.2 s and 8.8 s, respectively, the person was standing up.

As throughout this motion the person was not supported only by his feet on the ground but also by his hands and bottom on the bench, the support polygon does not extend only on the ground but also on the bench. This is why the standard Cartesian definition of the ZMP and the support polygon on the ground can not be used but the angular redefinition of these quantities must be applied. Fig. 8 shows the angles of the ZMP and the edges of the support polygon obtained with (3) and (5), respectively. When the person was supported also by his hands and his bottom on the bench, the x coordinate of his heel from (5) was substituted with the x coordinate of his hand or bottom, while the height

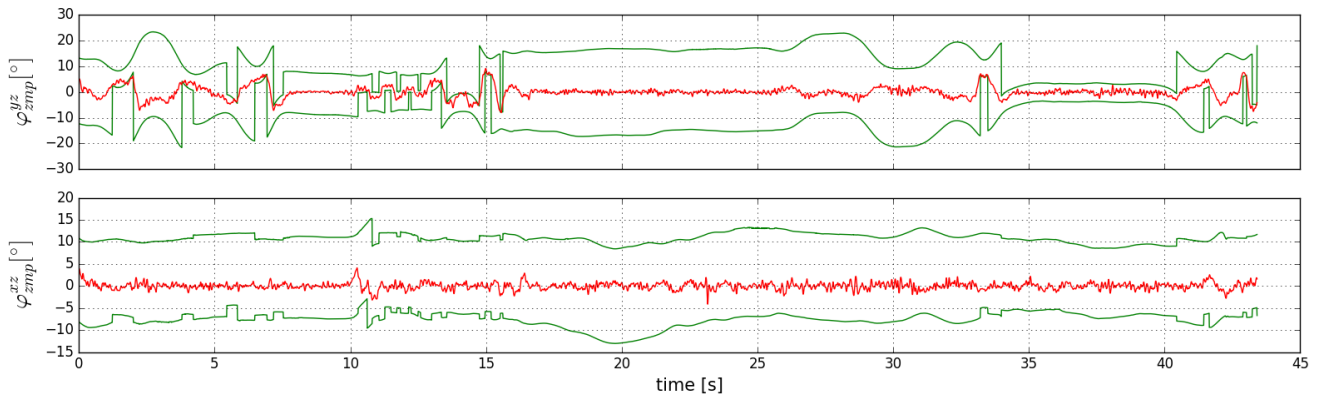


Fig. 6: The angles of the ZMP (red lines) and the edges of the support polygon (green lines) in the lateral (top) and the sagittal (bottom) planes of the measured person, for the first motion measurement

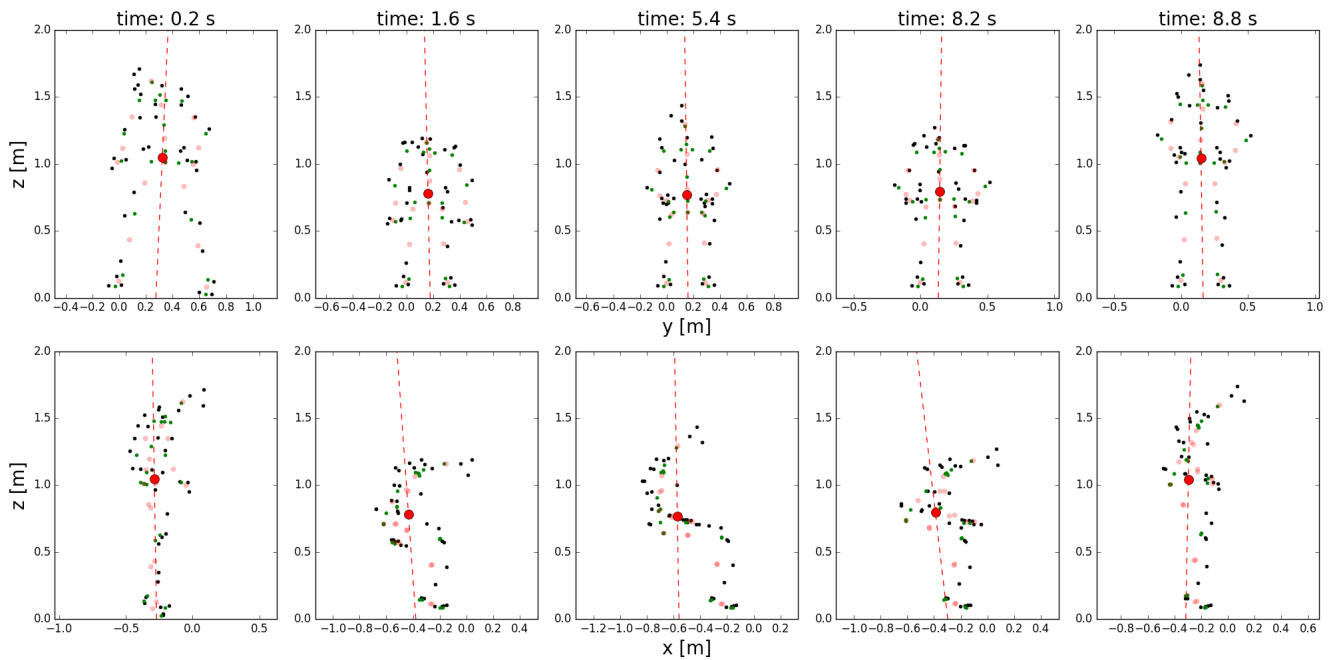


Fig. 7: Outtakes of the recorded and reconstructed data of the second filmed motion of sitting and standing. For the explanation of the meaning of different symbols see the text under Fig. 4. For the explanation of the body postures in each frame see the text.

of the COM was recalculated relatively to the bench height. In this measurement the ZMP was always within the support polygon which means that the person was stable all the time. In the beginning of the filming when the person started to sit down, he was supported only by his feet. At time 1.3 s he placed his fingers and at time 1.5 s also his palms on the bench, which can be seen as a widening of the support polygon angles in both the lateral and the sagittal planes, as the support polygon extended from his feet to the locations on the bench, where he was supported. At time 1.7 s the person sat on the bench and lifted up his arms

from the bench, which can be seen as a narrowing of the support polygon in the lateral plane and widening of the support polygon in the sagittal plane. The person was sited till the time 7.8 s, when he started to stand up which caused the narrowing of the support polygon in the sagittal plane, as the outermost location on his thighs where he was still in contact with the bench was approaching the edge of the bench above his feet. In the moment when he detached from the bench and was supported only by his feet at 8.1 s, the ZMP moved within the support polygon limited by the edges of his feet. Immediately after he was supported only by his feet the ZMP angle was in the sagittal plane close to the angle of his heel, but when he straightened up, the ZMP angle moved approximately in the middle between the angles

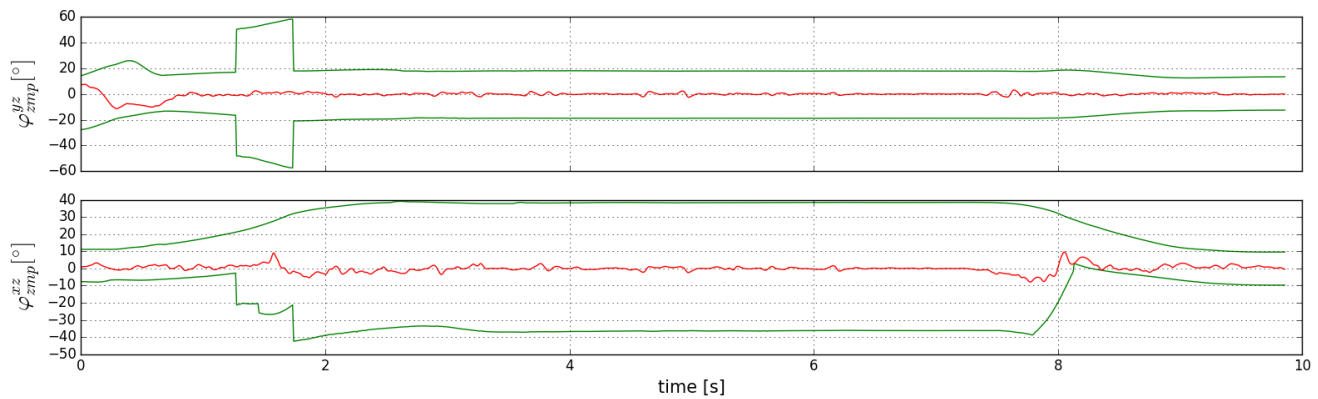


Fig. 8: The angles of the ZMP (red lines) and the edges of the support polygon (green lines) in the lateral (top) and the sagittal (bottom) planes of the measured person, for the second measurement of sitting and standing.

of his heels and his toes.

IV. CONCLUSIONS

The standard Cartesian definition of the ZMP on the ground is not suitable for the stability analysis in the situations when for example the ZMP lies at large distances from the investigated systems or if the system is supported at different heights. But in all these scenarios the angular definition of the ZMP can be used. If the system has support mechanisms that can act at different heights, such as for example a humanoid with arms, the situations when the ZMP would lie at far distances from the humanoid on the ground can be easily handled if the humanoid is supported at angles that embed the ZMP angle and if the friction forces and the forces in his joints allow for the satisfaction of the torque balance equation.

In this article it is shown how the stability of a person can be examined in the case when the person is supported only by his feet on the ground and in the case when the person is supported at different heights. In the first case both definitions of the ZMP and the support polygon were used, the one that defines these quantities in the Cartesian coordinates on the ground and the one that defines them as angles around the COM of the measured person. But in the second case, when the person was sitting on a bench and standing up, he was supported at different heights and the standard Cartesian definition of the ZMP and the support polygon on the ground could not be used and therefore our angular redefinition was applied. This way the stability of the person could be monitored in all the situations, no matter where the person was supported and whether the support mechanisms were only his feet or also his hands and his bottom. But the angular definitions of the ZMP and the support polygon are general and can therefore be applied to any system.

ACKNOWLEDGMENT

This work was supported by the Slovenian Research Agency grant N2-0153.

REFERENCES

- [1] T. Petrič, L. Žlajpah, G. Garofalo, and C. Ott, "Walking control using adaptive oscillators combined with dynamic movement primitives," 01 2013.
- [2] T. Petrič, A. Gams, J. Babič, and L. Žlajpah, "Reflexive stability control framework for humanoid robots," *Autonomous Robots*, vol. 34, 05 2013.
- [3] A. J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: A review," *Neural Networks*, vol. 21, no. 4, pp. 642 – 653, 2008, robotics and Neuroscience.
- [4] L. Righetti and A. Ijspeert, "Programmable central pattern generators: An application to biped locomotion control," vol. 2006, 06 2006, pp. 1585 – 1590.
- [5] J. Babič and J. Lenarčič, "Optimization of biarticular gastrocnemius muscle in humanoid jumping robot simulation," *Int. J. Humanoid Robotics*, vol. 3, pp. 219–234, 2006.
- [6] L. Lahajnar, A. Kos, and B. Nemeč, "Skiing robot - design, control, and navigation in unstructured environment," *Robotica*, vol. 27, pp. 567–577, 07 2009.
- [7] T. Petrič, A. Gams, J. Babič, and L. Žlajpah, "Reflexive stability control framework for humanoid robots," *Autonomous Robots*, vol. 34, 05 2013.
- [8] M. Vukobratović and D. Juričić, "Contribution to the synthesis of biped gait," *IEEE transactions on bio-medical engineering*, vol. 16 1, pp. 1–6, 1969.
- [9] M. Vukobratović and B. Borovac, "Zero-moment point - thirty five years of its life," *I. J. Humanoid Robotics*, vol. 1, pp. 157–173, 03 2004.
- [10] T. Brecej and T. Petrič, "Angular dependency of the zero moment point," in *Advances in Service and Industrial Robotics*, S. Zeghloul, M. A. Laribi, and J. Sandoval, Eds. Cham: Springer International Publishing, 2021, pp. 135–144.
- [11] S. Kajita and K. Tani, "Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode," in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, 1991, pp. 1405–1411 vol.2.
- [12] Optitrack motion capture and 3d tracking system. <https://optitrack.com/>.
- [13] Motive optical motion capture software. <https://optitrack.com/software/motive/>.
- [14] P. de Leva, "Adjustments to zatsiorsky-seluyanov's segment inertia parameters," *Journal of Biomechanics*, vol. 29, no. 9, pp. 1223–1230, 1996. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0021929095001786>
- [15] Y. T. E. Todorov and T. Erez, *MuJoCo: Modeling and Simulation of Multi-Joint Dynamics with Contact*, 0th ed., 2013.
- [16] V. Zatsiorsky and V. Seluyanov, in *Biomechanics VIII-B*. Human Kinetics, 1983, p. 1152–1159.