

Vodenje robotske roke s kretnjami in z uporabo sistema ROS

Domen Ulbl

Mentor: izr. prof. dr. Aleš Hace

**Fakulteta za elektrotehniko računalništvo in informatiko
Laboratorij za industrijsko robotiko, Koroška c. 46, 2000 Maribor
domen.ulbl@um.si, ales.hace@um.si**

Control of a Robotic Arm Using Human Gestures and ROS

People without expert knowledge of robotics may find it difficult to teach robots by usual methods. The robot teaching consists of the user demonstration of the task, which is conventionally performed by manual robot guiding, and subsequent repetition of the task by the robot. In this article, we will present a simple method for robotic task demonstration for advanced teaching methods based on a virtual avatar that mimics the motion of the user's arm. The mapping of human motion to the robot's trajectory has the potential to make the task demonstration easier and more intuitive. The system used for testing of our method is composed of two Myo Armbands and a PC running Linux with Robotic operating System and Kinova Mico robotic arm. In the first part of the article, the system and the teaching method are presented. In the second part, we evaluate our method by performing a simple robotic task.

Kratek pregled prispevka

Za ljudi brez specifičnega znanja iz robotike je učenje robotov s tradicionalnim programiranjem pogosto zahtevno. Učenje robota je sestavljeno iz uporabnikove demonstracije naloge, ki se konvencionalno izvede z ročnim vodenjem robota, in robotovega samostojnega ponavljanja te naloge. V članku bomo predstavili enostavno metodo demonstracije robotske naloge za napredne postopke učenja robotov. Metoda temelji na nadziranju virtualnega avatarja, ki posnema gibanje človeške roke. S preslikavo tega gibanja na želeno trajektorijo robota lažje načrtujemo robotsko nalogo znotraj računalniške simulacije. Virtualnega avatarja je sicer možno uporabiti tudi v kombinaciji s sistemom obogatene resničnosti, ko prilagajamo robotsko nalogo na realnem robotu. Sistem za demonstracijo je sestavljen iz senzorskih zapestnic Myo podjetja Thalmic Labs, osebnega računalnika s programsko opremo ROS in kolaborativne robotske roke Kinova Mico. V prvem delu članka opišemo sistem in metodo vodenja robota, v drugem delu pa testiramo kakovost interakcije med človekom in robotom na primeru preproste robotske naloge.

1 Uvod

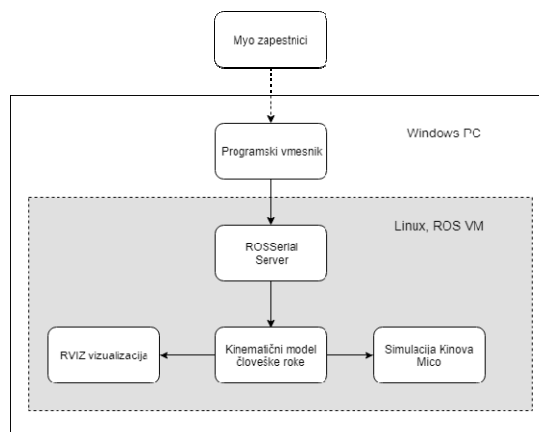
Roboti se v vedno večji meri pojavljajo izven industrijskega okolja z namenom izboljšanja produktivnosti in kakovosti v vsakdanjem življenju [1], [2]. Njihovo okolje postaja manj strukturirano in manj predvidljivo, naloge pa raznovrstnejše in kompleksnejše. Uspešno opravljanje naloge je v veliki meri odvisno od kakovosti navodil, ki jih robot prejme med učenjem naloge. Učenje je sestavljeno iz definiranja robotske naloge in robotovega ponavljanja le-te. Način definiranja robotske naloge lahko obsega metode programiranja z nadzorno-programirno napravo, »offline« programiranje v simulaciji in učenje z demonstracijo [6]. Prva dva načina zahtevata precej strokovnega znanja, ki ga uporabniki izven industrijskega okolja pogosto nimajo. Učenje z demonstracijo je zanje bolj naravno in ga hitreje usvojijo. Metode demonstracije vključujejo kinestetično vodenje robota s silo in teleoperiranje [3].

V našem članku predstavljamo sistem za teleoperiranje robota, ki temelji na zajemanju gibanja človeške roke in preslikavi tega gibanja na robotsko roko. Za zajem gibanja človeške roke uporabljamo nosljive zapestnice z inercialnimi senzorji in kinematični model človeške roke.

2 Opis sistema

Sistem za demonstracijo principa vodenja je sestavljen iz dveh zapestnic Myo, ki sta preko Bluetooth povezave povezani na osebni računalnik, na katerem se izvaja simulacija delovanja robota. Programska oprema Myo Connect na sistemu Windows sprejema signale iz zapestnic in jih preko TCP/IP protokola posreduje virtualnemu računalniku, na katerem teče Ubuntu Linux s programsko opremo ROS (ang. Robotic Operating System) [9]. Znotraj ROS se sočasno izvajajo trije procesi. Prvi je TCP/IP strežnik, ki sprejema signale zapestnic iz Windows gostiteljskega sistema in jih posreduje naprej. Drugi proces iz prejetih

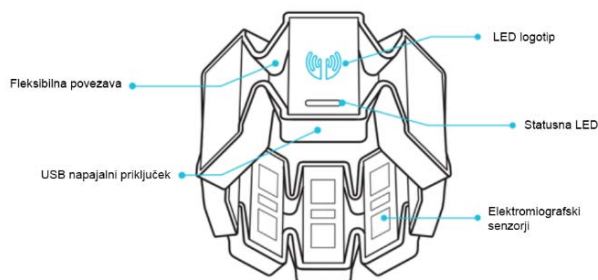
meritev izračuna konfiguracijo kinematičnega modela človeške roke in želene sklepne koordinate robotske roke. Tretji proces izvede simulacijo gibanja robota na podlagi sklepnih koordinat v okolju V-REP. Sistem je grafično predstavljen na sliki 1.



Slika 1: Blok diagram sistema.

2.1 Myo zapestnica

Senzorska zapestnica Myo je proizvod podjetja Thalmic Labs in omogoča določanje svoje orientacije v prostoru [10]. Meritve se izvajajo z enoto za zajem gibanja Invensense MPU-9150, ki vsebuje triosni pospeškometer, žiroskop in magnetometer. Meritve pospeška in kotne hitrosti se prenašajo preko I2C komunikacije na Cortex M4 mikrokontroler, ki izračunava orientacijo z algoritmi senzorskega zlivanja. Zapestnica omogoča identifikacijo gest uporabnikove dlani na podlagi meritve mišične aktivnosti, za kar skrbi osem elektromiografskih senzorjev na notranji strani zapestnice. Vsi podatki se prenašajo na osebni računalnik preko Bluetooth modula NRF51822 [10].



Slika 3: Deli zapestnice Myo (povzeto po [10]).

2.2 *Robotska roka Kinova Mico*

Kinova Mico je šestosna kolaborativna robotska roka, namenjena asistenci gibalno oviranim osebam in raziskavam interakcije med človekom in robotom [7]. Z dosegom 70 cm in nosilnostjo 750 g v iztegnjenem položaju se približa sposobnostim človeške roke. Kinematična struktura roke spominja na človeško in jo lahko razdelimo na dva dela, običajni »RRR« del iz prvih treh sklepov in posebno ukrivljeno zapestje definirano z zadnjimi tremi sklepi. Prijemanje objektov je omogočeno z dvoprstnim pod-aktuiranim prijemalom.

2.3 *Myo Connect in programski vmesnik*

Myo Connect je programska oprema za komunikacijo med zapestnicama in uporabniškimi programi na osebem računalniku, ki omogoča dostop do vseh funkcij zapestnice preko programskega vmesnika C++ API (ang. Application Program Interface). Nudi tudi kalibracijo elektromiografskih senzorjev za natančno klasifikacijo gest dlani.

Programski vmesnik prebere meritve iz zapestnic in jih pošlje sistemu ROS preko TCP/IP protokola.

2.4 *Robotic Operating System*

Robotic Operating System [9] je zbirka programskih knjižnic in orodij za razvoj robotskih aplikacij za Linux operacijske sisteme. Na najnižjem nivoju nudi vmesnik za pošiljanje sporočil med računalniškimi procesi, ki obratujejo sočasno. Vsak proces tvori vozlišče (ang. Node), ki glede na svojo vlogo založnika (ang. Publisher) ali naročnika (ang. Subscriber) pošilja ali prejema sporočila, imenovana teme (ang. Topics) preko TCP/IP ali UDP protokola. Posamezna vozlišča so neodvisna in lahko obratujejo v ločenih procesorskih nitih na različnih računalnikih v omrežju. Odpoved enega vozlišča običajno ne ogrozi delovanja celotnega sistema. Vozlišče običajno opravlja le neko določeno nalogo, kot je branje podatkov iz sensorja, računanje transformacij med koordinatnimi sistemi ali

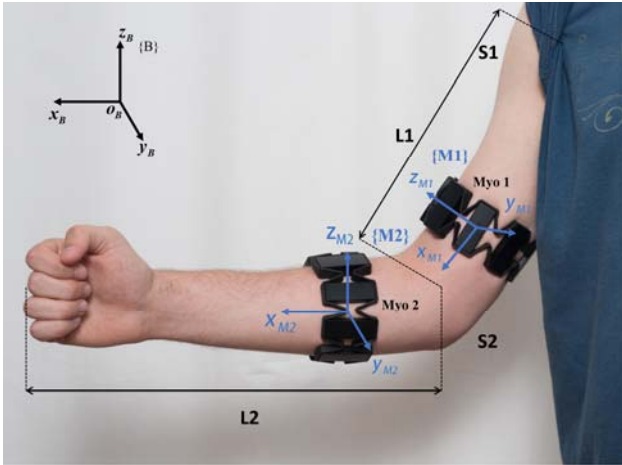
nadziranje aktuatorjev. Takšna modularna struktura omogoča veliko fleksibilnost pri snovanju sistema. Prehod med nadziranjem simulacije robota in dejanskim robotom je tako enostavnejši, ker poteka nadzor obeh na enak način. Funkcionalnosti ROS je mogoče nadgraditi z zmogljivimi knjižnicami, kot so knjižnica za načrtovanje trajektorij MoveIt!, knjižnica za strojni vid OpenCV in simulacijsko okolje Gazebo [8]. ROS je odprtokoden projekt pod licenco Creative Commons, tako da ga je mogoče prosto dopolnjevati. Pisanje kode poteka v programskih jezikih C++ ali Python, posebna orodja olajšajo integracijo novih vozlišč v sistem. Pomembna prednost ROS je velika skupnost uporabnikov, ki je do zdaj prispevala več kot 2000 paketov novih orodji in funkcionalnosti v obsegu 14 milijonov vrstic kode [9].

2.5 *V-REP*

Virtual Robotics Experimentation Platform (V-REP) je programsko okolje za simulacijo kinematike in dinamike poljubnih robotskih mehanizmov [4]. Program za fizikalne izračune upošteva trke med objekti in vpliv sil. Vsak model robota ima definirane sklepe, ki so lahko prizmatični ali rotacijski ter imajo definirane omejitve svoje poti, pospeškov, navorov... Model je kontroliran s skripto v jeziku Lua skript, ki se z vtičnikom RosInterface povezuje s sistemom ROS. Predstavljena simulacija robota Kinova Mico kombinira izračun gibanja robota preko direktne kinematike z odzivanjem na trke z okolico in s samim seboj.

3 *Metoda vodenja robotske roke*

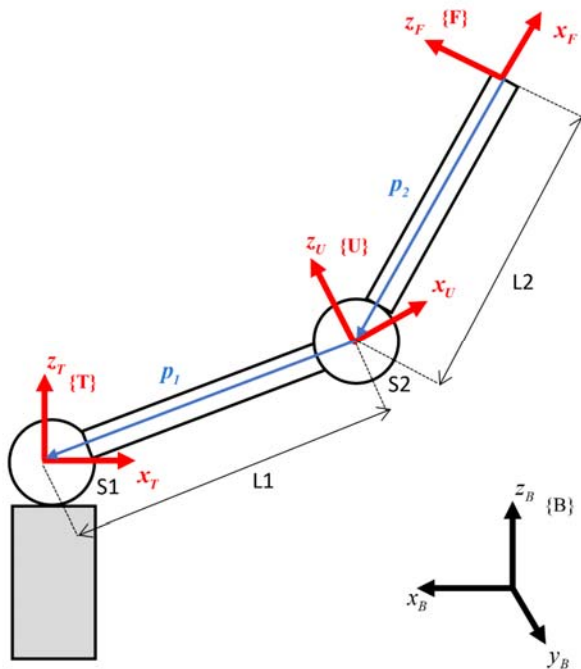
Uporabnik nosi dve zapestnici na eni roki; Myo 1 na podlakti in Myo 2 nadlakti. K zapestnici Myo 1 pripada koordinatni sistem $\{M1\}$, k zapestnici Myo 2 pa pripada koordinatni sistem $\{M2\}$. Dolžini nadlakti in podlakti sta označeni z L_1 in L_2 . Ramenski in komolčni sklep sta označena z S_1 in S_2 , kot je prikazano na sliki 4. Iz orientacije obeh zapestnic in dolžin nadlakti in podlakti L_1 in L_2 lahko rekonstruiramo pozo roke.



Slika 4: Roka uporabnika z nameščenima zapesticama M_1 na nadlakti (segment L_1) in M_2 na podlakti (segment L_2).

3.1 Poenostavljen kinematični model človeške roke

Kinematični model človeške roke, prikazan na sliki 5, je poenostavljen, in ima vlogo virtualnega vmesnega mehanizma pri prenosu gibanja človeške roke na ciljno robotsko roko. Sestavljen je iz dveh segmentov dolžine L_1 in L_2 ter dveh sferičnih sklepov S_1 in S_2 .



Slika 5: Kinematični model človeške roke.

Kinematični model je matematično opisan z homogenimi transformacijskimi matrikami med koordinatnimi sistemi {T} (ang. Torso), {U}

(ang. Upper arm) in {F} (ang. Forearm). Koordinatni sistem {T} Vektorja p_1 in p_2 sta v koordinatnih sistemih {U} in {F} izražena, kot kažeta enačbi (1) in (2).

$${}^U p_1 = [-L1 \ 0 \ 0]^T \quad (1)$$

$${}^F p_2 = [-L2 \ 0 \ 0]^T \quad (2)$$

Kinematični model je zapisan v enačbah (3), (4) in (5),

$${}^T T_F = {}^T T_U {}^U T_F \quad (3)$$

$${}^T T_U = \begin{bmatrix} {}^T R_U & -{}^T R_U {}^U p_1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^T R_U & {}^T p_U \\ 0 & 1 \end{bmatrix} \quad (4)$$

$${}^U T_F = \begin{bmatrix} {}^U R_F & -{}^U R_F {}^F p_2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^U R_F & {}^U p_F \\ 0 & 1 \end{bmatrix} \quad (5)$$

pri čemer matrika ${}^T T_U$ opisuje relativno transformacijo iz k. s. {T} v k. s. {U}, ${}^U T_F$ pa relativno transformacijo iz k. s. {U} v k. s. {F}, vektorja ${}^T p_U$ in ${}^U p_F$ predstavljata relativno pozicijo koordinatnih sistemov {U} in {F} v {T} oz. {U}:

$${}^T p_U = -{}^T R_U {}^U p_1 \quad (6)$$

$${}^U p_F = -{}^U R_F {}^F p_2 \quad (7)$$

Orientaciji zapestic M_1 in M_2 sta izraženi glede na fiksni koordinatni sistem {T}. Orientacija zapestnice M_1 se tako ujema z orientacijo koordinatnega sistema {U} glede na {T} in jo v enačbi (5) opišemo z rotacijsko matriko ${}^U R_F$. Orientacija zapestnice M_2 ustreza orientaciji {F} glede na {T}, ki jo lahko opišemo z rotacijsko matriko ${}^T R_F$. Orientacijo {F} glede na {U} lahko izračunamo po enačbi (8):

$${}^U R_F = {}^T R_U^T \cdot {}^T R_F \quad (8)$$

Končni kinematični model je opisan v enačbi (9):

$${}^T\mathbf{T}_F = \begin{bmatrix} {}^T\mathbf{R}_U & -{}^T\mathbf{R}_U {}^U\mathbf{p}_1 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} {}^U\mathbf{R}_F & -{}^U\mathbf{R}_F {}^F\mathbf{p}_2 \\ 0 & 1 \end{bmatrix}$$

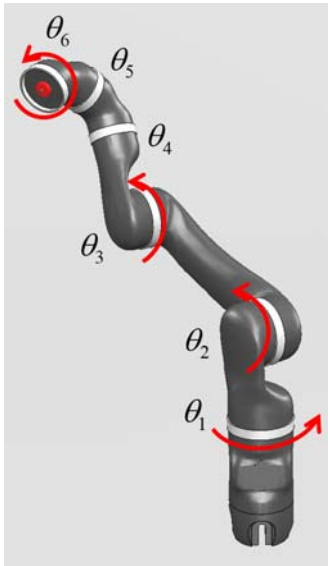
$$= \begin{bmatrix} {}^T\mathbf{R}_U {}^U\mathbf{R}_F & -{}^T\mathbf{R}_U {}^U\mathbf{R}_F {}^F\mathbf{p}_2 - {}^T\mathbf{R}_U {}^U\mathbf{p}_1 \\ 0 & 1 \end{bmatrix}$$

$${}^T\mathbf{T}_F = \begin{bmatrix} {}^T\mathbf{R}_F & {}^T\mathbf{p}_F \\ 0 & 1 \end{bmatrix} \quad (9)$$

Pozicija konca modela roke (izhodišče {F}) glede na {T} je torej podana s pozicijskim vektorjem ${}^T\mathbf{p}_F$ in rotacijsko matriko ${}^T\mathbf{R}_F$.

3.2 Preslikava na sklepne koordinate robotske roke

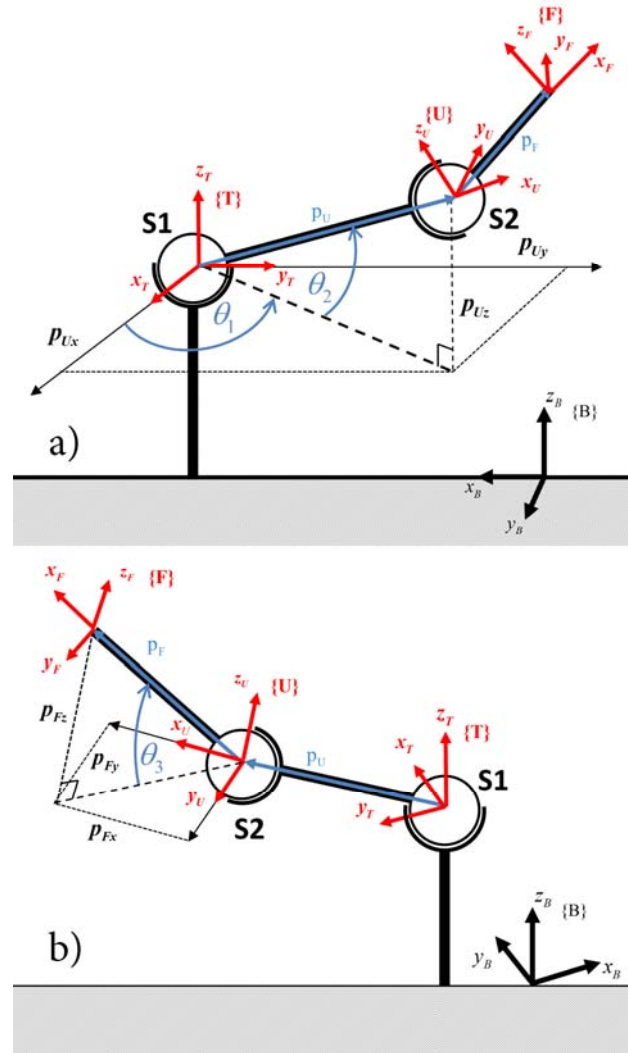
Na sliki 6 je prikazan model robota Kinova Mico z označenimi sklepi in nakazanimi smermi vrtenja sklepov.



Slika 6: Model robota Kinova Mico.

Koti θ_1 , θ_2 in θ_3 določajo zasuke 1. 2. in 3. sklepa, ki v grobem že določajo pozicijo robotskega konca. S kotom θ_6 vrtimo 6. sklep, ki služi kot del zapestja robota. Ta kot določimo z zaznavanjem gest uporabnikove dlani preko elektromiografskih (EMG) signalov. Kota zasuka 4. in 5. sklepa, θ_4 in θ_5 , zaenkrat ostajata fiksna.

Iz lege vektorjev ${}^T\mathbf{p}_U$ in ${}^F\mathbf{p}_U$ na modelu človeške roke lahko izračunamo kote θ_1 , θ_2 in θ_3 . Izračuni kotov so zapisani v enačbah (10), (11) in (12). Koti so na modelu človeške roke prikazani na sliki 7.



Slika 7: Robotski sklepni koti na modelu človeške roke: a) θ_1 in θ_2 , b) θ_3 .

$$\theta_1 = \tan^{-1}\left(\frac{{}^T\mathbf{p}_{Uy}}{{}^T\mathbf{p}_{Ux}}\right) \quad (10)$$

$$\theta_2 = \tan^{-1}\left(\frac{{}^T\mathbf{p}_{Uz}}{\sqrt{{}^T\mathbf{p}_{Ux}^2 + {}^T\mathbf{p}_{Uy}^2}}\right) \quad (11)$$

$$\theta_3 = \tan^{-1}\left(\frac{{}^U\mathbf{p}_{Fz}}{\sqrt{{}^U\mathbf{p}_{Fx}^2 + {}^U\mathbf{p}_{Fy}^2}}\right) \quad (12)$$

3.3 Geste dlani

Senzorji mišične aktivnosti na zapestnici Myo merijo električno napetost, ki nastane ob skrčitvi ali sprostitvi mišice [5]. Obdelava signalov je zahtevna, a Myo Connect vseeno omogoča klasifikacijo vzorcev signalov v pet značilnih gest, prikazanih na sliki 8.



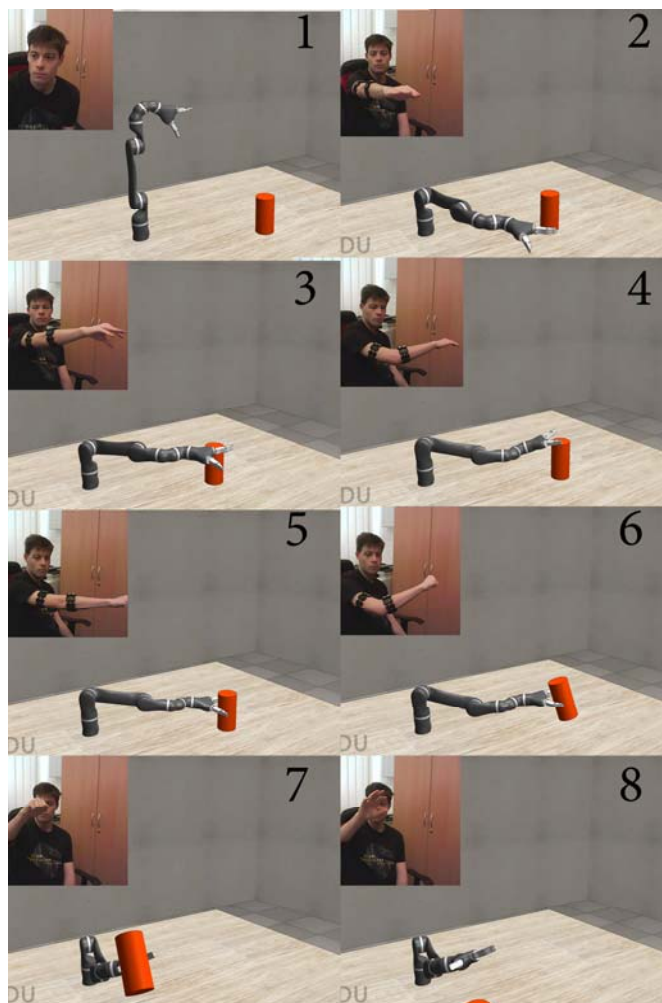
Slika 8: Geste dlani, klasificirane iz EMG signalov.[11]

Prisotnost gest uporabimo v vodenju kot digitalne signale, ki oponašajo uporabnikov pritisk na neko tipko. Tako kot zapestja θ_6 nastavljamo z gestama »Pomahaj levo« (ang. Wave Left) in »Pomahaj desno« (ang. Wave Right), odpiranje in zapiranje prijema pa nadziramo z gestama »Odpri dlan« (ang. Fingers Spread) in »Stisni v pest« (ang. Fist).

4 Rezultati

Delovanje sistema in ustreznost opisanega sistema vodenja je zahtevno ocenjevati na podlagi številskih izpisov, zato si pomagamo s simulacijo. Robotska roka je postavljena na virtualno mizo, na kateri se nahaja rdeč valj. Uporabnikova naloga je z gibanjem svoje roke prijeti in prestaviti rdeč valj. Potek simulacije je prikazan na videu [12], izsek je prikazan na sliki 9.

Nadzor nad robotom je za uporabnika precej naraven, ker ne zahteva veliko zavestnega razmišljanja. Celoten računalniški sistem je odziven, saj simulacija poteka praktično v realnem času. Ob poskusu prijete objekta pride v ospredje uporabnikova omejena percepcija dogajanja na zaslonu. Čeprav je geometrija izrisana tridimenzionalno, uporabnik iz običajnega računalniškega zaslona ne pridobi informacije o globini, kar izvajanje naloge oteži.



Slika 9: Posnetek simulacije virtualnega pobiranja predmeta z zapestnicami Myo.

5 Zaključek in nadaljnje delo

V članku smo predstavili princip vodenja robotske roke s kretjnami človeške roke. V ta namen smo razvili sistem na podlagi zmogljivega okolja ROS. Nadzor nad robotsko simulacijo od uporabnika ne terja večjega navora. Metodo bomo v prihodnosti dopolnili z dodatkom izračuna dveh kotov zapestja, ki za enkrat še nista določena. Težave s percepcijo, opisane v prejšnjem poglavju, bi lahko zmanjšali z vključitvijo očal za virtualno resničnost, ki s stereoskopsko sliko nudijo vizualno informacijo o globini. Za večjo robustnost vodenja bi bilo potrebno dodati nekaj omejitev v gibanje sklepov, da preprečimo morebitne trke s podlago. Končni cilj projekta je implementacija vodenja na realnem robotu.

6 Literatura

- [1] B. Akgun, *Novel Interaction Strategies for Learning from Teleoperation*: Georgia Institute of Technology, Atlanta, ZDA, 2012.
- [2] B. D. Argall, *A survey of robot learning from demonstrations*, *Robotics and Autonomous Systems*, Volume 57, Issue 5, 31 May 2009, Pages 469-483.
- [3] M. B. Barrette, *What Does Collaborative Robot Mean?*, Kanada, Robotiq Inc., 2015, Dostopno na: <https://blog.robotiq.com/what-does-collaborative-robot-mean>
- [4] Coppelia Robotics, *Virtual Robot Experimentation Platform USER MANUAL*, verzija 3.5.0, Zürich, Švica, Coppelia Robotics GmbH, 2018, Dostopno na: <http://www.coppeliarobotics.com/>
- [5] *Electromyography*, Dostopno na: <https://en.wikipedia.org/wiki/Electromyography>
- [6] A. Hace *Osnove robotike (zapiski predavanj)*. Maribor: Fakulteta za elektrotehniko, računalništvo in informatiko, 2016. Interno gradivo.
- [7] Kinova Inc., *KINOVA Ultra lightweight robotic arm User Guide*, Kanada, Kinova Inc., 2018 Dostopno na: https://beta.kinovarobotics.com/sites/default/files/KINOVA_MICO_Robotic_arm_USER_GUIDE.pdf
- [8] Nyro, *8 reasons why you should use ROS for robotics projects*, Dostopno na: <https://niryo.com/2018/01/8-reasons-use-ros-robotics-projects/>
- [9] Open Source Robotics Foundation, *About ROS*, Dostopno na: <http://www.ros.org/core-components/>
- [10] Thalmic Labs. *Myo SDK Manual 0.9.0*. Ontario, Kanada: Thalmic Labs, 2013. Dostopno na: https://developer.thalmic.com/docs/api_reference/platform/index.html
- [11] Thalmic Labs. *Using the Keyboard Mapper in Myo Connect*. Ontario, Kanada: Thalmic Labs, 2013. Dostopno na: <https://support.getmyo.com/hc/en-us/articles/204660665-Using-the-Keyboard-Mapper-in-Myo-Connect>
- [12] Ulbl D. *Intuitive control of Kinova Mico using Myo Armband and ROS*. Maribor, Fakulteta za elektrotehniko, računalništvo in informatiko, 2019. Dostopno na: <https://bit.ly/2TNEq5T>