

# Intuitivno učenje sodelujoče robotske roke s pomočjo mešane resničnosti

Maja Glušič<sup>1</sup>

<sup>1</sup>Maja Glušič, Fakulteta za elektrotehniko, računalništvo in informatiko, Koroška cesta 46, 2000 Maribor  
E-pošta: maja.glusic@student.um.si

## The intuitive teaching collaborative robotic arm system by mixed reality approach

**Abstract.** In this paper we deal with development of an experimental robotic system with human-robot interface by mixed reality. The goal of this experimental system is to enable user without skills to create a simple robotic application. The system is divided in three parts. The first part includes the user interface in which, the user manipulates elements of mixed reality on the real robotic workspace. The second part logs positions of the virtual objects. The third part of the system defines the robot path, which is then sent to the user and the robot in the virtual environment.

### 1 Uvod

Industrija 4.0 je osredotočena na izboljšanje produktivnosti in uporabniške izkušnje. V zadnjih letih je interakcija človek-robot priljubljen koncept v industriji [1], [2]. Namreč trg zaradi variabilnosti izdelkov, narekuje potrebo po fleksibilnem proizvodnem okolju. Takšen pristop oteži načrtovanje in avtomatizacijo proizvodnih procesov. S pristopom interakcije človek-robot, je mogoča združitev robotskih in človeških veščine za doseganje učinkovitejšega, fleksibilnejšega in produktivnejšega proizvodnega sistema. V tem primeru so roboti odgovorni za fizično naporne, monotone in ponavljajoče naloge, medtem ko se ljudje lotevajo nalog, ki zahtevajo spretno in fleksibilno manipulacijo, zahtevno zaznavanje ali inteligentno odločanje. Interakcija človek-robot lahko pomaga tako inženirjem pri učinkovitem preoblikovanju proizvodnje, kot tudi delavcem, pri hitrejši in lažji prilagoditvi na spremembe v proizvodnem toku. Pomembna tehnologija, ki zadeva interakcijo med človekom in robotom je mešana resničnost (MR) [3]. Prozoren zaslon, ki ga uporablja MR naprava HoloLens, ne izloči realnega okolja iz vidnega polja uporabnika. Operater lahko neposredno komunicira s fizičnim robotom na intuitiven način. MR naprave se uporabljajo v več vidikih proizvodnega procesa. Nekateri pomembni primeri uporabe so usposabljanje operaterjev, načrtovanje izdelkov, vzdrževanje, popravilo in pregled.

Zato so aplikacije mešane resničnosti za industrijske namene priljubljen predmet raziskovanja. V tem članku se bomo osredotočili na pomoč pri programiranju in enostavno manipulacijo robotske rok na sodelujočem robotu z uporabo demonstracijskih tehnik. Programiranje z demonstracijo je lahko smiselno orodje za operaterje, ki nimajo izkušenj in znanja programiranja. Na ta način bi bili sposobni naučiti

robotu določene naloge, tako, da sami opravijo nalogo in za tem robot nalogo ponovi. Novost te aplikacije temelji na prednostih, ki jih ponuja mešana resničnost pri programiranju robotu z uporabo naprave HoloLens in ROS [8] vmesnika. Takšen pristop programiranja ob izpopolnjenem sistemu, je lahko boljši zaradi preprostosti in intuitivnosti interakcije z robotom. Pristop lahko prihrani čas in trud, potreben za programiranje, zaradi vizualno načrtovane poti pa lahko preprečimo napake ali nevarne situacije. Članek smo razdelili na pet glavnih delov. V drugem delu članka bomo opisali že znane metode tovrstnih načinov interakcije z robotom, v tretjem delu pristop uporabnika skozi različne faze uporabe aplikacije, v četrtem delu obravnavamo arhitekturo in implementacijo glavnih delov sistema, v zadnjem razdelku pa bomo povzeli ugotovitve in nakazali smernice ter cilje za nadaljnje delo.

### 2 Sorodna dela

Jacopo Aleotti [5] je predlagal vizualno-haptični sistem za manipulacijo predmetov in učenje naloge s pomočjo človeške demonstracije. V ta namen se uporablja haptična naprava za interakcijo z objekti. Uporabniki haptične naprave niso solocirani z okoljem, kjer so prisotni realni predmeti. Prepoznavo realnih objektov na oddaljeni delovni površini poteka s pomočjo laserskega skenerja, nameščenega na robotski roki. Za povratno informacijo o postavitvi objektov, sistem podpira okolje mešane resničnosti. Rezultati, dobljeni z izvedenimi poskusi, kažejo, da naučeni robotski sistem uspešno izvede nalogo. Martin Rudorfer [8] je predstavil intuitivno metodo programiranja, ki bi jo lahko uporabljal operater brez znanja programiranja robotov. Pri izvedbi sta bila uporabljena HoloLens in robot UR5. Glavni cilj je bil, da uporabnik izbere predmet in ga postavi na zeleno mesto, robot pa nalogo ponovi. Robot je začel s pridobivanjem slike, nato je prepoznal predmet in njegovo pozicijo. Po prepoznavanju predmetov so bili predmeti prikazani v okolju MR, virtualni objekti so se tako prekrivali z realnimi. Na krmilni modul robotu so bile poslana začetne in končne koordinate zelenih lokacij, izvedene so bile referenčne transformacije iz kamere na robotsko pozicijo. Tako je bilo mogoče izvesti nalogo izbire in postavitve.

### 3 Pristop in uporaba

Predlagana aplikacija je namenjena industrijskim delavcem, za pomoč pri programiranju robotu. Takšen pristop je lahko učinkovit, saj uporabnik operira z objekti, ki so po izgledu in fizikalnimi lastnostmi

podobni realnim. Uporabnik vidi robotske funkcije in predlagane poti še preden jih robot izvede, le-te v realnemu okolju niso vidne.

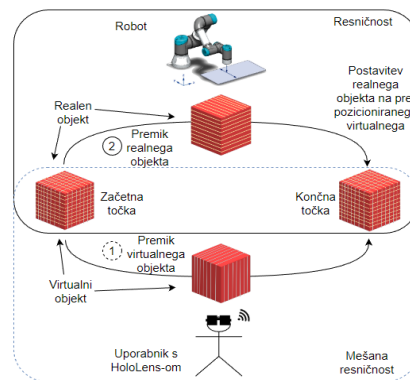
Z vidika uporabnikov sta glavna cilja aplikacije preprostost in intuitivnost pri načinu interakcije z robotsko roko. Primer te aplikacije omogoča učenje robota brez posebnih navodil ali predznanja robotike. V tem kontekstu je bil cilj, da je vmesnik preprost in jasan.

### 3.1 Faza inicializacije

Aplikacija združuje resničen svet z virtualnim, ki je znotraj HoloLens. Tako mora biti med obema svetoma povezava. Svet MR mora biti povezan z robotskim koordinatnim sistemom. Zato je bil postopek kalibracije obvezen del. V našem primeru smo postopek kalibracije izvedli z zaznavanjem QR kode, ki je postavljena na znano mesto. Metodo z uporabo markerjev smo uporabili za povezavo koordinatnih sistemov. Ko se aplikacija zažene mora uporabnik stati pred QR kodo, ki jo prepozna HoloLens. Tako je mogoče točke v uporabniškem prostoru preoblikovati v realni prostor.

### 3.2 Faza učenja

Po uspešnem skeniranju QR kode in pravilno orientirane virtualne površine, lahko uporabnik manipulira z virtualnimi predmeti, ki so postavljeni na mesta realnih. Manipuliranje z navideznimi predmeti je enostavno, naravno in intuitivno. Objekti so postavljeni na znano točko, začetna pozicija ni prepoznana. Uporabnik virtualne objekt postavi v zeleno točko na robotski delovni površini. Ko uporabnik konča s postavitvijo virtualnega objekta, se ob končanem stiku z objektom, ob njem pojavi virtualni gumb za potrditev končne pozicije. Ob potrditvi se točka končne pozicije in orientacija objekta pošlje v ROS [8] okolje. Ob prejeti točki v ROS-u [8] od načrtovalskega paketa zahtevamo, da ustvari trajektorjo poti robota od začetne, znane točke, do ciljne točke postavljenega objekta. Če je ciljni položaj mogoče doseči z robotom, se v Unity okolju ta pot tudi izriše. Predlagano pot uporabnik potrdi ali ovrže s pritiskom na virtualni gumb ob trajektorji. Z dodatnim gumbom uporabnik potrdi izvedbo te trajektorije. Sistem ima nekatere pomike v naprej določene npr. pot vračanja v začetno pozicijo in pot do začetne točke. Koncept uporabe je predstavljen na sliki 1. V začetni točki sta postavljena realni in na njem virtualni objekt. Ko uporabnik premakne željen virtualni objekt in ciljno točko premika potrdi, nalogo izvede tudi robot z realnim objektom.



Slika 1. Osnovni koncept sistema.

## 4 Arhitektura sistema in implementacija

Aplikacija je bila razvita z uporabo programskega okolja Unity [7], ki ga Microsoft predlaga za izdelavo aplikacij za njihovo napravo HoloLens. Programski jezik, ki ga uporablja Unity [7] je C# in knjižnica .NET. Vzporedno z razvojem virtualnega okolja je bil potreben razvoj na strani ROS-a [8]. V tem primeru je bil programski jezik, ki smo ga uporabili C++. V naslednjih podsekcijah smo opisali glavne dele razvitega sistema.

### 4.1 Komunikacija Unity-ROS

Prvi del razvojnega postopka je bila povezava aplikacije, ki teče znotraj HoloLens, s sistemom ROS [8]. Arhitektura sistema interakcije med človekom in robotom je predstavljena na sliki 3. Na levi strani je uporabniški vmesnik Microsoft HoloLens 2 z razvojnim okoljem Unity [7]. Na desni strani je ROS [8], ki deluje na Ubuntu OS. Za interakcijo z hologrami smo uporabili Mixed Reality Toolkit (MRTK) [9]. Na strani strežnika imamo okolje ROS Melodic in simulator Gazebo, to je potrebno za zagon robotske roke. Integracijo ROS [8] in Unity3D [7] omogoča implementacija odprtokodne knjižnice ROS-Sharp. Povezava z WebSocketom med Unity [7] in ROS [8] je vzpostavljena na vratih 10000. To omogoča, da ROS-Sharp serializira temo (»topic«). Tema se aktivira vsakič, ko Unity [7] poskuša poslati informacije o končni poziciji objekta ali drugih ukazih iz virtualnega okolja. V tem primeru bo Unity [7] deloval kot publisher, medtem ko bo ROS [8] simulator subscriber. Unity [7] pretvori podatkovne strukture C# v sporočilo JSON, podatki o poziciji pa se posredujejo prek "bridga". Z uporabo te povezave smo robota znotraj ROS-a [8] uvozili Unified Robot Description Format (URDF) datoteko. Format URDF zagotavlja vizualno in strukturno predstavitev, zato je uporaben za vizualizacijo robota na več mestih znotraj aplikacije. Poleg tega lahko prek povezave obeh naprav izmenjujemo informacije v obliki sporočil ROS [8]. S temi sporočili lahko MR aplikacija sporočila prejema v realnem času - planirane poti ali katere koli druge koristne informacije. In tudi nasprotno, lahko aplikacija MR pošilja sporočila v sistem ROS [8] na podlagi

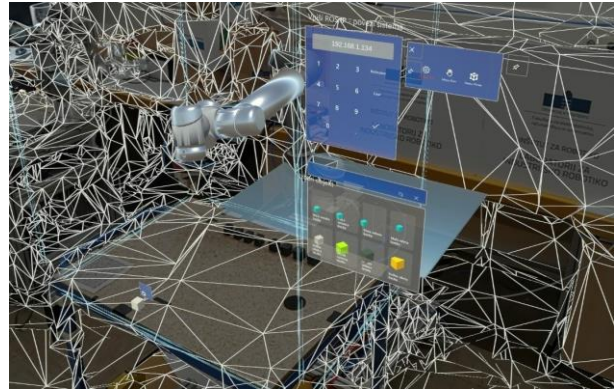
uporabniškega vnosa, v našem primeru pošilja pozicijo virtualnih objektov in ukaze drugih elementov uporabniškega vmesnika.

#### 4.2 Postavitev koordinatnih sistemov

Naslednja faza razvoja je bila povezava realnega, MR in ROS prostora. Za vzpostavitev te korelacije treh svetov smo uporabili QR kodo. Za zaznavanje QR kode uporabljamo okolje Vuforia [6]. Uporabnik QR kodo zajame v svoje vidno polje. QR kodo uporabljamo kot marker z značilkami, za postavitve virtualnega prostora na ustrezno mesto realnega. Ker so osi koordinatnega sistema v Unity [7] poravnane z levosučnim koordinatnim sistemom, večina MR aplikacij uporablja levo sučni koordinatni sistem. Da bi uskladili oba koordinatna sistema smo uporabili funkcijo Unity2Ros(), ki pretvori kvaternion Unity sporočila v ROS koordinatni sistem. Tako se koordinatna sistema okolja Unity in ROS ujemata [4].

#### 4.3 Virtualni elementi

V aplikaciji, zasnovani za MR, so za uporabniško interakcijo potrebna posebna orodja. Ta orodja zagotavlja Microsoft s pomočjo knjižnice za mešano resničnost MRTK (MixedRealityToolkit) [9]. Na podlagi tega kompleta orodij smo ustvarili elemente uporabniškega vmesnika, objekte za interakcijo v okolju MR in sceno, ki je združljiv s funkcijami, ki jih omogoča HoloLens. Vsak element uporabniškega vmesnika je posamezni objekt znotraj Unity [7], njegovo delovanje pa spremlja glavni skript, ki upravlja interakcijo med vsemi komponentami aplikacije. Objekte uporabnik premika z naravnimi kretnjami za manipulacijo. To je precejšnji približek pobiranja realnih predmetov. Ko uporabnik prime in premika virtualni predmet, je ta druge barve, tako uporabniku signaliziramo, kdaj ima stik z objektom. Prav tako smo virtualnim objektom dodelili fizikalne lastnosti objektov v realnem okolju. (gravitacijo, materiale, koeficient trenja na površini, približno maso objekta, viskoznost medija v katerem se giblje). Robotsko delovno površino smo omejili z virtualnimi ploščami, da lahko z objekti manipuliramo samo znotraj meja delovne površine. Sistem mešane resničnosti prikazuje slika 2.



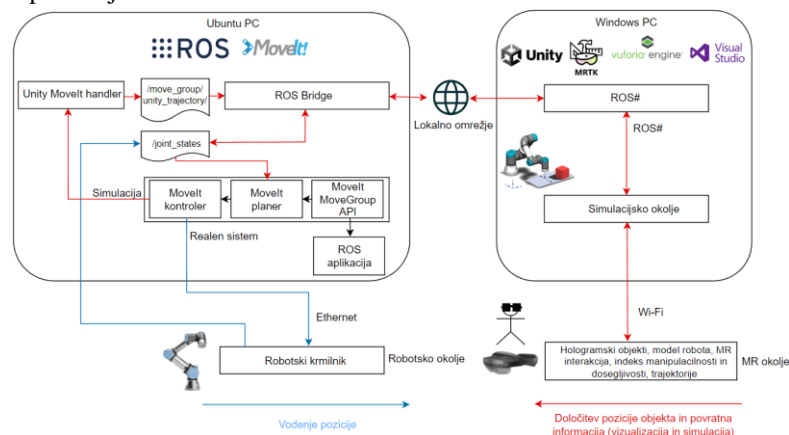
Slika 2. Realni sistem – okolje mešane resničnosti.

#### 4.4 Načrtovanje trajektorije -MoveIt

Vzporedno z razvojem na strani sistema Unity [7] je bilo potrebno zgraditi sistem na strani ROS-a [8]. Razvoj tega sistema lahko razdelimo na dva glavna dela. In sicer ustvarjanje v paketu MoveIt [8] in ustvarjanje vozlišča ROS [8], ki uporablja funkcije MoveIt [8] in obravnava sporočila ROS [8].

MoveIt [8] je odprtokodna knjižnica za načrtovanje gibanja z podporo za številne industrijske robote. Z orodjem, lahko zgradimo sceno z robotom UR in vsemi drugimi znanimi predmeti in na podlagi tega uporabimo načrtovalce gibanja za izračun poti. Potrebne geometrijske in kinematične informacije o robotu so na voljo prek datotek URDF v kombinaciji z konfiguracijskimi datotekami. V drugem delu smo ustvarili vozlišče ROS [8], ki je sposobno komunicirati tako z MoveIt-om kot z aplikacijo MR upravlja potrebna sporočila in jih pošilja ustreznim ROS [8] temam. V našem primeru je odgovoren predvsem za:

- Prezeti ciljne pozicije za načrtovanje ali izvedbo iz aplikacije MR in jih posredovati MoveIt-u [8]
- Načrtovati pot od znane pozicije do ciljne točke virtualnega objekta
- Upoštevanje v naprej znanih poti in položajev (vračanje v začetno pozicijo)
- Posredovanje rezultatov načrtovanja gibanja iz MoveIt-a [8] na aplikacijo MR.



Slika 3. Arhitektura sistema.

## 5 Zaključek in nadaljnje delo

Naša rešitev nakazuje preprost in intuitiven proces interakcije z robotom, kar lahko operaterju olajša delo z njim brez strokovnega znanja na področju robotike. Prednosti implementiranega sistema izhajajo iz narave tehnologije MR in osnovnih funkcionalnosti aplikacije. Obogatena resničnost lahko zagotovi vmesnike, ki se povezujejo z realnim svetom ter obogatijo fizične predmete in prostor z digitalnimi informacijami ter funkcijami, zaradi česar je njihova interpretacija in uporaba trivialna. Prikaz želenega položaja objektov in vizualizacija gibanja robota delata to metodo hitrejšo in lažjo. Takšna metoda lahko aktivno prispeva k rekonfiguraciji proizvodnega sistema z zmanjšanjem skupnega časa, potrebnega za prilagoditev proizvodnega mesta.

Omeniti je potrebno nekatere omejitve predlagane rešitve. Pomankljivost sistema se kaže v robustnosti in nenatančnosti, zato to orodje ne more izvajati naloge, ki zahtevajo višjo natančnost (< 50 mm).

Prav tako bi lahko v naš sistem dodali dodatne virtualne elemente, ki bi uporabniku še izboljšale izkušnjo in sistem naredile zmogljiveši.

Nadaljnje raziskave se bodo osredotočile na izvedbo okolja, ki bo uporabniku omogočil celoten postopek učenja robota z demonstracijo. Torej dodajanje posameznih generiranih trajektorij v program, določitev posameznih poti in premikov objektov ter generiranje celotnega robotskega programa s poljubno postavitvijo virtualnih objektov. Sistem želimo obogatiti z zaznavanjem kolizij z že postavljenimi objekti in načrtovanje trajektorije z izogibom objektom v realnem prostoru. Trenutna zmogljivost aplikacije je izvedba robotske naloge pobiranja in odlaganja objektov, v nadaljevanju bi zmogljivost lahko razširili do te mere, da bi s sistemom izvajali nalogo sestavljanja, ki zahteva vodenje robota po sili, s tem pa bi kompenzirali tudi nenatančnosti, ki se pojavijo. Prav tako bi vizualni del dopolnili z indeksi manipulabilnosti in dosegljivosti, kar sta ključna faktorja za hitro izvedbo robotske naloge. Za razširitev manipulacije z objekti bi dodali bazo objektov katere bi uporabili v postopku učenja in uporabe, prav tako bi objekti imeli poljubno začetno pozicijo in orientacijo v robotskem delovnem prostoru. Sistem bi lahko bil še bolj intuitiven v pripravi, da bi uporabnik manipuliral z realnimi objekti in s tem robota naučil želenih premikov, v takšen sistem pa bi bilo smiselno vključiti globinsko kamero za zaznavanje realnih objektov.

V aplikaciji niso bili temeljito upoštevani varnostni vidiki. Vizualizacija gibanja robota zagotavlja varnostni ukrep, vendar bi bilo potrebno ta vidik v prihodnje izboljšati. Vsekakor je končni cilj implementirati navedbe, ki bi ne le izboljšale uporabniško izkušnjo, ampak tudi zagotovile varnostne ukrepe, sistem pa naredile bolj funkcionalen in zmogljiv.

## Zahvala

Članek je nastal v sklopu magistrske naloge v okviru podiplomskega študija na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru. Pri nalogi je kot mentor sodeloval izr. prof. dr. Aleš Hace z Laboratorija za industrijsko robotiko. Zahvaljujem se za strokovno pomoč in vodenje pri izvedbi.

## Literatura

- [1] S.Stradovnik, R.Pučko, A.Hace, »Interaktivna laboratorijska aplikacija s kolaborativnim robotom in holografskim vmesnikom mešane resničnosti HoloLens na primeru vijačenja«, *Ventil*, str. 378-386, maj 2019
- [2] S.Stradovnik, R.Pučko, M.Golob, M.Divjak, D.Prelevič, A.Hace, »Fleksibilna kolaborativna robotska celica za površinsko obdelavo s hitrim intuitivnim učenjem in holografskim vmesnikom«, v *Zbornik dvanajste konference Avtomatizacija v industriji in gospodarstvu*, april 2021, Maribor, Slovenija
- [3] B.Gajšek, S.Stradovni, A.Hace. (Avgust 2020). »Sustainable move towards flexible, robotic, human-involving workplace«, *MDPI*, [Online]. Dosegljivo: <https://www.mdpi.com/2071-1050/12/16/6590/htm>. [20.6.2022]
- [4] S.Chung, N.Murray, Y.Qiao. (September 2020). »RoSTAR: ROS-based Telerobotic Control via Augmented Reality«, *ResearchGate*, [Online]. Dosegljivo: [https://www.researchgate.net/publication/345177207\\_RoSTAR\\_ROS-based\\_Telerobotic\\_Control\\_via\\_Augmented\\_Reality](https://www.researchgate.net/publication/345177207_RoSTAR_ROS-based_Telerobotic_Control_via_Augmented_Reality). [21.6.2022]
- [5] J.Aleotti, G.Micconi, S.Caselli. (Oktober 2015). »Object interaction and task programming by demonstration in visuo.haptic augmented reality«, *SpringerLink*, [Online]. Dosegljivo: <https://link.springer.com/article/10.1007/s00530-015-0488-z>. [21.6.2022]
- [6] *Vuforia*, [Online]. Dosegljivo: <https://developer.vuforia.com>. [21.6.2022]
- [7] *Unity*, [Online]. Dosegljivo: <https://unity.com>. [21.6.2022]
- [8] *ROS*, [Online]. Dosegljivo: <https://www.ros.org>. [21.6.2022]
- [9] *MRTK*, [Online]. Dosegljivo: <https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk2/?view=mrtkunity-2022-05>. [21.6.2022]
- [10] *Moveit*, [Online]. Dosegljivo: [http://docs.ros.org/en/melodic/api/moveit\\_tutorials/html/index.html](http://docs.ros.org/en/melodic/api/moveit_tutorials/html/index.html) [24.8.2022]
- [11] M.Rudorfer, J.Guhl, P.Hoffmann, J.Krüger (Oktober 2018). »Holo Pick'n'Place«, *IEEE Xplore*, [Online]. Dosegljivo: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8502527> [24.8.2022]