Upravljanje svesmjernim mobilnim manipulatorom

Kovač, Domagoj

Master's thesis / Diplomski rad

2017

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:761359

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-17**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Diplomski sveučilišni studij Računarstvo

UPRAVLJANJE SVESMJERNIM MOBILNIM MANIPULATOROM

Diplomski rad

Domagoj Kovač

Osijek, 2017.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

Obrazac D1: Obrazac za imenovanje Povjerenstva za obranu diplomskog rada

Osijek, 22.09.2017.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za obranu diplomskog rada

Ime i prezime studenta:	Domagoj Kovač
Studij, smjer:	Diplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	D 830 R, 09.10.2015.
OIB studenta:	09148476270
Mentor:	Doc.dr.sc. Ratko Grbić
Sumentor:	Prof.dr.sc. Robert Cupec
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	Doc.dr.sc. Emmanuel-Karlo Nyarko
Član Povjerenstva:	Prof.dr.sc. Robert Cupec
Naslov diplomskog rada:	Upravljanje svesmjernim mobilnim manipulatorom
Znanstvena grana rada:	**Procesno računarstvo (zn. polje računarstvo)**
Zadatak diplomskog rada:	Izraditi sustav upravljanja mobilnim robotskim manipulatorom zasnovanim na svesmjernoj mobilnoj platformi. (Sumentor: Robert Cupec)
Prijedlog ocjene pismenog dijela ispita (diplomskog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	22.09.2017.

Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis: Ratko Grbić
	Datum: 28.09.2017.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 27.09.2017.

Ime i prezime studenta:	Domagoj Kovač
Studij:	Diplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	D 830 R, 09.10.2015.
Ephorus podudaranje [%]:	1

Ovom izjavom izjavljujem da je rad pod nazivom: **Upravljanje svesmjernim mobilnim manipulatorom**

izrađen pod vodstvom mentora Doc.dr.sc. Ratko Grbić

i sumentora Prof.dr.sc. Robert Cupec

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

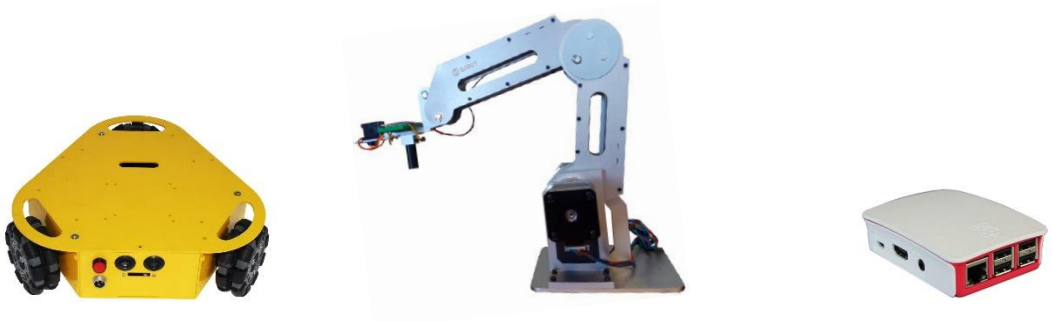
SADRŽAJ 2
1. UVOD 1
2. SVESMJERNI MOBILNI MANIPULATOR 2
2.1. Mobilna platforma 2
2.1.1. Kinematika svesmjerne mobilne platforme 3
2.2. Robotski manipulator 5
3. IZVEDBA SUSTAVA UPRAVLJANJA 7
3.1. Robotski operacijski sustav (ROS) 8
3.1.1. Arhitektura sustava upravljanja 9
3.2. Programska podrška mobilne platforme 11
3.2.1. Čvor daljinskog upravljanja 11
3.2.2. Čvor kinematike 13
3.2.3. Čvor mikroupravljača 15
3.3. Programska podrška robotskog manipulatora 16
3.3.1. Čvor daljinskog upravljanja 16
3.3.2. Čvor kinematike 17
4. RUKOVANJE SUSTAVOM UPRALJVANJA 18
4.1. Vrednovanje sustava upravljanja 19
4.1.1. Odašiljanje *WheelSpeed* poruke 20
4.1.2. Odašiljanje *DobotControl* poruke 21
5. ZAKLJUČAK 23

1. UVOD

Međudjelovanje robotskih manipulatora s okolinom ograničeno je njihovim radnim prostorom. Nemogućnost djelovanja izvan tog prostora podrazumijeva položaj objekata s kojima se rukuje u blizini manipulatora. Tako uređena okolina sužava primjenjivost robota, jer mogućnosti istog ovise o njegovu neposrednom okruženju. Opisano je ograničenje moguće otkloniti proširivanjem radnog prostora robotskog manipulatora primjenom mobilne platforme. Mobilni je manipulator sustav ostvaren ugradnjom robotskog manipulatora na mobilnu platformu. Takav je sustav pogodan za rad u neuređenim i dinamičnim okruženjima, zbog širokog radnog prostora i mogućnosti međudjelovanja s okolinom. Ovaj je rad usredotočen na izradu sustava upravljanja svesmjernog mobilnog manipulatora. Razmatrani sustav upravljanja mora osigurati mogućnost daljinskog upravljanja mobilnim manipulatorom te neovisnost o sredstvu upravljanja, što zahtijeva raspodijeljenost i modularnost sustava. Složen kinematički sustav podijeljen je na temeljne podsustave: mobilnu platformu i robotski manipulator. Tada ograničenja i mogućnosti pri razvoju sustava upravljanja ovise o odabranom sklopovlju neovisno o podsustavu. U drugom je poglavlju dan pregled korištenog sklopovlja i opisane su značajke oba podsustava. Za mobilnu je platformu raspisana kinematička analiza u svrhu razvoja programske podrške sustava upravljanja navedene unutar trećeg poglavlja. Treće poglavlje sadrži opis izvedbe raspodijeljenog sustava upravljanja mobilnim manipulatorom ostvarenog Robotskim operacijskim sustavom (ROS-om) i opisuje razvijenu programsku podršku. Pregled rada sustava upravljanja svesmjernog mobilnog manipulatora dan je unutar četvrtog poglavlja.

2. SVESMJERNI MOBILNI MANIPULATOR

Sustav se upravljanja razvija za mobilni manipulator sastavljen od svesmjerne mobilne platforme s tri neovisno pogonjena istosmjerna motora i robotskog manipulatora s četiri stupnja slobode. Odabrano se sklopovlje sastoji od platforme tvrtke *Nexus Robot* i manipulatora *Dobot Arm*. Navedeni se podsustavi mobilnog manipulatora povezuju središnjom upravljačkom jedinicom ostvarenom ugradbenim računalom *Raspberry Pi* (Sl. 2.1.). Podsustavi su povezani preko USB-a.



a) b) c)

Sl. 2.1. *Mobilni manipulator čine: a) Nexus Robot svesmjerna mobilna platforma, b) robotski manipulator Dobot Arm i c) ugradbeno računalo Raspberry Pi.*

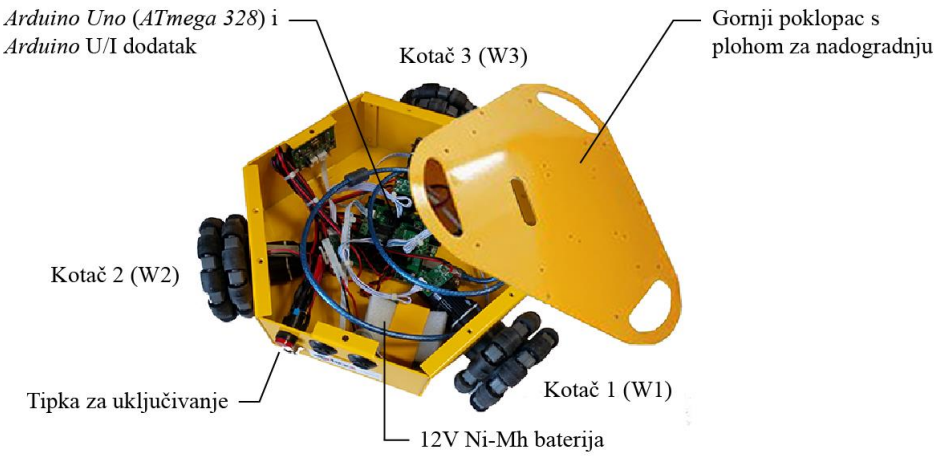
Mobilni se manipulator, za potrebe razvoja sustava upravljanja, razmatra kao sustav sastavljen od dvaju zasebnih robotskih podsustava. Tada ograničenja i mogućnosti pri razvoju sustava upravljanja ovise o odabranom sklopovlju i pripadnoj kinematici neovisno o podsustavu [1]. Korišteno sklopovlje opisano je u nastavku.

2.1. Mobilna platforma

Platforma s pogonom[1] na tri svesmjerna kotača može se gibati u proizvoljnom smjeru na plohi bez prethodnog okretanja. Sadrži mikroupravljač zasnovan na *Arduino Uno* mikroupravljaču i tri istosmjerna motora s optičkim enkoderima za pokretanje kotača (Sl. 2.2.). Povezivanje motora s enkoderima na mikroupravljač omogućeno je *Arduino* U/I dodatkom (P1.1.), jer mikroupravljač samostalno ne posjeduje dovoljan broj naponskih nožica. Također, platforma sadrži ultrazvučne i infracrvene senzore, no isti nisu korišteni u ovome radu. Komunikacija s računalom ostvarena je serijskim sučeljem SPI i USB-om. SPI je korišten za programiranje mikroupravljača, a USB se

[1] Za svesmjerni je pogon s tri motora u literaturi moguće pronaći naziv „Kiwi“ pogon [2].

koristi za preostalu komunikaciju. Raspored nožica mikroupravljačkog sustava nalazi se u prilogu (P1.2.). Mobilnu je platformu moguće proširiti nadogradnjama na gornjoj plohi [3].



Sl. 2.2. *Svesmjerna mobilna platforma s označenim dijelovima.*

2.1.1. Kinematika svesmjerne mobilne platforme

Svesmjerni kotači sadrže, uzduž vlastita opsega, niz manjih valjaka koji su okomiti na os rotacije kotača. Opisana struktura omogućuje klizanje kotača i zbog toga se koristi u holonomnim[2] pogonima. Promjenu je smjera gibanja tada moguće ostvariti bez kružnog gibanja cjelokupnog sustava. Za sustave bez mogućnosti bočnog kretanja navedeno svojstvo ne vrijedi, što povećava vrijeme obavljanja radnji koje zahtijevaju promjenu smjera gibanja [3].

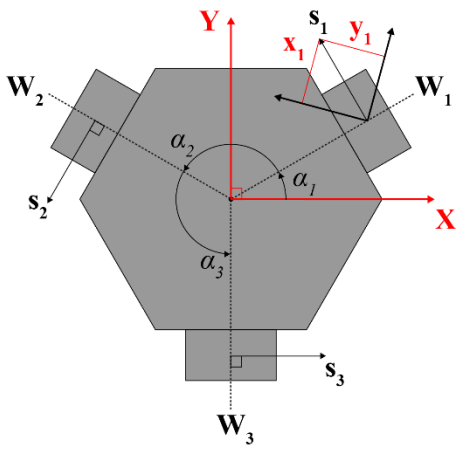
Ovisno o smjerovima i brzini okretanja kotača, moguće je odrediti smjer gibanja cijele platforme. Kinematika se platforme tada svodi na određivanje odnosa smjera gibanja platforme i brzina okretanja kotača s obzirom na koordinatni sustav platforme (Sl. 2.3.) [2]. Na slici 2.3. s_i predstavlja brzinu okretanja kotača. Isti je potrebno razložiti na pripadne **x** i **y** komponente koordinatnog sustava mobilne platforme u svrhu određivanja konačnog smjera gibanja. Prema [2], za svaki kotač vrijedi

$$\mathbf{x}_i = s_i cos(\alpha_i + \frac{\pi}{2}) \tag{2-1}$$

$$\mathbf{y}_i = s_i sin(\alpha_i + \frac{\pi}{2}) \tag{2-2}$$

[2] Sustav je holonoman kada mu je broj upravljivih stupnjeva slobode jednak ukupnom broju stupnjeva slobode [4].

gdje α_i označava otklon osovine kotača s *x*-osi koordinatnog sustava platforme. Osovine se nalaze pod kutevima: $\alpha_1 = \frac{\pi}{6}$, $\alpha_2 = \frac{5\pi}{6}$ i $\alpha_3 = \frac{9\pi}{6}$. Zbroj **x** i **y** komponenata svih kotača u (2-1) i (2-2) predstavlja vektor smjera gibanja mobilne platforme u cijelosti. Opisani vektor ovisi o brzini okretanja svakog kotača [2].



Sl. 2.3. *Shema platforme s pripadnim koordinatnim sustavom i vektorima gibanja kotača [2].*

Također, pomoću brzina okretanja kotača s_i moguće je opisati i kružno gibanje platforme. Okretanjem svih kotača jednakom brzinom u istom smjeru uzrokuje kružno gibanje platforme oko *z*-osi vlastita koordinatnog sustava [2]. Vrijedi

$$\boldsymbol{\theta} = \sum_{i=1}^{3} s_i \tag{2-3}$$

gdje $\boldsymbol{\theta}$ predstavlja vektor kružnog gibanja mobilne platforme. Zbog mogućnosti pokretanja platforme u proizvoljnom smjeru na plohi, bez prethodnog okretanja, izraz (2-3) nije neophodan za izvedbu razmatranog sustava upravljanja, ali uključen je u isti. Izraze (2-1), (2-2) i (2-3) moguće je sjediniti, tako da se matricom opišu ovisnosti vektora smjera gibanja platforme i brzina okretanja kotača (2-4). Vektor smjera gibanja mobilne platforme predstavljen je pravocrtnim vektorima **x** i **y** te kružnim vektorom $\boldsymbol{\theta}$ [2].

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \boldsymbol{\theta} \end{bmatrix} = \begin{bmatrix} cos(\alpha_1 + \frac{\pi}{2}) & cos(\alpha_2 + \frac{\pi}{2}) & cos(\alpha_3 + \frac{\pi}{2}) \\ sin(\alpha_1 + \frac{\pi}{2}) & sin(\alpha_2 + \frac{\pi}{2}) & sin(\alpha_3 + \frac{\pi}{2}) \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} \tag{2-4}$$

Gornji izraz pokazuje kako je množenjem brzina okretanja svesmjernih kotača i navedene matrice moguće odrediti pravac kojim će se platforma kretati uz zadane brzine. Opisani se odnos naziva izravnom ili direktnom kinematikom i zapisuje kraće kao

$$\vec{\mathbf{v}} = \mathbf{M}\vec{\mathbf{s}} \tag{2-5}$$

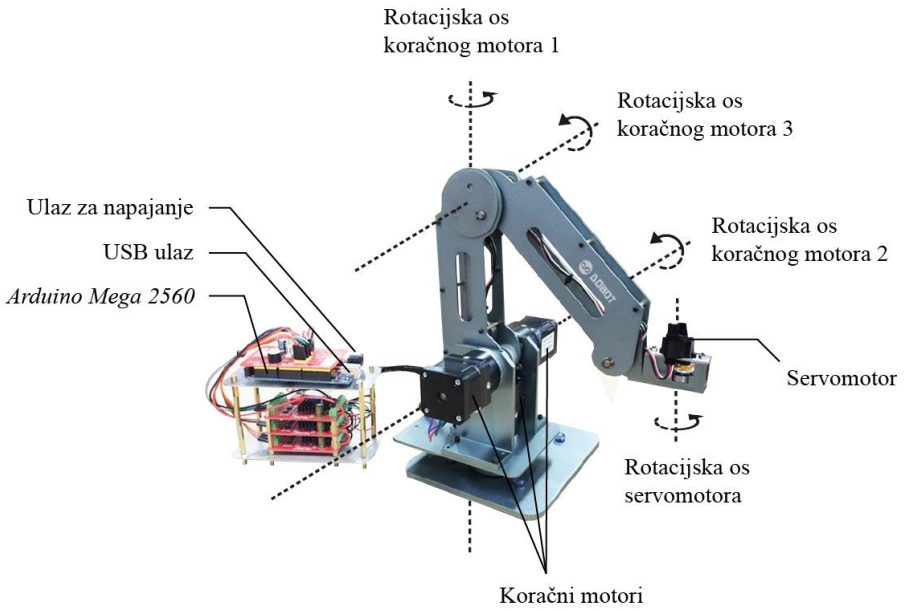
gdje $\vec{\mathbf{v}}$ označava vektor smjera gibanja platforme predstavljen s $[\mathbf{x}\ \mathbf{y}\ \boldsymbol{\theta}\ ]^T$, a $\mathbf{M}$ matricu odnosa vektora gibanja platforme i brzina okretanja kotača iz (2-4) [2].

Za potrebe razvoja sustava upravljanja mobilnim manipulatorom nužno je poznavanje željenog smjera gibanja platforme i tada valjanim izračunom moći odrediti odgovarajuće brzine okretanja kotača. Taj se postupak naziva inverznom kinematikom i provodi se množenjem (2-5) s inverzom matrice $\mathbf{M}$:

$$\vec{\mathbf{s}} = \mathbf{M}^{-1}\vec{\mathbf{v}} \tag{2-6}$$

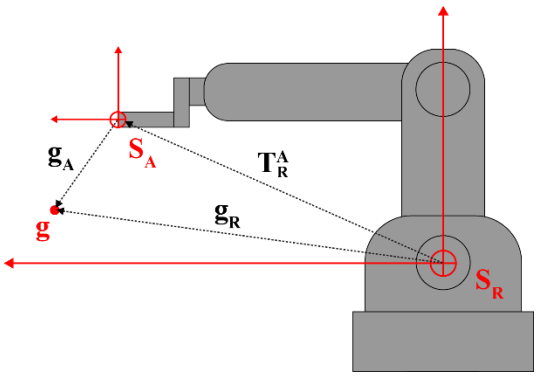
2.2. Robotski manipulator

Manipulator *Dobot Arm* posjeduje četiri stupnja slobode ostvarena s pomoću tri koračna i jednog servomotora. Koračni se motori koriste za pokretanje prvih triju zglobova, a servomotor za okretanje krajnjeg (Sl. 2.4.). Robotskim manipulatorom upravlja mikroupravljačka jedinica zasnovana na *Arduino Mega 2560* mikroupravljaču i FPGA-u. Komunikacija s računalom ostvarena je putem USB-a [5]. Tablica značajki manipulatora dana je u prilogu (P2).



Sl. 2.4. *Robotski manipulator Dobot Arm V1.0 s označenim dijelovima [6].*

Za razliku od mobilne platforme, za koju je bilo nužno provesti kinematičku analizu u svrhu razvoja programske podrške, za odabrani robotski manipulator to nije potrebno. Naime, *Dobot Arm* dolazi s razvijenom programskom podrškom koja uključuje kinematiku robotskog manipulatora. Programska podrška omogućuje postavljanje vrha alata manipulatora u zadanu točku s obzirom na referentni koordinatni sustav manipulatora S_R (Sl. 2.5.) [7].



Sl. 2.5. *Shema manipulatora s pripadnim koordinatnim sustavima i ciljnom točkom.*

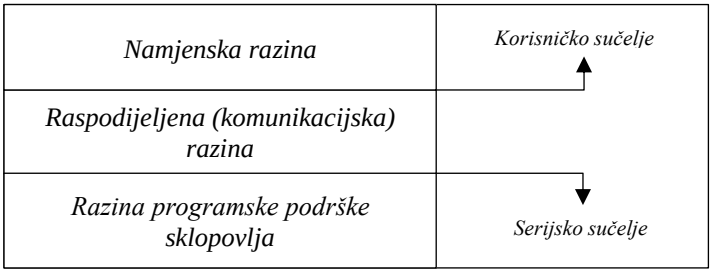
Neka je $\mathbf{T_R^A}$ transformacijska matrica koordinatnog sustava S_R u S_A, a $\mathbf{g_R}$ vektor udaljenosti ciljne točke od ishodišta S_R. Tada izraz (2-7) daje vektor udaljenosti ciljne točke od koordinatnog sustava vrha alata manipulatora [7]. Provedba navedenog izraza u programskoj podršci razmatranog sustava upravljanja omogućuje postavljanje vrha alata manipulatora u korisnički zadanu točku s obzirom na referentni koordinatni sustav robotskog manipulatora.

$$\mathbf{g_A} = \mathbf{T_R^A} \cdot \mathbf{g_R} \tag{2-7}$$

3. IZVEDBA SUSTAVA UPRAVLJANJA

Prethodno izvedbi sustava upravljanja potrebno je ustanoviti zahtjeve koje isti mora ispunjavati. Mobilnim je manipulatorom nužno moći upravljati na daljinu, jer daljinsko upravljanje (engl. *teleoperation*) omogućuje migraciju središnje upravljačke jedinice s robotskog sklopovlja na udaljeno računalo. Time je osigurana potrebna računska moć za pokretanje algoritama za međudjelovanje robota u dinamičnom okruženju i ostvareno pristupačnije sučelje korisnika prema robotskom sustavu. Sučelje mora omogućiti korisniku zasebno upravljanje podsustavima mobilnog manipulatora i osigurati neovisnost o sredstvu upravljanja (tipkovnica, igraća palica, upravljačka skripta i slično). Za mobilnu je platformu nužno moći odrediti smjer i brzinu kretanja, dok je za manipulator potrebno omogućiti postavljanje vrha alata u korisnički zadanu točku s obzirom na njegov referentni koordinatni sustav. Navedeni se zahtjevi ispunjavaju izvedbom raspodijeljenog sustava upravljanja, što znači da su procesi unutar robotskog sustava određeni jedinstvenim programskim cjelinama koje obrađuju i međusobno razmjenjuju odgovarajuće podatke.

Razmatrani sustav upravljanja mobilnim manipulatorom oslanja se na postojeću programsku podršku niže razine mobilne platforme i robotskog manipulatora. Naime, za oba podsustava postoje pogonski programi (engl. *drivers*) i koriste se u izradi programske podrške više razine za izvedbu sustava upravljanja. Pogonski program i pripadne biblioteke robotskog manipulatora iz [7] izmijenjeni su za potrebe istraživanja [5], dok je programska podrška niže razine za mobilnu platformu, uz uputstva za korištenje, dostupna na mrežnim stranicama proizvođača [3]. Programska podrška razmatranog sustava upravljanja sastoji se od tri razine (Sl. 3.1.).



Sl. 3.1. *Razine sustava upravljanja mobilnim manipulatorom.*

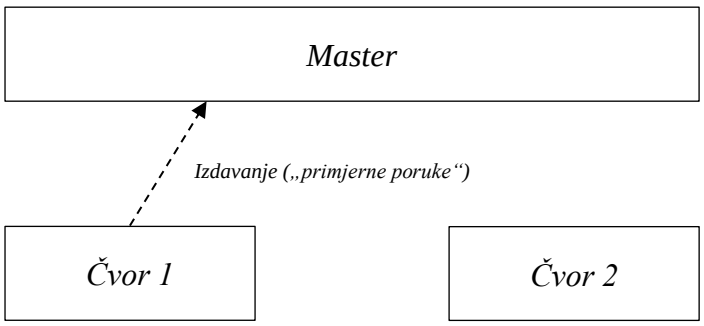
Razina programske podrške sklopovlja zadužena je za pogon mobilne platforme i robotskog manipulatora. Sadrži programsku podršku jedinstvenu za svaki podsustav: pogonske programe i pripadne biblioteke. Raspodijeljenom (ili komunikacijskom) razinom podsustavi se objedinjuju u jedan sustav s pomoću Robotskog operacijskog sustava (engl. *Robotic Operating System*).

Namjenska je razina sučelje korisnika i sustava upravljanja mobilnog manipulatora, gdje korisnik može biti čovjek, ili upravljačka skripta.

3.1. Robotski operacijski sustav (ROS)

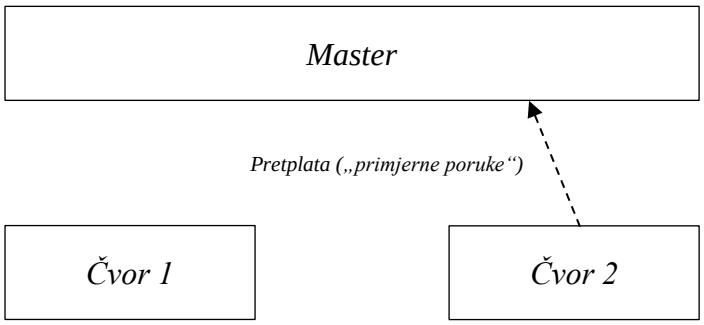
Robotski operacijski sustav skup je biblioteka i alata za pisanje programske podrške namijenjene robotima. Cilj je ROS-a pojednostaviti izradu složenih robotskih sustava na različitim sklopovljima [8]. Za potrebe sustava upravljanja mobilnog manipulatora razmatranog ovim radom koristit će se *Kinetic Kame* inačica ROS-a sa *Xenial* izdanjem *Ubuntu* operacijskog sustava na prijenosnom računalu i *Mate* izdanjem na *Raspberry Piju*. U nastavku će biti opisani isključivo dijelovi ROS-a nužni za shvaćanje rada sustava upravljanja.

Komunikacija u ROS-u, korištena za potrebe ovoga rada, sastoji se od sustava raspodijeljenih programskih cjelina: čvorova. Čvorovi ROS-u zadaju procese kojima se izvršava jedinstvena funkcija unutar sustava upravljanja robota. Uobičajeno je da se sustavi upravljanja zasnovani ROS-om sastoje od više čvorova. Primjerice, jedan čvor upravlja motorima, drugi izvodi lokalizaciju i treći planira putanju robota. Čvorovima je ostvarena modularnost sustava i otpornost na greške, jer su ispadi vezani za pojedini čvor [8]. Svaki čvor može biti izdavač (engl. *publisher*), pretplatnik (engl. *subscriber*), ili oboje. Odnos izdavač/pretplatnik svodi se na rukovanje porukama, što uključuje njihovo primanje, obradu i prosljeđivanje dalje u sustav [8]. Moguća je i izravna komunikacija između čvorova (bez posredovanja poruka) s pomoću usluga (engl. *services*), no takav oblik komunikacije nije obuhvaćen ovim radom. Rukovanje porukama omogućuje nadređeni čvor ROS *Master* i taj se čvor pokreće prvi. Objavljene poruke šalju se na odgovarajuću sabirnicu s koje je iste moguće i preuzeti. Sabirnice se za poruke u ROS-u nazivaju temama (engl. *topics*). Slikama u nastavku prikazan je odnos izdavač-pretplatnik i posredovanje nadređenog čvora na jednostavnom primjeru komunikacije dvaju čvorova. Ponajprije, nadređenom se čvoru najavljuje izdavanje poruke na temu „primjerne poruke“ (Sl. 3.2.) [8].



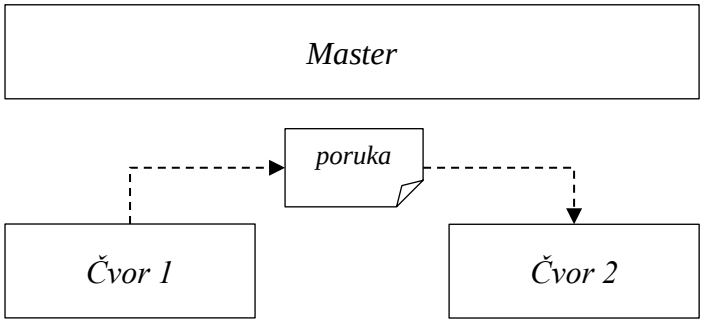
Sl. 3.2. *Čvor 1 izdaje poruke na temu „primjerne poruke“ [8].*

Poruke su tada dostupne kroz spomenutu temu i drugi se čvor može pretplatiti na istu (Sl. 3.3.) [8].



Sl. 3.3. *Čvor 2 pretplaćuje se na temu „primjerne poruke“[8].*

Nakon što tema ima izdavača i pretplatnika, nadređeni čvor obavještava izdavača o postojanju pretplatnika, odnosno pretplatnika o postojanju izdavača te čvorovi tada kreću s međusobnom razmjenom poruka (Sl. 3.4.) [8].

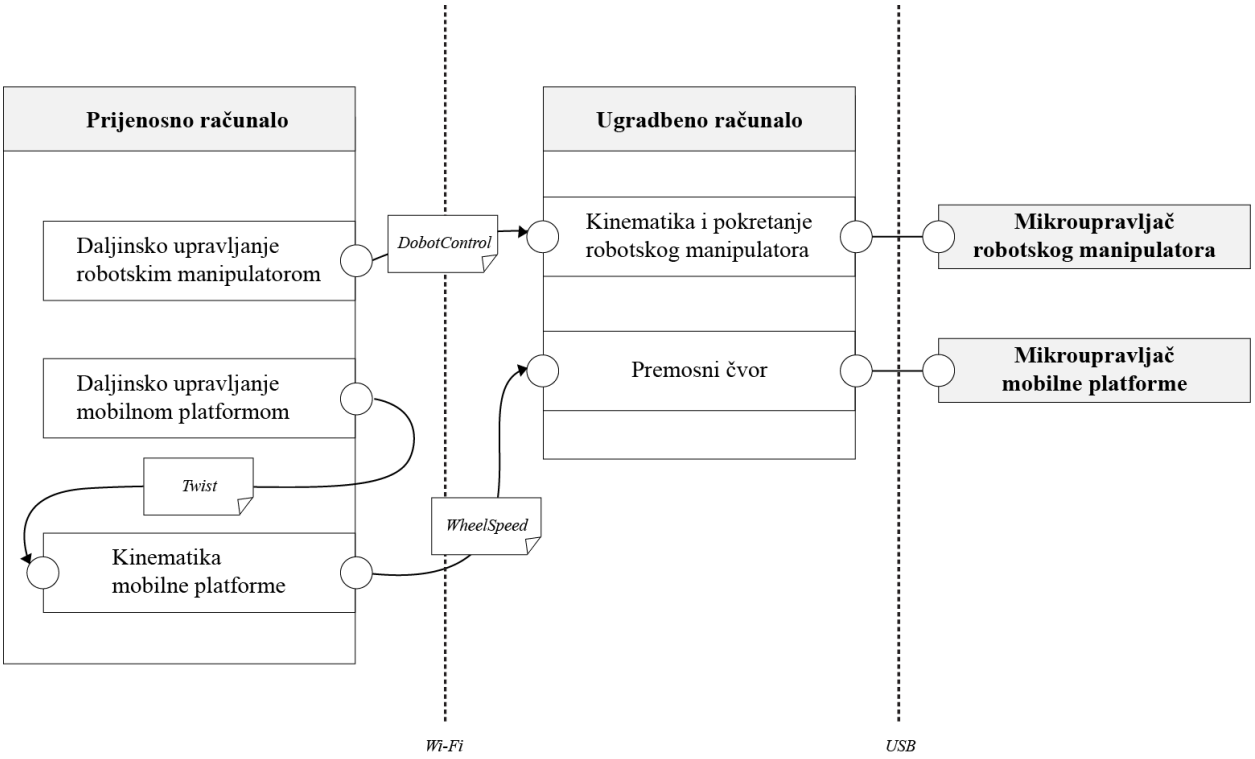


Sl. 3.4. *Čvor 1 šalje poruku čvoru 2 [7].*

3.1.1. Arhitektura sustava upravljanja

Slika 3.5. prikazuje sustav upravljanja mobilnim manipulatorom kao raspodijeljeni sustav zasnovan na ROS-u. Sustav se upravljanja sastoji od pet ROS čvorova, od koji se dva pokreću na ugradbenome računalu i serijski komuniciraju s mikroupravljačima mobilnog manipulatora. Prijenosno je računalo sučelje korisnika i robota: sadrži čvorove za daljinsko upravljanje, koji naredbe s tipkovnice šalju na daljnju obradu, te čvor kinematike mobilne platforme. Čvor kinematike mobilne platforme zasnovan je na izrazima u odjeljku 2.1.1. i pretvara korisnički zadan smjer gibanja platforme u odgovarajuće brzine okretanja kotača. Prijenosno i ugradbeno računalo povezani su na istu bežičnu mrežu preko koje ostvaruju komunikaciju. Za razliku od mobilne platforme, čvor za kinematiku (i pokretanje) robotskog manipulatora nalazi se na ugradbenome

računalu. Razlog je izvedba pogonskih programa i pripadnih biblioteka [5, 7] koji izravno koriste serijsku vezu s manipulatorom za njegovo pokretanje.



Sl. 3.5. *Sustav upravljanja mobilnim manipulatorom.*

Osnovna jedinica organiziranja programske podrške u ROS-u je paket. Paketi sadrže čvorove, biblioteke, skupove podataka (engl. *datasets*), datoteke postavki (engl. *configuration files*) i sve drugo što pripada jednoj smislenoj cjelini [8]. Primjerice, sva programska podrška raspodijeljene razine mobilnog manipulatora sadržana je u jednom paketu imena *omnibot* (P3). Paket *omnibot* uključuje:

- Manifest paketa
- Upute za jezični prevoditelj (engl. *compiler*) i povezivač (engl. *linker*)
- Direktorij programskih skripti
- Direktorij ROS poruka

Više je paketa moguće sjediniti u jedan metapaket u svrhu modularnosti [8], ali za mobilni manipulator razmatran ovim radom to nije učinjeno, iako podsustavi platforma i manipulator jesu odvojivi. Sustav upravljanja mobilnim manipulatorom pisan je programskim jezikom *Python,* izuzet čvora mikroupravljača platforme, koji je *C++*. Uz navedene, ROS podržava i *Lua* programski jezik [8].

Na slici 3.5., između čvorova, navedeni su nazivi korištenih poruka. Poruke sadrže uobičajene tipove podataka, kao što su cjelobrojna vrijednost, vrijednost s pomičnim zarezom, *bool* i drugi. Također, moguće je izraditi strukture podataka i nizove [8]. Tablicom 3.1. navedene su poruke korištene u izvedbi sustava upravljanja.

Tab. 3.1. *Poruke sustava upravljanja mobilnog manipulatora.*

Poruka	**Sadržaj**	**Opis**
Twist.msg	*Vector3 linear* *Vector3 angular*	Poruka paketa *geometry_msgs*. Sadrži vektore za pravocrtno i kružno gibanje od tri *float64* varijable: *x*, *y* i *z*. Za slučaj korištene mobilne platforme porukom se prenose vrijednosti koje opisuju korisnički zadani vektor smjera gibanja platforme iz (2-6).
WheelSpeed.msg	*float64 s1* *float64 s2* *float64 s3*	Izrađena za potrebe sustava upravljanja mobilnim manipulatorom. Poruka nosi vrijednosti brzina okretanja kotača iz (2-6).
DobotControl.msg	*float64 x* *float64 y* *float64 z* *float64 speed*	Izrađena za potrebe sustava upravljanja mobilnim manipulatorom. Sadrži koordinate cilja u odnosu na referentni koordinatni sustav manipulatora. Također, sadrži i brzinu gibanja robotskog manipulatora.

3.2. Programska podrška mobilne platforme

Prema shemi prikazanoj slikom 3.5., programska se podrška mobilne platforme može podijeliti na dvije jedinstvene cjeline s obzirom na razine sustava upravljanja. Raspodijeljena se razina sastoji od čvorova koji obavljaju procese više razine s pomoću ROS-a i korisniku omogućuju upravljanje platformom. Raspodijeljenoj razini pripadaju čvorovi za daljinsko upravljanje (*omni_teleop*), kinematiku (*omni_kinematics*) te za serijsku komunikaciju s mikroupravljačem (*serial_node*). Razina programske podrške sklopovlja sadrži biblioteke nužne za pogon platforme preko mikroupravljača. Sadrži biblioteku za rad motora (*MotorWheel*), upravljanje PID regulatorom (*PID_Beta6*) te biblioteku više razine za upravljanje mobilnom platformom (*Omni3WD*). Datoteka *omni.ino* sadrži programski kod mikroupravljača, prilagođen za primanje poruka raspodijeljene razine. Ključni dijelovi spomenutih datoteka bit će pojašnjeni odjeljcima u nastavku.

3.2.1. Čvor daljinskog upravljanja

Daljinsko upravljanje (engl. *teleoperation*) mobilnom platformom izvedeno je datotekom *omni_teleop.py* koja predstavlja istoimeni čvor. Tipkovnica prijenosnog računala sučelje je korisnika i sustava upravljanja mobilnim manipulatorom pa tako i mobilnom platformom. Upravljanje je omogućeno preko numeričke tipkovnice (Tab. 3.2.). Prema izrazu (2-6), brzine

okretanja svih kotača moguće je dobiti množenjem pretvorbene matrice i vektora smjera gibanja platforme. Čvor daljinskog upravljanja zadužen je za određivanje vektora smjera gibanja iz tipki prijenosnoga računala i njihovo prosljeđivanje dalje u sustav upravljanja. Unutar datoteke čvor se postavlja donjom naredbom (Pr. 3.1.).

```
rospy.init_node('omni_teleop')
```

Pr. 3.1. *Postavljanje čvora imena „omni_teleop“ u Pythonu.*

Tab. 3.2. *Tipke prijenosnog računala i pripadni vektori gibanja platforme*.

Tipka	Smjer	x-os	y-os	z-os
1	Lijevo-nazad	-1	-1	0
2	Nazad	0	-1	0
3	Desno-nazad	-1	-1	0
4	Lijevo	-1	0	0
6	Desno	1	0	0
7	Lijevo-naprijed	-1	1	0
8	Naprijed	0	1	0
9	Desno-naprijed	1	1	0
a	Okret-lijevo	0	0	1
d	Okret-desno	0	0	-1

Izvedba gornje tablice u *Pythonu* prikazana je primjerom u nastavku (Pr. 3.2.). Omogućeno je i povećanje, odnosno smanjenje brzine u iznosu od 10 % trenutne vrijednosti tipkama *w* i *s*.

```
moveBindings =  {'1':(-1,-1,0),
                 '2':(0,-1,0),
                 '3':(1,-1,0),
                 '4':(-1,0,0),
                 '6':(1,0,0),
                 '7':(-1,1,0),
                 '8':(0,1,0),
                 '9':(1,1,0),
                 'a':(0,0,1),
                 'd':(0,0,-1)}

speedBindings = {'w':(1.1,1.1),
                 's':(.9,.9)}
```

Pr. 3.2. *Povezivanje vektora smjera gibanja s odgovarajućim tipkama.*

Čvor dohvaća pritisnutu tipku funkcijom *getKey()* koja koristi biblioteke za raspoznavanje tipkovnice i sučeljavanje tipkovnice s terminalom *Ubuntu* operacijskog sustava (Pr. 3.3.). Provjera podudaranja tipki s onima iz tablice 3.2. izvodi se ako-onda grananjem (Pr. 3.4.).

```
def getKey():
    tty.setraw(sys.stdin.fileno())
    select.select([sys.stdin], [], [], 0)
    key = sys.stdin.read(1)
    termios.tcsetattr(sys.stdin, termios.TCSADRAIN, settings)
    return key
```

Pr. 3.3. *Funkcija za dohvaćanje pritisnute tipke u programskom jeziku Python.*

```
while(1):
      key = getKey()
      if key in moveBindings.keys():
            x = moveBindings[key][0]
            y = moveBindings[key][1]
            th = moveBindings[key][2]
      elif key in speedBindings.keys():
            speed = speed * speedBindings[key][0]
            turn = turn * speedBindings[key][1]
```

Pr. 3.4. *Provjera podudaranja pritisnute tipke s odgovarajućom tipkom.*

Otkrivanjem tipke iz tablice 3.2., odgovarajući se vektor postavlja i oblikuje u *Twist* poruku, radi slanja drugom čvoru sustava upravljanja (Pr. 3.5.). *Twist* se poruka sastoji od dva tročlana niza tipa *float64* koji predstavljaju komponente gibanja u prostoru s obzirom na određeni koordinatni sustav (Tab. 3.1.). Za mobilnu je platformu to pripadni koordinatni sustav iste. Gibanje u ravnini zahtjeva korištenje pravocrtnih komponenata u smjerovima *x* i *y*-osi koordinatnog sustava platforme, a omogućeno je i kružno gibanje platforme oko *z*-osi vlastitog koordinatnog sustava.

```
twist = Twist()
twist.linear.x = x*speed; twist.linear.y = y*speed; twist.linear.z = 0;
twist.angular.x = 0; twist.angular.y = 0; twist.angular.z = th*turn
pub.publish(twist)
```

Pr. 3.5. *Izrada i izdavanje Twist poruke s vektorom smjera gibanja, gdje su varijable speed i turn jedinične konstante, ako ne dolazi do promjene brzine pritiskom na tipke odgovarajuće tipke.*

Objekt *pub* iz gornjeg primjera predstavlja instancu izdavača čvora. Prije izdavanja poruke potrebno je odrediti temu na koju se poruka izdaje (Pr. 3.6.). U datoteci je nužno uključiti poruke paketa koje se želi koristiti. *Twist* je dio paketa *geometry_msgs* [8].

```
from geometry_msgs.msg import Twist
pub = rospy.Publisher('cmd_vel', Twist, queue_size = 1)
```

Pr. 3.6. *Postavljanje izdavača na temu cmd_vel s Twist porukom. Argument queue_size određuje broj pohranjenih poruka za slanje.*

3.2.2. Čvor kinematike

Ovim je čvorom ostvarena pretvorba vektora smjera gibanja mobilne platforme u brzine okretanja kotača prema izvodima iz odjeljka o kinematici mobilne platforme (Str. 3). Izraz (2-4) sadrži matricu pretvorbe brzina okretanja kotača u vektor gibanja platforme, a izraz (2-6) njen

inverz. U izvedbi čvora korištena je *Python* biblioteka za višedimenzionalan račun *NumPy*. Prema (2-4) i (2-5), izrađen je programski kod za dohvaćanje vrijednosti brzina okretanja svesmjernih kotača (Pr. 3.7.). Niz *wdata* sadrži podatke prenesene *Twist* porukom iz prethodno opisanog čvora daljinskog upravljanja.

```
x = wdata[0]
y = wdata[1]
th = wdata[2]
pi = 3.1415
alpha = [pi/6, (5*pi)/6, (9*pi)/6]

M = mat([ [cos(alpha[0]+(pi/2)),cos(alpha[1]+(pi/2)),cos(alpha[2]+(pi/2))],
          [sin(alpha[0]+(pi/2)),sin(alpha[1]+(pi/2)),sin(alpha[2]+(pi/2))],
          [1, 1, 1]])

M_inv = inv(M)
v = mat([[x],[y],[th]])
refSpeed = M_inv.dot(v)
```

Pr. 3.7. *Pretvorba vektora smjera gibanja, dobivenog Twist porukom, u brzine okretanja kotača.*

Kako bi se niz *wdata* mogao popuniti odgovarajućim podatcima, potrebno je postaviti pretplatnika na temu *cmd_vel* i poruku *Twist* (Pr. 3.8.). Prilikom postavljanja pretplatnika, istom se predaje funkcija povratnog poziva (engl. *callback function*) koja se izvršava prilikom pojave nove poruke na temi.

```
wdata = [0] * 4

def tfCallback(data):
    # Twist data
    wdata[0] = data.linear.x
    wdata[1] = data.linear.y
    wdata[2] = data.angular.z

rospy.Subscriber('cmd_vel', Twist, tfCallback)
```

Pr.3.8. *Pretplatnik na cmd_vel s pripadnom funkcijom povratnog poziva.*

Vrijednosti dobivene pretvorbom iz primjera 3.7. oblikuju *WheelSpeed* poruku i šalju se čvoru mikroupravljača mobilne platforme na daljnju obradu i pogon motora. Slanje je moguće nakon postavljanja izdavača na temu *w_speed* koja sadrži *WheelSpeed* poruku. Ime teme određuje se unutar programskog koda, dok je ime poruke određeno imenom datoteke koja sadrži podatkovnu strukturu poruke (Tab. 3.1.) unutar ROS paketa [8].

```
from omnibot.msg import WheelSpeed
pub = rospy.Publisher('w_speed', WheelSpeed, queue_size=1)
```

Pr. 3.9. *Postavljanje izdavača na temu w_speed s WheelSpeed porukom.*

3.2.3. Čvor mikroupravljača

Unutar čvora koristi se paket *rosserial* zbog potrebe za sudjelovanjem mikroupravljača mobilne platforme u komunikaciji unutar sustava upravljanja. Kako bi se premostila veza serijske komunikacije ugradbenog računala i mikroupravljača s ostatkom sustava upravljanja, *rosserial* pokreće čvor na ugradbenom računalu, koji upravlja porukama namijenjenima mikroupravljaču. Postavljanje *rosserial* paketa na ugradbeno računalo opisano je u [9], a pokretanje premosnog čvora prikazano je primjerom 4.4. unutar poglavlja o rukovanju sustavom upravljanja. Na *Raspberry Piju* nalazi se postavljena inačica ROS-a *Kinetic Kame* kao i nužne biblioteke prema [9].

Programski je kod na mikroupravljaču nužno oblikovati u ROS čvor kako bi se omogućilo primanje premošćenih poruka. Naime, mikroupravljač je zadužen za postavljanje brzina okretanja kotača i pokretanje motora. Dakle, veza između mikroupravljača i čvora kinematike mobilne platforme mora biti uspostavljena kako bi isti pokretao motore s brzinama dobivenim pretvorbom korisnički zadanog vektora smjera gibanja platforme. Unutar *omni.ino* datoteke uključuju se *ros.h* i *WheelSpeed.h* zaglavne datoteke (Pr. 3.10.). Zaglavna datoteka *WheelSpeed.h* nastaje nakon provedenog postupka postavljanja *rosserial* paketa u [9]. Postupak je nužan kako bi se pojedine ROS biblioteke prilagodile za korištenje s *Arduino* mikroupravljačem.

```
#include <ros.h>
#include <omnibot/WheelSpeed.h>
```

Pr. 3.10. *Potrebne biblioteke za korištenje ROS-a unutar Arduino razvojnog okruženja.*

Nakon uključivanja zaglavnih datoteka moguće je koristiti ROS naredbe unutar *Arduino* okruženja. Potrebno je postaviti pretplatnika radi primanja poruka s brzinama okretanja kotača na mikroupravljač (Pr. 3.11.). Za razliku od prijašnjih, ovaj je čvor pisan u *C++* programskom jeziku kako bi se podudarao s programskim jezikom *Arduino* razvojnog okruženja.

```
float wheelSpeedMMPS[] = {0.,0.,0.};
void callback(const omnibot::WheelSpeed& data)
{
  wheelSpeedMMPS[0] = data.s1;
  wheelSpeedMMPS[1] = data.s2;
  wheelSpeedMMPS[2] = data.s3;
}
ros::Subscriber<omnibot::WheelSpeed> sub("w_speed", &callback);
```

Pr. 3.11. *Postavljanje izdavača na temu w_speed s WheelSpeed porukom.*

Postavljanje brzina motora provodi se funkcijama programske podrške mobilne platforme, koje primaju podatak o brzini okretanja kotača izražene u milimetrima u sekundi, kao i smjer okretanja [3]. Funkcijama se predaju vrijednosti *WheelSpeed* poruke uz odgovarajuće pojačanje (Pr. 3.12.).

```
if (wheelSpeedMMPS[1] < 0.)
   Omni.wheelLeftSetSpeedMMPS((int)(abs(wheelSpeedMMPS[1]*300)),DIR_ADVANCE);
else
   Omni.wheelLeftSetSpeedMMPS((int)(abs(wheelSpeedMMPS[1]*300)),DIR_BACKOFF);
```

Pr. 3.12. *Postavljanje brzine motora za lijevi kotač.*

3.3. Programska podrška robotskog manipulatora

Slika 3.5. prikazuje sastavnice programske podrške robotskog manipulatora. Istu je moguće podijeliti na dvije jedinstvene cjeline s obzirom na razine sustava upravljanja. Raspodijeljena se razina sastoji od ROS čvorova koji korisniku omogućuju upravljanje manipulatorom. Pripadaju joj čvorovi za daljinsko upravljanje (*dobot_teleop*) i kinematiku uz serijsko slanje podataka na mikroupravljač putem USB-a (*dobot_kinematics*). Čvor za daljinsko upravljanje nalazi se na prijenosnome računalu, a čvor za kinematiku i pokretanje manipulatora na ugradbenome. Razina programske podrške sklopovlja sadrži *open-dobot* biblioteke [7] za pogon manipulatora, uz prilagodbu i dodatne datoteke za potrebe istraživanja u [5]. Razina se programske podrške sklopovlja u cijelosti nalazi na ugradbenome računalu.

3.3.1. Čvor daljinskog upravljanja

Daljinsko upravljanje robotskim manipulatorom izvedeno je datotekom *dobot_teleop.py* koja predstavlja istoimeni čvor. Omogućeno je zadavanje koordinata ciljne točke vrha alata s obzirom na referentni koordinatni sustav manipulatora (Sl. 2.3.). Čvoru se ciljna točka zadaje tipkovnicom kao niz znakova iz kojeg je potrebno izlučiti koordinate cilja (Pr. 3.13). Raščlamba niza izvodi se regularnim izrazima *Python* biblioteke *re*.

```
def getGoalCoordinates():
    s = raw_input()
    data = [float(num) for num in re.findall(r'[+-]?\d+(?:\.\d+)?',s)]
    return data
```

Pr. 3.13. *Funkcija za dohvaćanje koordinata ciljne točke zadane ulaznim nizom znakova preko tipkovnice.*

Poput sukladnog mu čvora mobilne platforme, čvor daljinskog upravljanja robotskog manipulatora sadrži funkciju *getKey()* za dohvaćanje pritisnute tipke na tipkovnici (Pr. 3.3.). Navedena se funkcija koristi za upravljanje rada čvora. Omogućuje prekid rada čvora kombinacijom tipki *Ctrl* + *c* te postavljanje robotskog manipulatora u početni položaj tipkom *i* (Pr. 3.14.). Koordinate i

brzina manipulatora pohranjuju se u poruku *DobotControl* i objavljuju temom *cmd_xyz* istovjetnim postupkom kao kod mobilne platforme (Pr. 3.5. i Pr. 3.6.).

```
key = getKey()
if (key == '\x03'):
    break
elif (key == 'i'):
    x = 210.9
    y = 0.0
    z = 238.0
    speed = 40.0
```

Pr. 3.13. *Funkcija za dohvaćanje koordinata ciljne točke zadane ulaznim nizom znakova preko tipkovnice.*

3.3.2. Čvor kinematike

Programski kod unutar datoteka *DobotDriver.py*, *DobotSDK.py*, *DobotKinematics.py* i *Robot.py*, koja uključuje navedene, omogućuje pokretanje robotskog manipulatora *Dobot Arm* [5, 7]. S pomoću navedenih datoteka izrađen je ROS čvor imena *dobot_kinematics* za postavljanje vrha alata u zadanu ciljnu točku. Koordinate točke sadržane su u poruci *DobotControl* unutar teme *cmd_xyz* pa je potreban pretplatnik radi dohvaćanja sadržaja poruke (Pr. 3.14.). Funkciji *Move()* predaju se koordinate ciljne točke i brzine postavljanja vrha alata manipulatora u istu (Pr. 3.15.).

```
def dbtCallback(data):
    controlData[0] = data.x
    controlData[1] = data.y
    controlData[2] = data.z
    controlData[3] = data.speed
rospy.Subscriber('cmd_xyz', DobotControl, dbtCallback)
```

Pr. 3.14. *Pretplatnik na cmd_xyz s pripadnom funkcijom povratnog poziva.*

```
dobot.Move(controlData[0], controlData[1], controlData[2], controlData[3])
```

Pr. 3.15. *Postavljanje vrha alata manipulatora u točku određenu trima početnim argumentima funkcije uz brzinu određenu četvrtim argumentom.*

Prethodno pozivu gornje funkcije potrebno je izraditi objekt klase *Robot* iz *Robot.py*, otvoriti serijski priključak (engl. *serial port*) za komunikaciju s mikroupravljačem te postaviti manipulator u početni položaj funkcijom *MoveToInitial()* (Pr. 3.16.). Početni je položaj manipulatora određen u odnosu na referentni koordinatni sustav manipulatora (vrijednosti pripadnih koordinata prikazane su primjerom 3.13). Zatvaranje serijskog priključka izvodi se funkcijom *Close()* [5].

```
dobot = Robot()
dobot.Open()
dobot.MoveToInitial()
# Control arm with Move()
dobot.Close()
```

Pr. 3.16. *Niz naredbi potreban za valjano postavljanje upravljanja robotskim manipulatorom.*

4. RUKOVANJE SUSTAVOM UPRALJVANJA

Sustav upravljanja mobilnim manipulatorom pokreće se nizom *Ubuntu* i ROS naredbi. Ponajprije, potrebno je uspostaviti komunikaciju prijenosnog i ugradbenog računala. To je ostvareno putem SSH veze (engl. *Secure Shell*). SSH je alat za zaštićenu prijavu u udaljeni sustav preko nezaštićene mreže kriptografskim metodama [10]. Veza prijenosnog i ugradbenog računala zaštićena je asimetričnom metodom s javnim i privatnim ključem. Povezivanje na ugradbeno računalo izvodi se naredbom u nastavku (Pr. 4.1.).

```
master@ubuntu:~$ ssh pi@raspberrypi
```

Pr. 4.1. *Povezivanje korisnika prijenosnog računala na ugradbeno računalo Raspberry Pi.*

Nakon uspješnog povezivanja potrebno je postaviti ROS varijable ROS_IP i ROS_MASTER_URI, kako bi čvorovi bili vidljivi jedni drugima preko zajedničke mreže. Na ugradbenom je računalo potrebno postaviti obje varijable (Pr. 4.2.), dok je na prijenosnom potrebno postaviti samo pripadni ROS_IP (Pr. 4.3.). Postavljene su vrijednosti IP adrese računala [8].

```
master@ubuntu:~$ hostname -I
192.168.7.116
master@ubuntu:~$ export ROS_IP=192.168.7.116
```

Pr. 4.2. *Postavljanje ROS_IP varijable za prijenosno računalo.*

```
pi@raspberrypi:~$ hostname -I
192.168.7.43
pi@raspberrypi:~$ export ROS_IP=192.168.7.43
pi@raspberrypi:~$ export ROS_MASTER_URI=http://192.168.7.116:11311
```

Pr. 4.3. *Postavljanje ROS_IP i ROS_MASTER_URI varijable za ugradbeno računalo.*

Po završetku postavljanja navedenih ROS varijabli moguće je započeti s pokretanjem sustava upravljanja. Prvi čvor koji je nužno pokrenuti je nadređeni čvor ROS *Master*. Pokreće se naredbom *roscore* (Pr. 4.4.). Sustav se upravljanja sastoji od pet čvorova, od koji se dva pokreću na ugradbenome računalu i serijski komuniciraju s mikroupravljačima mobilnog manipulatora (Pr. 4.5. i pr. 4.6.). Potrebno je naglasiti kako se svi podređeni čvorovi pokreću u zasebnim prozorima koristeći sintaksu „naredba-paket-čvor“ kao što je prikazano primjerima u nastavku [8].

```
master@ubuntu:~$ roscore
```

Pr. 4.4. *Naredba za pokretanje nadređenoga čvora.*

```
pi@raspberrypi:~$ rosrun rosserial_python serial_node
```

Pr. 4.5. *Pokretanje rosserial čvora za komunikaciju ROS-a s Arduino mikroupravljačem mobline platforme [9].*

Kako bi se preko SSH veze otvorio novi prozor na ugradbenome računalu koristi se *screen* [11]. U novom se prozoru pokreće sljedeći čvor (Pr. 4.6.).

```
pi@raspberrypi:~$ rosrun omnibot dobot_kinematics
```

Pr. 4.6. *Pokretanje čvora kinematike za pogon mobilnog manipulatora.*

Nakon pokretanja gornjih čvorova, ostvarena je serijska komunikacija ugradbenog računala i mobilnog manipulatora. Nadalje, potrebno je pokrenuti čvorove na prijenosnome računalu kako bi se omogućilo međudjelovanje korisnika i mobilnog manipulatora posredovanjem sustava upravljanja (Pr. 4.7. – pr. 4.9.). Redoslijed pokretanja čvorova nije značajan, ali intuitivno je posljednje pokrenuti čvorove daljinskog upravljanja.

```
master@ubuntu:~$ rosrun omnibot omni_kinematics
```

Pr. 4.7. *Pokretanje čvora kinematike mobilne platforme.*

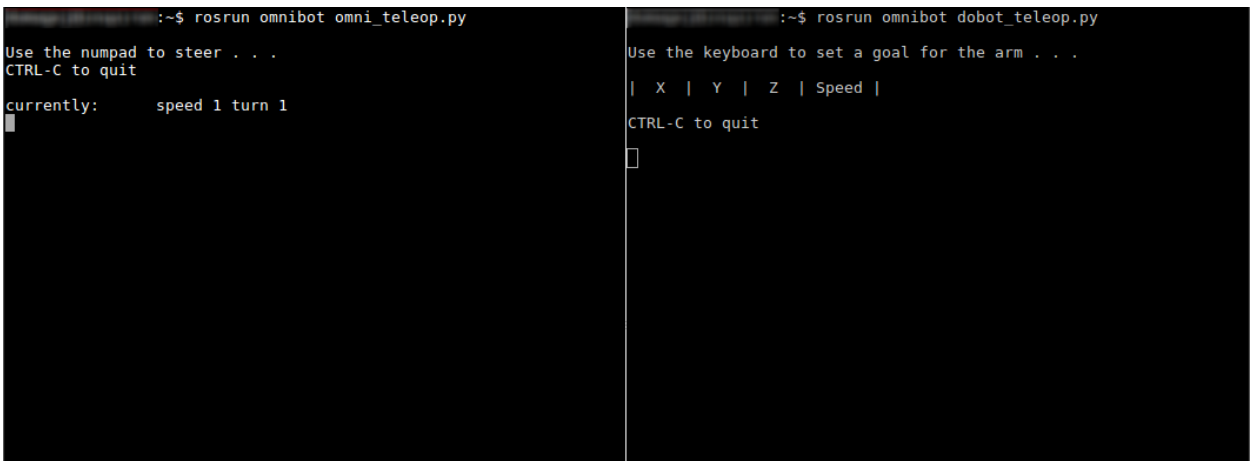
```
master@ubuntu:~$ rosrun omnibot omni_teleop
```

Pr. 4.8. *Pokretanje čvora daljinskog upravljanja mobilnom platformom.*

```
master@ubuntu:~$ rosrun omnibot dobot_teleop
```

Pr.4.9. *Pokretanje čvora daljinskog upravljanja robotskim manipulatorom.*

Pokretanjem je gornjih čvorova sustav upravljanja mobilnim manipulatorom postavljen i spreman za korištenje. Upravljanje platformom izvršava se unutar prozora *omni_teleop*, a manipulatorom unutar *dobot_teleop* prozora (Sl. 4.1.).



Sl. 4.1. *Korisničko sučelje za daljinsko upravljanje mobilnim manipulatorom.*

4.1. Vrednovanje sustava upravljanja

Provedena su mjerenja nad komunikacijom između čvorova sustava upravljanja mobilnog manipulatora. Mjerenja uključuju praćenje propusnosti (engl. *bandwidth*) i frekvencije odašiljanja

poruka, kao i promjene vrijednosti sadržaja poruka u vremenu, ako to dizajn čvora omogućuje. Zasebno se vrednuje svaki podsustav mobilnog manipulatora. Platforma se kreće u smjerovima omogućenim numeričkom tipkovnicom prema tablici 3.2., a manipulator vrh alata postavlja konstantnom brzinom s točke g_0 na g_1 za deset mjernih ponavljanja (Tab. 4.1.). Mjerenja su omogućena kroz grafički ROS alat *rqt* [8].

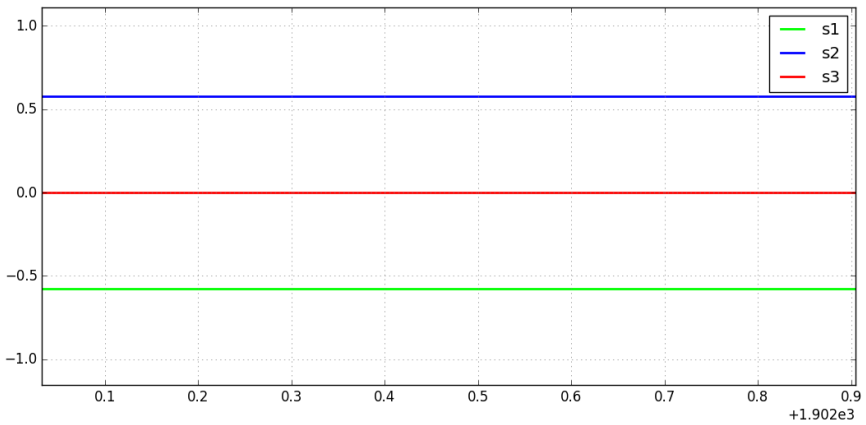
4.1.1. Odašiljanje *WheelSpeed* poruke

Dizajn čvora kinematike mobilne platforme uvjetuje odašiljanje poruka frekvencijom jednakom 10 Hz (Pr. 4.9.). Mikroupravljač platforme neprestano prima poruku sadržaja nula, dok god odgovarajuća tipka nije pritisnuta, što omogućuje iscrtavanje grafa odašiljanja *WheelSpeed* poruka u vremenu pri zadanoj frekvenciji. Prosječna propusnost prilikom odašiljanja ove poruke iznosi 240.17 bita u sekundi. Slikama u nastavku prikazan je odziv čvora kinematike mobilne platforme za gibanje unazad (pritiskom tipke 2 na numeričkoj tipkovnici). Pripadne grafove preostalih smjerova gibanja platforme moguće je pregledati unutar DVD priloga.

```
rate = rospy.Rate(10) #10Hz
```

Pr. 4.9. *Postavljanje frekvencije odašiljanja poruka na 10 Hz.*

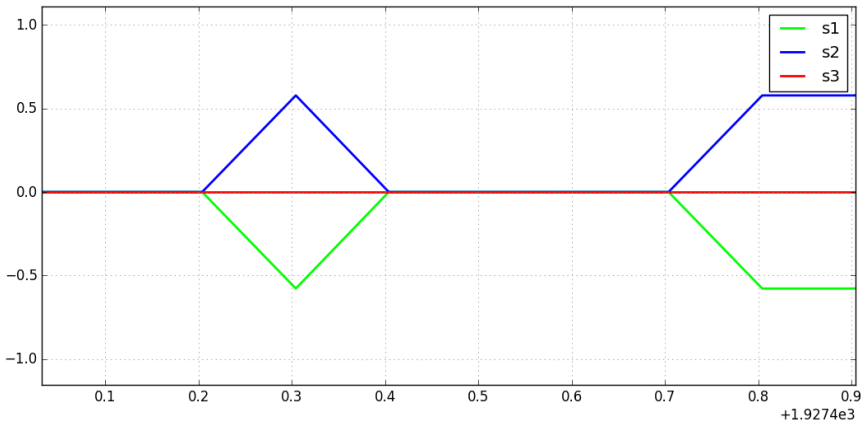
Prema slici 2.2., kretanje platforme unazad, odnosno unaprijed, omogućuje suprotno okretanje kotača *W1* i *W2* uz mirovanje kotača W3. Prikazom odziva čvora kinematike mobilne platforma vidljivo je podudaranje sadržaja poslane *WheelSpeed* poruke sa zadanim gibanjem platforme unazad (Sl. 4.2.).



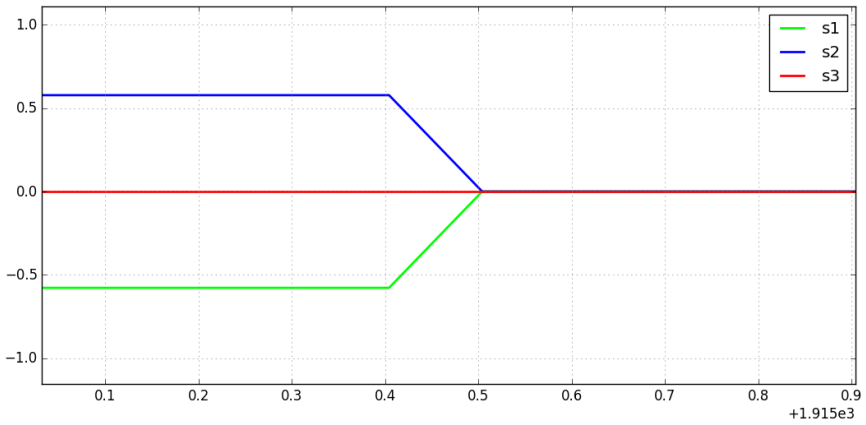
Sl. 4.2. *Odziv čvora kinematike mobilne platforme prilikom držanja tipke 2.*

Slika 4.2. prikazuje odziv nakon što pripadna tipka biva određeno vrijeme pritisnuta. Prilikom prvobitnog pritiska tipke, javlja se skok u odzivu koji uzrokuje neželjeni pokret platforme pri njenu pokretanju (Sl. 4.3.). Vidljivo je kako prilikom prvobitnog pritiska tipke odgovarajuća vrijednost

brzine jest postavljena, no sljedeći period odašiljanja vraća vrijednost na nulu. Vrijednost sadržaja *WheelSpeed* poruke ostaje na nuli sljedeća tri perioda odašiljanja, nakon kojih se postavlja na odgovarajuću vrijednost i ustaljuje do otpuštanja pritisnute tipke. Prilikom otpuštanja tipke ne dolazi do pojave skokova (Sl. 4.4.). Navedeno je ponašanje moguće uočiti upravljanjem platforme svim zadanim tipkama. Također, za svaku tipku je postojan trostruki period mirovanja između skoka i poželjnog odziva. Opisano ponašanje potrebno je otkloniti u svrhu poboljšanja pokretljivosti svesmjernog mobilnog manipulatora te za potrebe budućih radova temeljenih na razmatranom sustavu upravljanja. Moguće je rješenje problema odgađanje slanja poruka za zadani period odašiljanja, ali navedeno nije ispitano.



Sl. 4.3. *Odziv čvora kinematike mobilne platforme prilikom pritiska tipke 2.*



Sl. 4.4. *Odziv čvora kinematike mobilne platforme prilikom otpuštanja tipke 2.*

4.1.2. Odašiljanje *DobotControl* poruke

Tablica 4.1. prikazuje promjenu propusnosti prilikom odašiljanja poruka za niz ponavljanja predodređene radnje robotskog manipulatora. Ovim se mjerenjima pokazuje doprinos *DobotControl* poruke zasićenju komunikacije unutar sustava upravljanja. Razlikuju se propusnosti prije (BW_0) i prilikom odašiljanja (BW_1). Naime, vidljiva je promjena propusnosti nakon što se

poruka pojavi na temi, što je i očekivano, jer se odašilje podatkovna struktura s određenim brojem varijabli tipa *float64*. Prosječna promjena propusnosti odašiljanja *DobotControl* poruke iznosi 0.038 bita u sekundi. Frekvencija odašiljana približno je konstantna u iznosu od 0.03 Hz. Uspoređujući propusnost *DobotControl* i *WheelSpeed* poruke vidljivo je da daljinsko upravljanje robotskim manipulatorom ne zasićuje komunikacijski sustav kao upravljanje mobilnom platformom.

Tab. 4.1. *Propusnost i frekvencija slanja DobotControl poruke kroz deset ponavljanja, gdje BW_0 označava početnu propusnost, a BW_1 propusnost prilikom odašiljanja poruke.*

ROS tema: cmd_xyz *ROS poruka: DobotControl* *Brzina: 100 [mm/s]* *Pomak: $g_0(200, -200, 200) \rightarrow g_1(150, 200, 150)$*				
#	BW_0 [Bit/s]	BW_1 [Bit/s]	ΔBW [Bit/s]	f [Hz]
1	0.73	0.78	0.05	0.03
2	0.74	0.79	0.05	0.02
3	0.75	0.79	0.04	0.02
4	0.76	0.80	0.04	0.02
5	0.78	0.82	0.04	0.02
6	0.81	0.84	0.03	0.03
7	0.82	0.85	0.03	0.03
8	0.84	0.87	0.03	0.03
9	0.85	0.89	0.04	0.03
10	0.87	0.90	0.03	0.03

5. ZAKLJUČAK

Razvijen je sustav upravljanja za dani svesmjerni mobilni manipulator. Sustav se upravljanja temelji na Robotskom operacijskom sustavu (ROS) i omogućuje daljinsko upravljanje mobilnim manipulatorom koristeći računalo. U radu se zagovara raspodijeljenost i modularnost pa je mobilni manipulator promatran kao robotski sustav sastavljen od jedinstvenih podsustava bez razmatranja njihova međusobna odnosa. Takav pristup omogućuje zaseban razvoj programske podrške za svaki podsustav i pospješuje modularnost sustava upravljanja. Raspodijeljenost osigurava jednostavnost nadogradnje sustava i otpornost na ispad prilikom pojave izoliranih grešaka.

Podsustavi su povezani ugradbenim računalom *Raspberry Pi* čime je omogućena bežična veza s upravljačkim računalom korisnika. Daljinsko je upravljanje ostvareno putem tipkovnice korisnikova računala, ali dizajn sustava omogućuje izvedbu novih pristupa. Primjerice, moguće je razviti programsku podršku za upravljanje igraćom palicom, podlogom osjetljivom na dodir i slično, te istu izravno uključiti u postojeći sustav upravljanja.

Rukovanje sustavom upravljanja izravno je i jednostavno, no prvotno postavljanje sustava upravljanja u rad zahtijeva unos niza naredbi unutar upravljačkog sučelja *Ubuntu* operacijskog sustava. Svi se dijelovi sustava upravljanja moraju ručno unijeti, što zahtijeva poznavanje potrebnih naredbi i produžuje vrijeme upogonjenja mobilnog manipulatora. Navedeni je problem moguće riješiti izradom postavne UNIX skripte (*bash, sh* ili druge) koja će pokrenuti sve nužne dijelove sustava upravljanja.

Nadalje, uočena je pojava neželjenog ponašanja prilikom pokretanja platforme mobilnog manipulatora, što narušava okretnost razmatranog robotskog sustava. Uzrok pojave nije utvrđen unutar ovoga rada, no za potrebe daljnjeg razvijanja sustava upravljanja problem je potrebno riješiti.

LITERATURA

[1] G. Nikhade, V. Deshpande, S. Chiddarwar, Dynamic Interaction between Manipulator and Omni-Directional Platform, International Journal of Applied Research in Mechanical Engineering, No. 2, Vol. 3, 62-75, 2013., dostupno na: http://interscience.in/ijarme_vol3Iss2.html [19. srpnja 2017.].

[2] Robotics Projects, A Simple Introduction to Omni Roller Robots, 3. travnja 2017., dostupno na: http://modwg.co.uk/wp-content/uploads/2015/06/OmniRoller-Holonomic-Drive-Tutorial.pdf [23. svibnja 2017.].

[3] Nexus Automation Limited, 3WD Triangular 100 mm Omni Wheel Mobile Robotics Car 10003, dostupno na: http://www.nexusrobot.com/product/3wd-triangular-100mm-omni-wheel-mobile-robotics-car-10003.html [2. veljače 2017.].

[4] Robot Platform, Robot Locomotion: Holonomic and Non-Holonomic Drive, dostupno na: http://www.robotplatform.com/knowledge/Classification_of_Robots/Holonomic_and_Non-Holonomic_drive.html [2. kolovoza 2017.].

[5] P. Đurović, R. Grbić, R. Cupec, D. Filko, Low Cost Robot Arm with Visual Guided Positioning, MIPRO Proceedings / Biljanović, Petar (ur.), 1332-1337, Rijeka, 2017.

[6] K. Ellison, Future Tech: 5 Awesome Robot Arms For Your High Tech Home, GoExplore, 18. studenog 2015., dostupno na: http://www.goexplore.net/future-tech/robot-arms-the-5-best/ [11. rujna 2017.].

[7] maxosprojects, open-dobot, GitHub, dostupno na: https://github.com/maxosprojects/open-dobot [3. srpnja 2017.].

[8] ROS Documentation, Open Source Robotics Foundation, dostupno na: http://wiki.ros.org/ [20. veljače 2017.].

[9] M. Ferguson, P. Bouchier, M. Purvis, rosserial, Open Source Robotics Foundation, dostupno na: http://wiki.ros.org/rosserial [1. ožujka 2017.].

[10] D. Barrett, R. Silverman, R. Byrnes, SSH, the Secure Shell: The Definitive Guide, O'Reilly Media, 2005

[11] Screen, Cannonical Ltd, dostupno na: https://help.ubuntu.com/community/Screen [7. srpnja 2017.].

[12] Dobot User Manual, Shenzhen Yuejiang Technology Co., Ltd, dostupno na: http://www.dobot.cc/downloadcenter/dobot-arm-v1.html?sub_cat=101#sub-download [3. srpnja 2017.].

SAŽETAK

Potreba za međudjelovanjem robota današnjice unutar neuređenih i dinamičnih okruženja zahtijeva visok stupanj pokretljivosti, kao i sposobnosti rukovanja okolnim objektima. Mobilni manipulatori povezuju mogućnosti mobilnih platformi s onima robotskih manipulatora. Za pogon takvog sustava potreban je sustav upravljanja. Unutar ovoga rada razmatra se raspodijeljeni sustav upravljanja temeljen na ROS-u. Razmatrani sustav izvršava kinematičke izračune za mobilnu platformu i robotski manipulator te omogućuje daljinsko upravljanje robotskim sustavom.

Ključne riječi: robot, upravljanje, mobilni, manipulator, svesmjeran, ROS, raspodijeljen, čvor

OMNIDIRECTIONAL MOBILE MANIPULATOR CONTROL

ABSTRACT

The need for todays robots to interact with unstructured and dynamic environments requires superior mobility and manipulation capabilities in robotic systems. Mobile manipulators combine features of mobile platforms with those of robotic manipulators. A control system is needed to drive the joined system. Therefore, development of a ROS-based distributed control system is proposed. The proposed system handles kinematic computations of both the mobile platform and the robotic manipulator, and allows for remote system control.

Keywords: robot, control, mobile, manipulator, omnidirectional, ROS, distributed, node

ŽIVOTOPIS

Domagoj Kovač rođen je 4. rujna 1993. godine u Osijeku, Hrvatska. Od 2000. do 2008. godine pohađa osnovnu školu „Grigor Vitez“ u Osijeku. Srednjoškolsko obrazovanje započinje 2008. godine upisom gimnazijskog programa I. gimnazije Osijek. Istu završava 2012. godine polaganjem državne mature. Preddiplomski sveučilišni studij Računarstvo, na tadašnjem Elektrotehničkom fakultetu Osijek Sveučilišta Josipa Jurja Strossmayera u Osijeku, upisuje 2012. godine. Po završetku preddiplomskog studija, obrazovanje nastavlja na diplomskom studiju Računarstvo, blok Procesno računarstvo Fakulteta elektrotehnike, računarstva i informacijskih tehnologija Osijek.

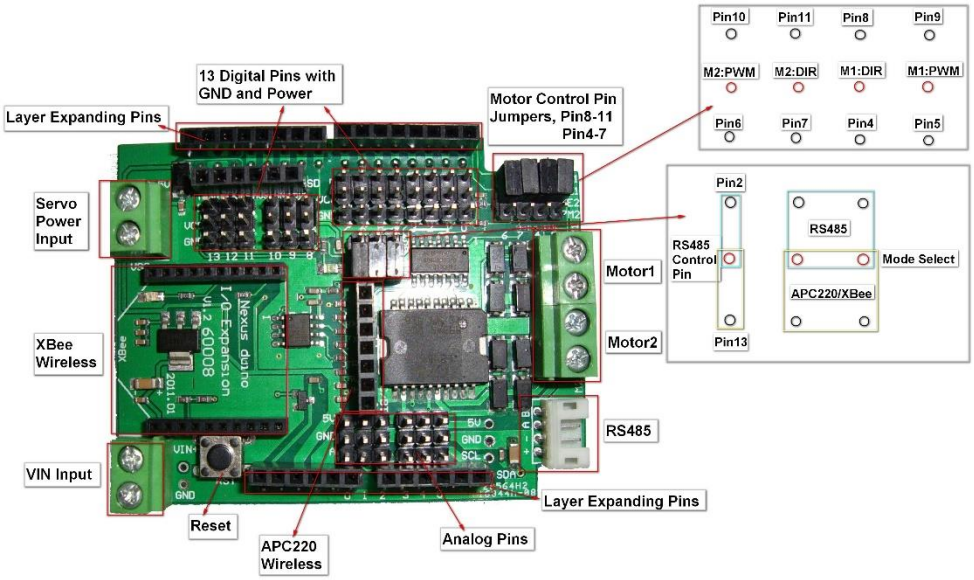
PRILOZI

Prilozi sadrže materijale koji nisu neophodni za praćenje rada. Pojedini dijelovi rada pozivaju se na priložene materijale.

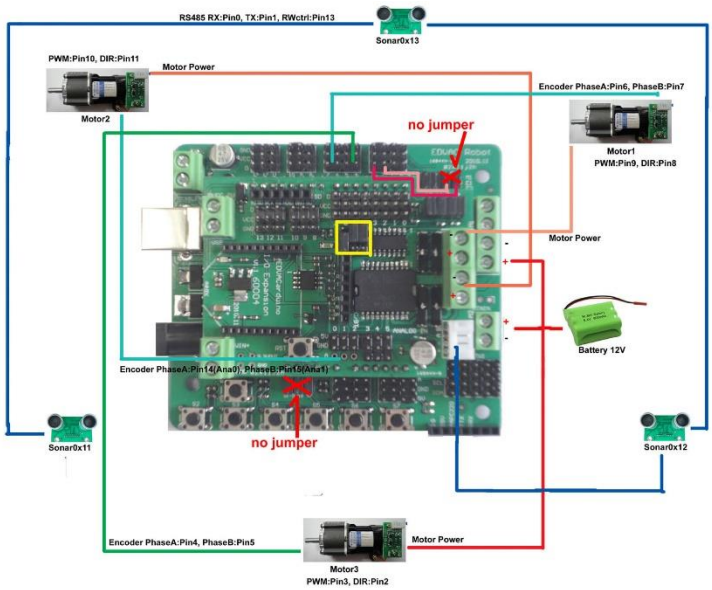
Prilog 1. Mikroupravljački sustav mobilne platforme

Materijali u nastavku preuzeti su s [3].

P1.1. Prikaz nožica U/I dodatka za mikroupravljač mobilne platforme



P1.2. Raspored nožica mikroupravljačkog sustava mobilne platforme



Prilog 2. Značajke robotskog manipulatora Dobot Arm V1.0

Tablica u nastavku izvorno je dio korisničkih uputstava odabranog robotskog manipulatora [12].

Broj osi	*4*
Masa [kg]	*3*
Dimenzije baze [mm]	*170 * 150*
Nosivost [g]	*500*
Ponovljivost [mm]	*0.2*
Materijal	*Aluminij 6061*
Upravljačka jedinica	*Arduino Mega 2560*
Komunikacija	*UART/Bluetooth*
Napajanje	*12V/5A DC*

Zglob	*Radni prostor (u stupnjevima)*
Zglob 1	*Od 135 do -135*
Zglob 2	*Od 85 do -5*
Zglob 3	*Od 95 do -10*
Zglob 4	*Od 90 do -90*

Prilog 3. ROS paket *omnibot*

Sve korištene datoteke moguće je pregledati u cijelosti unutar DVD priloga.

- *omnibot*
  - *include*
    - *omnibot*
  - *package.xml*
  - *CMakeLists.txt*
  - *scripts*
    - *dobot_kinematics.py*
    - *dobot_teleop.py*
    - *DobotDriver.py*
    - *DobotKinematics.py*
    - *DobotSDK.py*
    - *omni_kinematics.py*
    - *omni_teleop.py*
    - *Robot.py*
    - *serial_aggregator.py*
  - *msg*
    - *DobotControl.msg*
    - *WheelSpeed.msg*