Razvoj strojne in programske infrastrukture za module rekonfigurabilne robotske celice

Jakob Purg[1], **Timotej Gašpar**[2]

[1]*Fakulteta za elektrotehniko, Tržaška 25, 1000 Ljubljana,*
[2]*Institut "Jožef Stefan", Jamova cesta 39, 1000 Ljubljana*
E-pošta: jakobpurg@gmail.com

Developement of hardware and software infrastructure for reconfigurable robot cell modules

Frequent product launches and changes in demand require rapid adaptations in industrial manufacturing. Small and medium sized enterprises (SMEs) are feeling the arising pressure the most as they tend to fine tune their production in a "just-in-time" manner. To tackle these problems there is an increase of research in the field of reconfigurable robotic cells.

Those robot workcells can have different peripheral equipment. In this paper we present peripheral modules that can be quickly plugged or unplugged to the cell with the custom made "plug-and-produce" (P&P) connector. The connector allows the pass-through of pressured air, electrical power and communication via Ethernet. This allows the modules to receive all the necessary for their operation and at the same time they are connected to the cell's software network. To ease the access to the functionalities of these modules, we built a web Server that runs on a micro-computer on board of the module.

The web server was built using NodeJS and JavaScript programming language so that it can be easily transferred to different modules and set working with as little modifications as possible.

1 Uvod

Globalni trg je zelo kompetitiven in pritiski po zagotavljanju dovoljšne proizvodnje ter zagotavljanje kakovosti so za proizvodno usmerjena podjetja visoki. Te pritiske še bolj občutijo majhna oz. srednje velika podjetja z maloserijskimi proizvodnjami. Prav v tovrstnih podjetjih pa je avtomatizacija redka saj je vložek sredstev vanjo večinokrat previsok. Visoki stroški pa ne izhajajo le iz visokih cen strojev ampak tudi iz porabljenega časa za postavitev avtomatizirane proizvodne linije [1]. Da bi približali avtomatizacijo in robotiko tudi takim podjetjim je v porastu trend razvoja prilagodljivih oz. rekonfigurabilnih robotskih celic. To so celice, ki jih je mogoče relativno hitro postaviti, je njihovo obliko enostavno prilagoditi in so v principu modularne.

Primere implementacije modularnih celic lahko vidimo tudi kot trend h kateremu se giblje razvoj rekonfigurabilnih celic [2, 3, 4]. Dober primer take implementacije predstavi Chen v svojem delu, ki se osredotoča na iskanje optimalnih konfiguracij modulov celice za izvajanje specifične naloge [5].

Robotska celica, ki je predmet tega dela, je bila razvita s sledenjem smernicam za rekonfigurabilne robotske celice v sklopu projekta "ReconCell" [6, 7]. Celica je zgrajena iz ogrodja, ki omogoča hitro postavitev ter rekonfiguracijo. Ima dva robota UR-10m, oba opremljena s sistemom izmenjevalcev orodja. Celica je seveda tudi modularna, kar omogočajo posebni konektorji, ki so bili razviti v sklopu projekta.

Informacijski sistem celice je bil zgrajen na podlagi sistema "Robot Operating System – ROS". Gre za odprtokodno delovno okolje (*ang. framework*), ki ponuja velik nabor knjižnic in orodij za hiter razvoj robotske programske opreme [8].

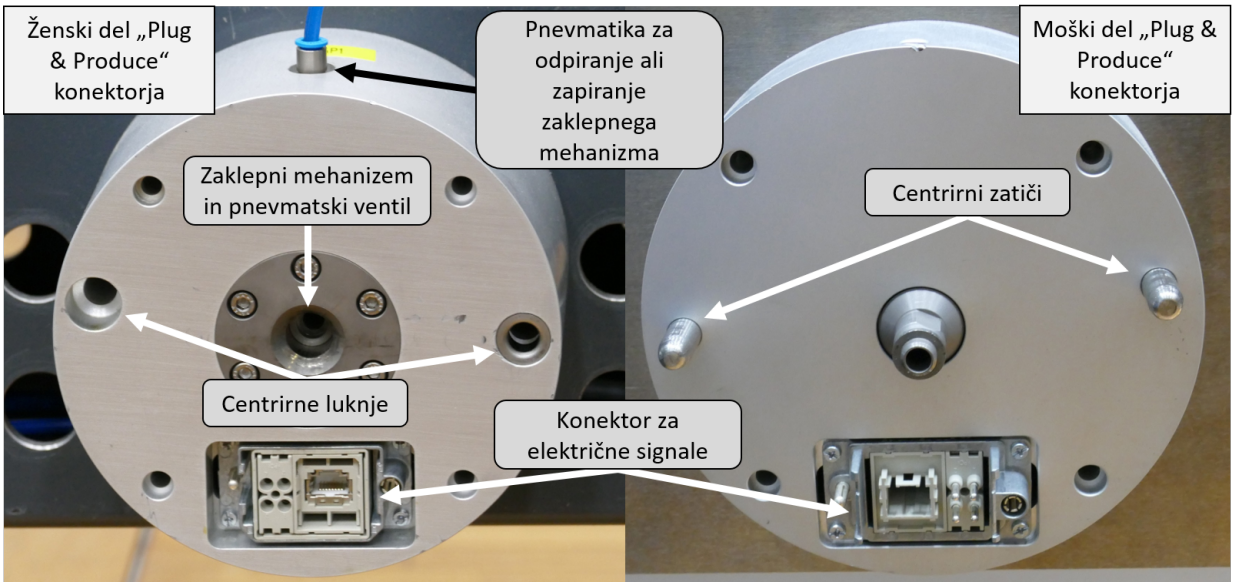
V tem delu bomo predstavili opremo ter orodja, ki smo jih razvili za podporo uporabe modulov celice. Cilj je bil module opremiti z mikroračunalnikom, ki omogoča povezovanje posameznih modulov v ROS omrežje ter njihovo krmiljenje. Hkrati pa smo želeli, da je na mikroračunalnikom postavljen strežnik, ki omogoča upravljanje z nekaterimi funkcionalnostimi modula preko spletnega brskalnika. Razvito strojno opremo bomo opisali v poglavju 2, programsko opremo pa v poglavju 3.

2 Moduli v rekonfigurabilni celici

Da bi povečali stopnjo rekonfigurabilnosti robotske celice, smo določene komponente celice razvili kot module. Namen je bil omogočiti hitro spremembo namembnosti celice s tem, da se lahko module v celico poljubno priklopi ali od nje odklopi. Zahteva po hitrih priklopih in odklopih pa je privedla do razvoja dodatne strojne kot tudi programske opreme. Tekom projekta je bil razvit poseben konektor za priklop modulov v celico ter programska oprema za podporo hitrih priklopov in odklopov modulov.

2.1 Konektorji za hiter priklop modulov

Moduli morajo biti zasnovani na takšen način, da dopuščajo hiter priklop in odklop. Hkrati pa mora ta priklop zagotavljati togo in ponovljivo sklopitev s celico, saj v nasprotnem primeru robot ne more natančno uporabljati modulov. Dodatna lastnost, ki jo mora konektor zagotavljati je, da modul dobi vse potrebne priklope za obratovanje,



Slika 1: “Priklopi in Proizvajaj” konektor.

t. j. napajanje, komprimiran zrak ter komunikacijski kanal. Tem zahtevam zadošča konektor, ki smo ga tekom projekta razvili in sicer t. i. “Priklopi in Proizvajaj” (*ang. Plug and produce – P&P*) konektor. Ženski del konektorja je togo pričvrščen na ogrodje celice, moški pa na perifernih modulih. Konektor omogoča poleg toge sklopitve tudi pretok komprimiranega zraka, omrežno napetost ter Ethernet povezavo. Tako lahko zagotovimo, da periferni moduli dobijo vse potrebno za delovanje ob sklopu na konektor.

Konektor je okrogle oblike in ima na sredini zaklepni mehanizem, ki hkrati služi kot ventil za dovod zraka v module. Na spodnjem delu ima priključke za električne signale, kar pomeni ethernet povezavo ter omrežno napetost. Ethernet povezava se sklene z RJ45 konektorjem, omrežna napetost pa preko namenskih priključkov. Ob straneh ima centrirne luknje (ženski del) oz. zatiče (moški del), za natančno orientacijo ter togo sklopitev (slika 1). Ventil na sredini je zgrajen tako, da je zaprt, torej zraka ne prepušča, ob odsotnosti modula. Konektorjev zaklepni mehanizem se odklene s pomočjo pnevmatike, ki je dovedena na vrhu ženskega konektorja. To pomeni, da je potrebna le takrat, ko želimo modul od celice odklopiti, kadar pa želimo modul v celico priklopiti, pa je dovolj, da dva konektorja sklenemo skupaj s potiskom.

2.2 Obstoječi moduli

Razvito robotsko celico smo do sedaj preizkusili na treh različnih proizvodnih procesih v laboratorjiskih pogojih, kar pomeni, da se nismo osredotočali na hitre cikle ampak na izvedljivost. Prvi proces je bil sestavljanje dveh različnih avtomobilskih luči, drugi se je osredotočal na sestavljanje pogonskega sklopa motoriziranega pohištva, tretji pa je predvideval sestavljanje elektromagnetne sklopke.

Da bi lahko vse te proizvodne procese izvedli v eni celici, je bilo razvitih nekaj različnih modulov. S tem smo dosegli želeno fleksibilnost celice in zadostili zahtevi po hitri rekonfiguraciji. Za naštete primere smo razvili sledeče module:

- pilagodljiv vpenjalni modul za avtomobilske luči,
- vpenjalni modul za pogonski sklop motoriziranega pohištva (slika 2),
- modul s prešo za sestavljanje elektromagnetne sklopke,
- modul z naborom robotskih orodij, ki jih lahko robota po potrebi uporabita,
- modul s katerega robota pobirata kose za sestavljanje.

Trenutno so vsi razviti moduli na kolesih in jih je za premikanje potrebno potiskati. Namen je, da se v bodoče razvije tudi avtonomne module, ki bi se lahko v celico priklopili in odklopili sami. Hkrati, pa je želja tudi robote v celico dodati kot modul, saj bi na ta način lahko dosegli večjo fleksibilnost pri sprotni rekonfiguraciji celice.

3 Programska oprema

Kot že prej omenjeno, je informacijska infrastruktura robotske celice zgrajena na podlagi sistema ROS (slika 3). Vsi programski gradniki med seboj komunicirajo tako, kot predvidevajo konvencije v ROS-u. To nam omogoča relativno enostaven razvoj novih gradnikov oz. funkcionalnosti celice. Da smo lahko module vključili v ROS, jih je bilo potrebno opremiti z neke vrste računalnikom, na katerem lahko tečejo ROS programi. V našem primeru je bil to mikroračunalnik *Raspberry Pi 3 model B*, ki je zelo popularen v ROS-ovi mednarodni skupnosti, ima zadostno število vhodno/izhodnih vrat ter je primerno majhen.

Slika 2: Vpenjalni modul za pogonski sklop motoriziranega pohištva.

3.1 ROS infrastruktura

Centralni del sistema ROS sestavlja t. i. ROS jedro (*ang. roscore*). To je program, ki mora teči na enem od gradnikov celice, do katerega imajo vsi ostali gradniki dostop. Jedro nosi informacije o vseh ostalih programih oz. vozliščih (*ang. node*) ter o podatkih, ki jih ta vozlišča od sebe dajajo oz. prejemajo. To je narejeno po sistemu "prijavi" in "objavi" (*ang. subscribe and publish*). Ti podatki so nato v omrežju dostopni vsak na svoji temi (*ang. topic*). Spisek vseh objavljenih tem je v jedru, tako da vozlišča ne potrebujejo informacije o tem kdo podatke obljavlja, ampak le informacijo na kateri temi podatke najti.

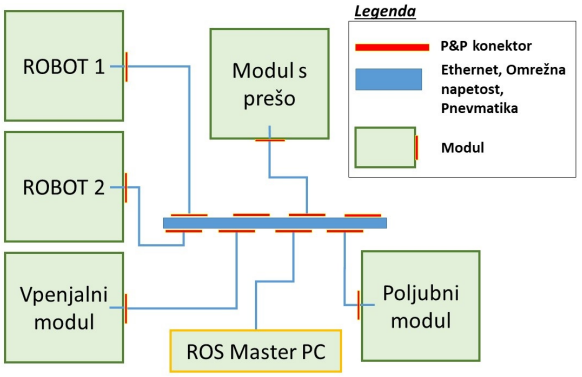
Krmiljenje modulov smo zastavili tako, da na mikroračunalniku teče program, ki ob ukazu spremeni stanje digitalnih izhodov. Poslužili smo se ROS mehanizma storitev (*ang. service*). Ta mehanizem omogoča, da vozlišče v sistemu objavi katere storitve lahko izvede in na kakšen način se jih proži. Storitev, ki smo jo razvili omogoča spreminjanje stanja digitalnih izhodov mikroračunalnika. Ti so nato povezani na elektromagnetne pnevmatske ventile. Slednji lahko nato odpirajo ter zapirajo vpenjala ali pa sprostijo zavore prilagodljivih vpenjal [9].

Razvoj vozlišča na modulovem mikroračunalniku sicer omogoča dostop do funkcionalnosti modula znotraj ROS omrežja. Težava tega pristopa je, da mora računalnik, iz katerega želimo dostopati do teh funkcionalnosti, imeti naložene knjižnice in programe, ki omogočajo komunikacijo s sistemom ROS. Da bi povečali dostopnost do teh funkcionalnosti, smo razvili strežnik, ki uporabnikom, s poznavanjem IP naslova modula, preko internetnega brskalnika ponuja nekatere od teh funkcionalnosti.

3.2 Strežnik

Strežnik je bil razvit v programskem jeziku *JavaScript*, poganja pa ga *Node.js* [10]. Za dostop do nekaterih ROS funkcionalnost, znotraj programskega okolja *JavaScript*, smo uporabili knjižnico *rosnode*. Tako samo lahko razvili strežnik, ki teče na modulu in streže *HTML* stran komurkoli, ki se nanj poveže. Dostopnost do strani je omogočena le, če je odjemalec v omrežju celice.

Strežnik najprej iz datoteke, ki vsebuje celoten zaganjalni paket - *roslaunch*, prebere imena ter tipe storitev, ki so na voljo na posameznem modulu, se prijavi na storitve, ki so pod istim imenom že v ROS omrežju, ter prebere trenutna stanja in tipe posameznih funkcij oz. funkcionalnosti. To informacijo nato preko JSON-a (*ang. JavaScript Object Notation*) pošlje do odjemalca (*ang. Client*). Odjemalec samodejno zgenerira primeren tip ter stanje funkcije oz. funkcionalnosti na grafičnem vmesniku v HTML datoteki, glede na prejete podatke. To stori s pomočo knjižnice jQuerry. Funkcije na strežniku krmilimo preko JSON POST formata, kjer je za delovanje potrebna tudi primerena prošnja (*ang. Request*) na strežnik (glej sliko 5). Do teh funkcij lahko nato s poznavanjem URL-ja ter prošnje dostopamo tudi preko drugih storitev, v kolikor smo v istem ROS omrežju.



Slika 3: Komunikacijska shema robotske celice.

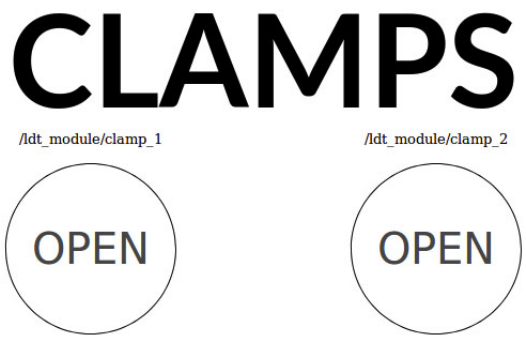
3.3 Spletni vmesnik

Stran je tako uporabniški vmesnik za modul. Primer vmesnika, ki je bil razvit za vpenjalni modul je viden na sliki 4.

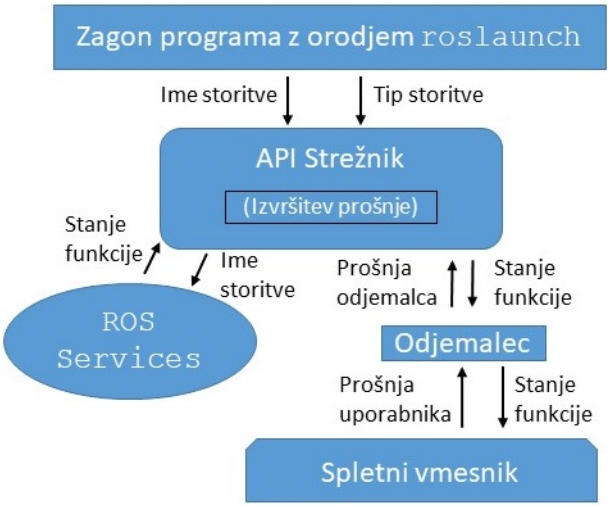
Prednost takšnega pristopa je, da uporabnik ne potrebuje ROS orodij za dostop do nekaterih funkcionalnosti celice. Dostop do funkcionalnosti vpenjalnega modula preko vmesnika med proizvodnim procesom seveda ni potreba, saj takrat celica deluje popolnoma samostojno oz. avtonomno. Taki vmesniki so predvsem uporabni med načrtovanjem ter programiranjem proizvodnega procesa.

4 Zaključek in nadalnje delo

V pričujočem članku smo predstavili podporno strojno in programsko opremo za module v rekonfigurabilni robotski celici. Pokazali smo, kako lahko s posebnimi "Priklopi in proizvajaj" konektorji dosežemo hitro izmenjavo



Slika 4: Uporabniški vmesnik za uporabljanje vpenjalnega modula.



Slika 5: Diagram poteka izmenjave informacije.

modulov celice in s tem tudi hitro spremembo njene namembnosti. Da bi zagotovili enostavnost uporabe modulov smo razvili *HTTP* strežnik, ki uporabnikom v spletni brskalnik streže uporabniški vmesnik. S tem smo pripomogli k dodatnemu lajšanju postopka programiranja dotične robotske celice.

Nadalnje delo vključuje vlaganje v razvoj avtonomnih modulov, ki bi se lahko v celico, preko *P&P* konektorjev sami priključili. Taki moduli bi lahko v industrijskih okoljih dodatno pripomogli k povečanju avtonomije rekonfiguracije robotske celice.

Literatura

[1] T. Dietz, U. Schneider, M. Barho, S. Oberer-Treitz, M. Drust, R. Hollmann, and M. Hägele, "Programming system for efficient use of industrial robots for deburring in SME environments," in 7th German Conference on Robotics; Proceedings of ROBOTIK 2012, pp. 1–6, VDE.

[2] R. M. Setchi and N. Lagos, "Reconfigurability and reconfigurable manufacturing systems: state-of-the-art review," in IEEE International Conference on Industrial Informatics (INDIN), pp. 529–535, 2004.

[3] Z. M. Bi, S. Y. T. Lang, M. Verner, and P. Orban, "Development of reconfigurable machines," The International Journal of Advanced Manufacturing Technology, vol. 39, pp. 1227–1251, Dec. 2008.

[4] M. Fulea, S. Popescu, E. Brad, B. Mocan, and M. Murar, "A literature survey on reconfigurable industrial robotic work cells," Applied Mechanics and Materials, vol. 762, p. 233, 2015.

[5] I.-M. Chen, "Rapid response manufacturing through a rapidly reconfigurable robotic workcell," Robotics and Computer-Integrated Manufacturing, vol. 17, no. 3, pp. 199–213, 2001.

[6] T. Gašpar, B. Ridge, R. Bevec, M. Bem, I. Kovač, A. Ude, and v. Gosar, "Rapid hardware and software reconfiguration in a robotic workcell," in 18th International Conference on Advanced Robotics (ICAR), pp. 229–236, IEEE.

[7] "ReconCell." `http://www.reconcell.eu/`. Accessed: 2018-07-10.

[8] "ROS - The Robot Operating System." `http://www.ros.org/`. Accessed: 2018-07-10.

[9] M. Bem, M. Deniša, T. Gašpar, J. Jereb, R. Bevec, I. Kovač, and A. Ude, "Reconfigurable fixture evaluation for use in automotive light assembly," in 2017 18th International Conference on Advanced Robotics (ICAR), pp. 61–67, July 2017.

[10] "NodeJS - JavaScript Node library." `https://nodejs.org/en/`. Accessed: 2018-07-11.