



Dimić, Z., Milutinović, D., Živanović, S., Mitrović, S., Kvrgić, V. ¹⁾

METOD KONFIGURISANJA UPRAVLJAČKOG SISTEMA OTVORENE ARHITEKTURE REKONFIGURABILNE ROBOTSKE ČELIJE ZA OBRADU²⁾

Rezime

U radu je prikazan metod konfigurisanja/rekonfigurisanja upravljačkog sistema otvorene arhitekture rekonfigurabilnih robotskih ćelija za obradu. Na primeru rekonfigurabilnog upravljačkog sistema koji je razvijen primenom softverskih alata i tehnologija otvorene arhitekture, posredstvom dijagrama za modelovanje dinamičkih procesa definisane su aktivnosti sistem-integratora/korisnika u toku konfigurisanja/rekonfigurisanja upravljačkog sistema, a uspostavljene su i relacije između softverskih i hardverskih komponenta upravljanja.

Ključne reči: upravljački sistem, rekonfigurabilnost, otvorena arhitektura, robot za obradu

1. UVOD

Ideja da se postojeći CAD/CAM sistemi koriste za programiranje robota u G-kôdu za višeosnu obradu rezanjem navodi na razvoj upravljačkog sistema koji treba korisniku da omogući emulaciju različitih 5-osnih mašina alatki robotima vertikalne zglobove konfiguracije. S obzirom na činjenicu da upravljački sistem svojom rekonfigurabilnošću treba da pokrije više konfiguracija robotskih ćelija za obradu, ali i da omogući integratorima sistema da sami kreiraju nove konfiguracije, odgovarajući softverski upravljački moduli, kao što su kinematički moduli, interfejs moduli i sl., moraju biti dostupni za modifikaciju i/ili proširenje. Uz činjenicu da mogućnost konfigurisanja i rekonfigurisanja upravljanja zavisi od otvorenosti, modularnosti i skalabilnosti upravljačke arhitekture [1], u ovom radu je data metoda konfigurisanja/rekonfigurisanja softverski orijentisanog upravljačkog sistema robota za obradu, realizovanog primenom softverskih alata i tehnologija otvorene arhitekture, uz razvoj algoritama i softverskih funkcija upravljanja.

2. KONCEPT RAZVOJA REKONFIGURABILNOG UPRAVLJAČKOG SISTEMA ROBOTA ZA OBRADU

Realizacija upravljačkog sistema robotskih ćelija za obradu se u predmetnom radu zasniva na primeni sopstvenih algoritama upravljanja, softverski implementiranih korišćenjem *OROCOS (Open ROBOT COntrol Software)* alata otvorene arhitekture. Otvorenost arhitekture je sve prisutnija u oblasti upravljanja robotima i mašinama alatkama. Na sličan način na koji je uticala na revolucionaran rast industrije personalnih računara (PC), otvorenost utiče na razvoj i širenje primene industrijskih robota. Prema *IEEE 1003.0 (Institute of Electrical and Electronics Engineers)* [2] modelu, otvoreni sistem je definisan kao sistem koji omogućava da se aplikacije implementirane na odgovarajući način mogu izvršavati na hardverskim platformama različitih proizvođača, komunicirati sa drugim sistemskim aplikacijama i omogućiti konzistentni način interakcije sa korisnikom.

Prema usvojenom konceptu u ovom radu, razvoj rekonfigurabilnog upravljačkog sistema zasniva se na:

¹⁾ Dr Zoran Dimić, dipl. inž. el., Lola institut d.o.o., Beograd, (zoran.dimic@li.rs), prof. dr Dragan Milutinović, Univerzitet u Beogradu, Mašinski fakultet, (dmilutinovic@mas.bg.ac.rs), prof. dr Saša Živanović, Univerzitet u Beogradu, Mašinski fakultet, (szivanovic@mas.bg.ac.rs), Stefan Mitrović, mast. inž. maš., Lola institut d.o.o., Beograd, (stefan.mitrovic@li.rs), dr Vladimir Kvrgić, dipl. maš. inž, Institut „Mihajlo Pupin“, (vladimir.kvrgic@pupin.rs)

²⁾ U okviru ovog rada saopštavaju se rezultati istraživanja koja se sprovode na projektima: TR_35023: Razvoj uređaja za trening pilota i dinamičku simulaciju leta modernih borbenih aviona i to 3-osne centrifuge i 4-osnog uređaja za prostornu dezorijentaciju pilota i TR_35022: Razvoj nove generacije domaćih obradnih sistema, koji finansijski podržava Ministarstvo prosvete, nauke i tehnološkog razvoja

- Sopstvenom razvoju kompilatora programa obrade (*G-kôda*, *STEP-NC-a*);
- Sopstvenom razvoju interpolatora;
- Implementaciji razvijenih kinematičkih modula;
- Primeni softverskih alata otvorene arhitekture za kreiranje i sinhronizaciju upravljačkih modula ili realizaciju sopstvenih alata;
- Primenu softverskih alata otvorene arhitekture za odabir i povezivanje upravljačkih modula, odnosno konfigurisanje i rekonfigurisanje sistema ili realizaciju sopstvenih alata;
- Operativnom sistemu računara otvorene arhitekture za rad u realnom vremenu;
- Računarskoj hardverskoj platformi otvorene arhitekture.

Razvoj upravljačkog sistema na bazi softverskih alata i hardverskih tehnologija otvorene arhitekture, uz razvoj algoritama upravljanja, omogućava precizno ispunjenje zahteva korisnika, kako u pogledu performansi sistema, tako i po pitanju izgleda i strukture korisničkih interfejsa, kao i verifikacionih softverskih alata.

Kako procedura rekonfigurisanja mora biti praćena odgovarajućim testovima i simulacijama u cilju potvrde funkcionalnosti i bezbednosti novoformirane konfiguracije, uz konfiguracioni interfejs i odgovarajući verifikacioni softverski alati moraju biti integralni deo rekonfigurabilnog upravljačkog sistema.

3. PRIKAZ MORFOLOŠKOG METODA ZA KONFIGURISANJE I REKONFIGURISANJE UPRAVLJAČKOG SISTEMA NA BAZI *OROCOS-A*

OROCOS [3] je nastao kao rezultat evropskog projekta pokrenutog 2001. godine na belgijskom Katoličkom univerzitetu u Luvenu, uz učešće Laboratorije za analizu i arhitekturu sistema (*CNRS/LAAS*) iz Francuske i švedske laboratorije *KTH* (*Kungl Tekniska Högskolan*). Pored ovih institucija, na razvoju i dizajnu *OROCOS-a* učestvuju i mnoge druge laboratorije i instituti. Osnovni cilj projekta je razvoj modularnog softverskog okruženja (eng. *framework*) otvorene arhitekture (*open-source*) koje pruža funkcionalnu osnovu za implementaciju upravljačkih sistema robota.

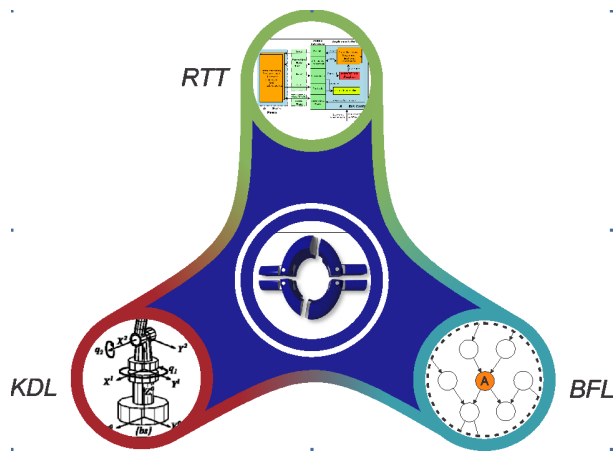
OROCOS je realizovan kao softverski sistem, nezavisan od platforme i aplikacije koju implementira, sa sledećim karakteristikama:

- Otvorenost i dostupnost softverskog sistema za proučavanje, upotrebu i modifikacije;
- Modularanost i fleksibilnost koja pruža mogućnost korisnicima da izgrade svoj sistem po želji, a programerima da doprinesu izgledu komponenata, bez potrebe za detaljnim sagledavanjem suštine celokupnog softverskog sistema;
- Nezavisnost od komercijalnih proizvođača robota, ali i kompatibilnost sa robotskim sistemima nekoliko evropskih proizvođača;

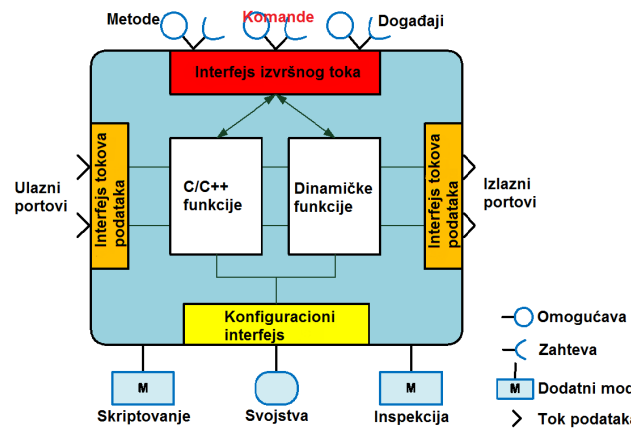
OROCOS softverski sistem čine sledeće softverske biblioteke, slika 1:

- *RTT* (*Real-Time Toolkit*), koja sama po sebi ne predstavlja aplikaciju, već obezbeđuje infrastrukturu i funkcionalnosti za razvoj aplikacija za upravljanje u *C++*. Naglasak je na *real-time*, *on-line* interaktivnim aplikacijama baziranim na komponentama. *RTT* se oslanja na *real-time* kernel matičnog operativnog sistema i enkapsulira ga svojim *real-time* aplikativnim interfejsom.
- *OCL* (*OROCOS Component Library*) obezbeđuje biblioteku sa ograničenim brojem komponenata već spremnih za upotrebu. Neke komponente mogu da posluže kao primer, a neke implementiraju veoma kompleksne algoritme uz korišćenje *RTT-a*;
- *KDL* (*Kinematics and Dynamics Library*) je *C++* biblioteka koja omogućava numeričko izračunavanje kinematike serijskih mehanizama u realnom vremenu.
- *BFL* (*Bayesian Filtering Library*) obezbeđuje aplikaciono nezavisni okvir za realizaciju dinamičkih Bajesovih mreža, npr. rekurzivnu obradu i estimaciju algoritama zasnovanih na Bajesovoj teoremi (Kalmanovi filtri, itd.).

Izgradnja rekonfigurabilnog upravljačkog sistema na bazi *OROCOS-a*, o kome je reč u predmetnom radu, podrazumevala je kreiranje softverskih komponenata različite hijerarhijske pripadnosti u upravljačkom sistemu [4]. Pojedinačni funkcionalni blokovi su implementirani u kontekstu komponente uz definisanje aplikativnog interfejsa, slika 2. To podrazumeva da su svi upravljački algoritmi implementirani u okviru izvršnog sistema *OROCOS* komponente, uz realizaciju servisnih funkcija i tokova podataka pripadajućih elemenata aplikativnog interfejsa.



Slika 1. Gradivne softverske komponente OROCOS-a [3]



Slika 2. Aplikativni interfejs OROCOS komponente

Osnovna konfiguracija za upravljanje rekonfigurabilnom robotskom ćelijom za obradu u realnom vremenu sadrži sledeći skup komponenata:

- Interpreter međukoda (*P-kôda*) sa odabranim modulima interpolatora - *generatorPos*;
- Kinematički modul - *kinematics*;
- Servo kontroler - *nAxesContorllerPos*;
- Kontroler hardvera - *robotHardware*;
- Komponentu za ručno upravljanje - *manualMode*;
- Komponentu za inicijalizaciju osa robota - *initRobot*.

Procedura za konfigurisanje, odnosno rekonfigurisanje upravljačkog sistema obavlja se u sledećim fazama:

- Inicijalno konfigurisanje;
- Statičko rekonfigurisanje
- Dinamičko rekonfigurisanje.

U fazi **inicijalnog konfigurisanja** neophodno je koristiti razvojne kompilatorske alate, radi izgradnje komponenata karakterističnih za odabran skup konfiguracija robotske ćelije. Faze statičkog i dinamičkog rekonfigurisanja sprovode se izborom i/ili modifikacijama konfiguracionih datoteka, za šta nije potrebno imati programerska umeća.

Pod pojmom inicijalnog konfigurisanja upravljačkog podsistema podrazumeva se formiranje konačnog broja inicijalnih konfiguracija za upravljanje rekonfigurabilnom robotskom ćelijom za obradu. Razmatrani upravljački sistem treba da obuhvati sve softverske komponente potrebne za upravljanje odabranim konfiguracijama robotske ćelije za obradu, kao i zahtevanim aplikacijama obrade. Prema tome, zadatak integratora sistema, koji vrši inicijalno konfigurisanje *real-time* upravljačkog podsistema, treba da obuhvati:

- Izradu odgovarajućeg broja kinematičkih komponenata robotske ćelije sa pripadajućim *XML (eXtensible Markup Language)* datotekama svojstava;
- Odabir podmodula interpolatora i integraciju istih u jedinstvenu komponentu;
- Modifikaciju komponente *initRobot* uz implementaciju *XML* datoteka svojstava za sve definisane konfiguracije;
- Modifikaciju komponente *nAxesContorllerPos* uz implementaciju *XML* datoteka svojstava za sve definisane konfiguracije;
- Konfigurisanje komponente *robotHardware* prema broju aktivnih osa;
- Konfigurisanje komponente *manualMode* prema broju aktivnih osa;
- Integraciju komponenata *real-time* podsistema posredstvom *XML* konfiguracionog interfejsa;
- Konfigurisanje mašina stanja posredstvom *XML* interfejsa.

UML (Unified Modelling Language™) [5] dijagram slučajeva korišćenja u toku inicijalnog konfigurisanja *real-time* upravljačkog podsistema prikazan je na slici 3.

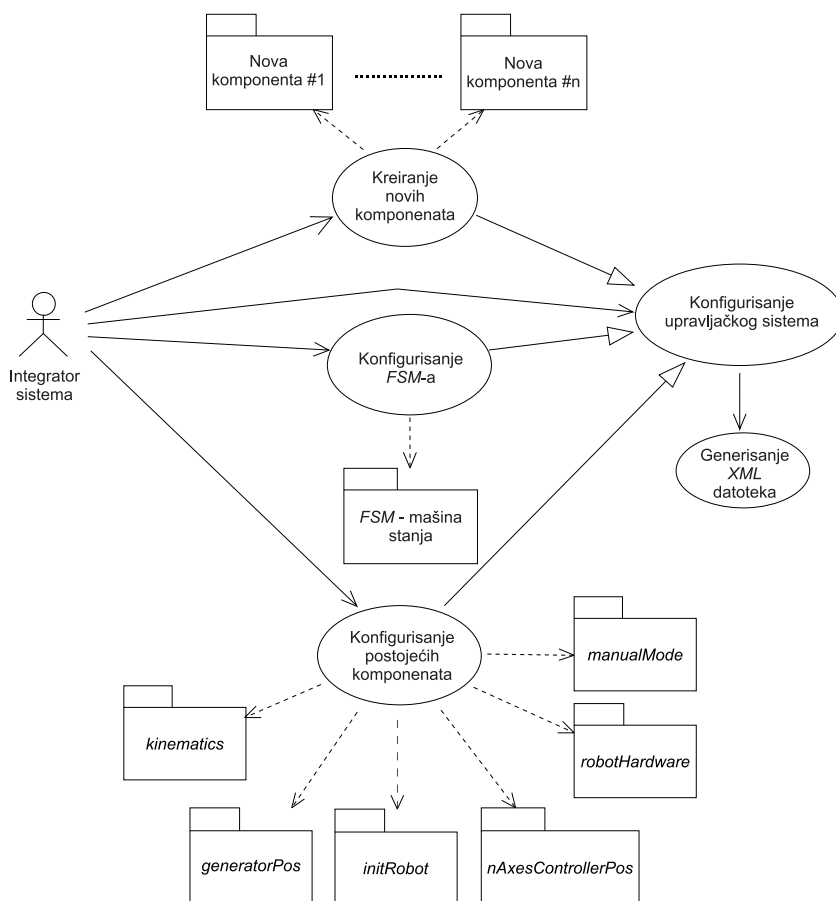
U fazi inicijalnog konfigurisanja formira se više konfiguracija. Informacije o komponentama koje ulaze u sastav jedne konfiguracije, kao i o načinu njihove interkonekcije, sadržane su u pridruženoj konfiguracionoj *XML* datoteci.

Faza **statičkog rekonfigurisanja** upravljačkog sistema podrazumeva izbor upravljačke konfiguracije robotske ćelije za obradu pre nego što sistem postane aktivan. Ova faza zahteva preduzimanje nekoliko

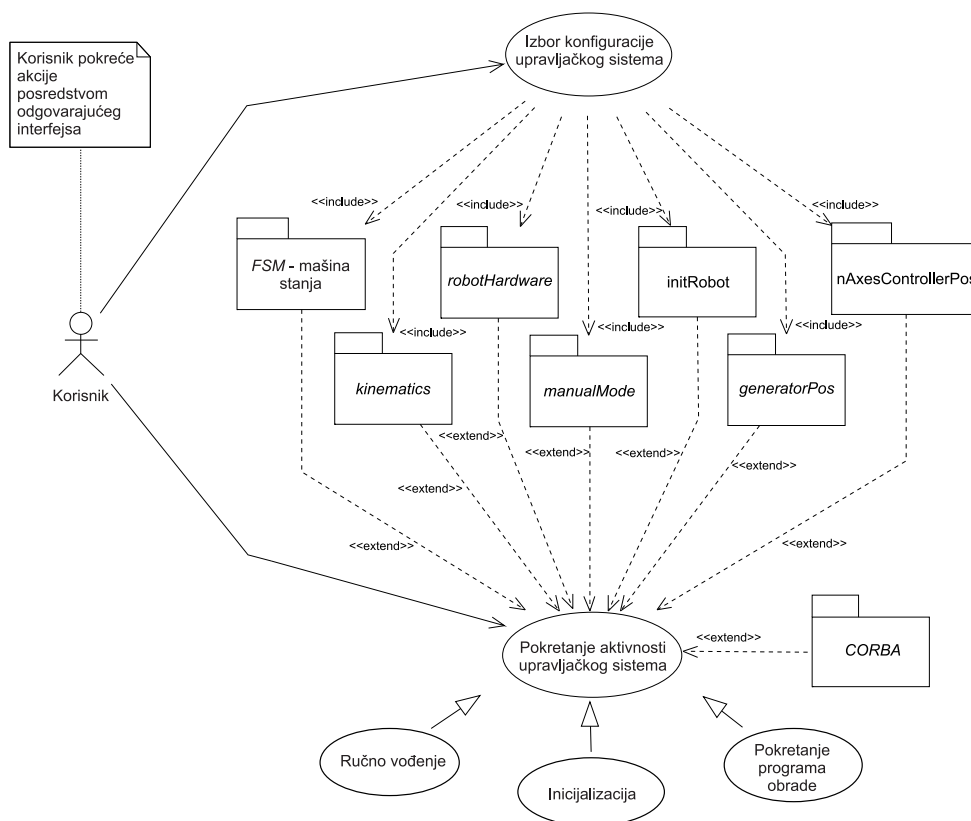
aktivnosti u cilju dobijanja funkcionalnog upravljačkog sistema. Pod pretpostavkom da je sistem inicijalno konfigurisan, odnosno da su definisane moguće konfiguracije robotske ćelije i upravljačkog sistema, izborom odgovarajuće XML konfiguracione datoteke pre pokretanja sistema određuje se upravljačka konfiguracija robotske ćelije.

Na opisan način određuju se komponente sa XML svojstvima koje ulaze u sastav upravljačke konfiguracije, veze između njih, kao i mašina stanja koja će voditi sistem tokom rada. Na slici 4 prikazan je UML dijagram slučajeva korišćenja u toku statičkog rekonfigurisanja upravljačkog podsistema.

Kada je reč o **dinamičkoj rekonfigurabilnosti**, smatraćemo da se akcije u cilju promene konfiguracije upravljačkog sistema sprovode u realnom vremenu, bez isključenja pogona i bez dealociranja memorije računarskog sistema, zauzete aktivnim upravljačkim komponentama.



Slika 3. Inicijalno konfigurisanje upravljačkog sistema



Slika 4. Primer statičkog rekonfigurisanje upravljačkog sistema

Drugim rečima, dinamičko rekonfigurisanje sprovodi se u trenucima kada je upravljački sistem aktivan. S obzirom na činjenicu da je potreba za promenama u kinematici robotske ćelije tokom obrade glavni razlog za dinamičkom izmenom konfiguracije upravljačkog sistema, u daljem tekstu će biti reči o mehanizmu za dinamičko rekonfigurisanje kinematičke komponente predloženog upravljačkog sistema.

Struktura robotske ćelije može se promeniti na više načina. Sa stanovišta rekonfigurisanja upravljačke jedinice od značaja su sledeća dva načina:

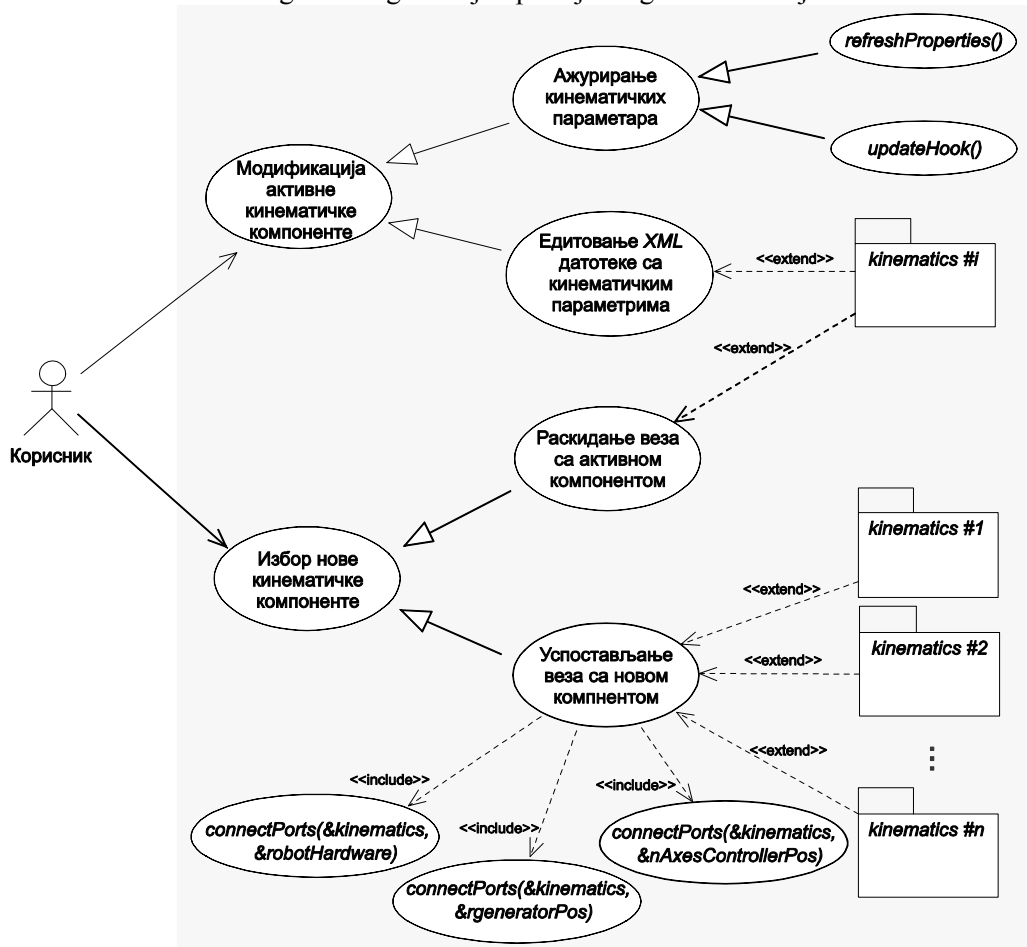
- Promene kinematičkih svojstava koje se reflektuju na geometrijske parametre, kao što su dužine segmenata i/ili broj stepeni slobode sistema;
- Promene u morfološkoj strukturi robotske ćelije.

Pomenuti načini rekonfigurisanja robotske ćelije definišu dva načina dinamičke rekonfigurabilnosti *real-time* upravljačkog podsistema:

- Modifikacijom kinematičke komponente;
- Zamenom kinematičke komponente.

Pod modifikacijom kinematičke komponente podrazumeva se izmena parametara kinematičkih funkcija, sadržanih u XML datoteci svojstava, koja treba da isprati promene geometrijskih parametara mehanizama robotske ćelije. S obzirom na to da XML datoteka predstavlja tekstualnu datoteku, izmena njenog sadržaja može se obaviti posredstvom nekog editora teksta ili specijalizovanog softvera izrađenog za datu namenu. Kako bi se modifikacije XML datoteke reflektovale na funkcionalnost kinematičke komponente, izmene moraju biti ispraćene odgovarajućim sistemskim pozivima.

Zamena kinematičke komponente podrazumeva raskidanje veza stare kinematičke komponente sa upravljačkim sistemom i drugim komponentama, a zatim obnavljanje prekinutih veza sa novom kinematičkom komponentom. Ovakav scenario je moguć samo u slučaju kada je nova kinematička komponenta, koja treba da preuzme ulogu, rezidentna u memoriji upravljačkog sistema. To praktično znači da prilikom podizanja softvera upravljačkog sistema moraju biti podignute sve komponente koje će u trenutku aktiviranja sistema, ili kasnije, ući u sastav upravljačke konfiguracije. UML dijagram slučajeva korišćenja u toku faze dinamičkog rekonfigurisanja upravljačkog sistema dat je na slici 5.



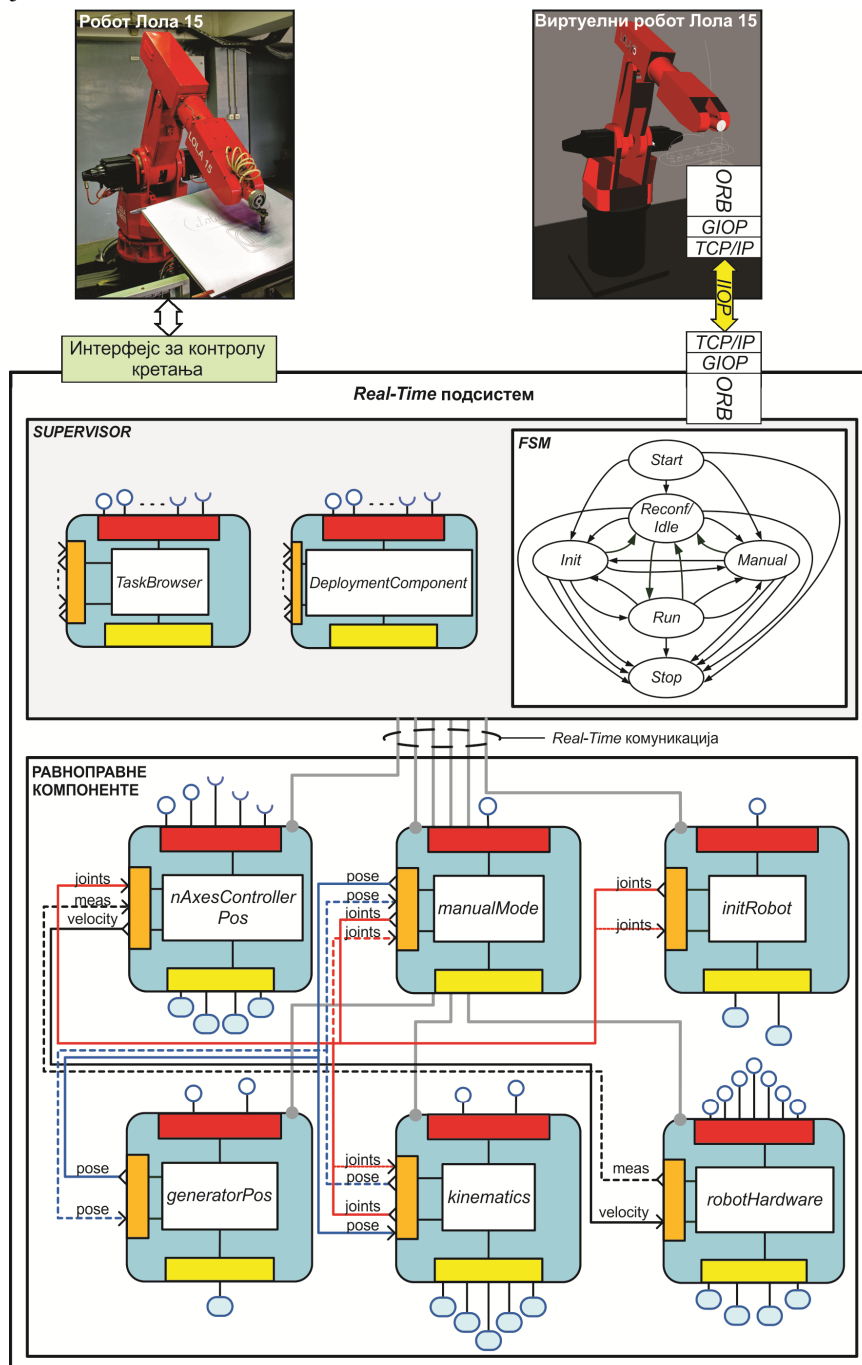
Slika 5. Primer dinamičkog rekonfigurisanja upravljačkog sistema

4. REALIZACIJA PROTOTIPA ELEMENTARNE KONFIGURACIJE UPRAVLJAČKOG SISTEMA PRIMENOM *OROCOS-A*

Na osnovu razvijenih upravljačkih algoritama i softverskih komponenta, uz korišćenje uspostavljene metodologije konfigurisanja upravljanja, u Laboratoriji za mašine alatke i robote instituta Lola konfigurisan je eksperimentalni prototip upravljačkog sistema. Elementarna konfiguracija prototipa upravljačkog sistema koja je testirana obuhvatila je sledeće softverske komponente:

- Komponente interpolatora, koje čine funkcionalnu osnovu komponente *generatorPos*;
- *kinematics* sa implementiranim funkcijama direktne i inverzne kinematike robota Lola 15;
- *nAxisContrillerPosition*;
- *robotHardver*, koja sadrži drajvere primenjenog spreženog hardvera;
- *manualMode*;
- *initRobot*.

Šematski prikaz veza između komponenta upravljačkog sistema, konfigurisanog za upravljanje robotom Lola 15, prikazan je na slici 6.



Slika 5. Upravljački sistem konfigurisan za upravljanje robotom Lola 15

Upravljačka logika implementirana je supervizorskom mašinom stanja (*FSM*), posredstvom *OROCOS* softverskih alata. *FSM* radi u reaktivnom režimu, pri čemu prelazak upravljačkog sistema iz jednog u drugo radno stanje podrazumeva aktivaciju odgovarajućih komponenata, odnosno deaktivaciju drugih, uz uspostavljanje, odnosno raskidanje odgovarajućih veza.

5. ZAKLJUČAK

Sa ciljem da se robotu za obradu omogući promena konfiguracije, a samim tim poveća fleksibilnost tokom obrade, u radu je predložen metod inicijalnog konfigurisanja upravljačkog sistema robota za obradu, kao i metod rekonfigurisanja kojim se omogućava emulacija različitih višeosnih mašina alatki. Primenom softverskih alata za modelovanje dinamičkih procesa, na transparentan način su prikazani postupci sistem integratora, odnosno korisnika, koje je neophodno izvesti tokom konfigurisanja, odnosno rekonfigurisanja upravljačkog sistema robota za obradu.

6. LITERATURA

- [1] Z. M. Bi & Sherman Y. T. Lang & M. Verner & P. Orban: *Development of reconfigurable machines*, International Journal of Advanced Manufacturing Technology, 39:1227–1251, 2008.
- [2] IEEE 1003.0: *IEEE GUIDE TO THE POSIX OPEN SYSTEM ENVIRONMENT (OSE)*. The Institute of Electrical and Electronics Engineers, Inc., 1995.
- [3] OROCOS - Open RObot COntrol Software. URL: <http://www.orocos.org>. maj 2018.
- [4] Dimić, Z. (2016) Upravljački sistem otvorene arhitekture rekonfigurabilnih robotskih ćelija za obradu, Doktorska disertacija, Univerzitet u Beogradu, Mašinski fakultet, Beograd
- [5] UML[®] - Unified Modeling Language[™], URL: <http://www.omg.org/spec/UML>, maj 2018.

Dimić, Z., Milutinović, D., Živanović, S., Mitrović, S.

A CONFIGURATION METHOD FOR OPEN ARCHITECTURE CONTROL SYSTEM OF RECONFIGURABLE ROBOTIC MACHINING CELLS

Abstract: *A method for configuration/reconfiguration of an open-architecture control system of reconfigurable robotic machining cells is given. By utilizing dynamic process modelling diagrams, activities of systems integrator/user regarding control system configuration/reconfiguration shall be presented by the example of open-architecture control software, altogether with established relations between hardware and software components of a given control system.*

Key words: *control system, reconfigurability, open architecture, machining robot*