

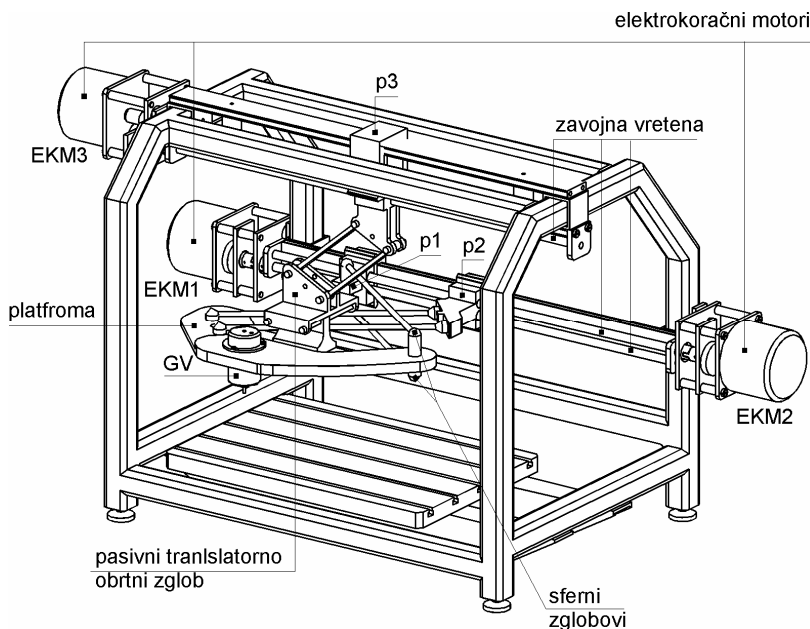
Saša Živanović, Zoran Dimić¹⁾

UPRAVLJANJE MODELA TROOSNE MAŠINE SA PARALELNO KINEMATIKOM pn101 NA BAZI EMC SISTEMSKOG SOFTVERA²⁾

Rezime

U radu je prikazano konfigurisanje sistema za upravljanje modelom troosne mašine sa paralelnom kinematikom. Na fizičkom modelu je testiran softver, koji može da se koristi i za višeosno upravljanje mašina alatki. Softver je verifikovan za troosnu obradu na fizičkom modelu mašine. Ostvareno je upravljanje koračnim motorima na modelu, na PC platformi pod Linux operativnim sistemom.

Ključne reči: upravljanje, mašina sa paralelnom kinematikom, koračni motori



Slika1. Fizički model pn101sa koračnim motorima

jedne od pogonskih osa. Za takvo izdvajanje jedne dominantne ose razvijen je paralelni mehanizam za horizontalne i vertikalne glodalice pod nazivom pn101. Detaljno je opisan u [1,2,3]. U poređenju sa sličnim napravljenim mehanizmima ovaj mehanizam ima bitno pravilniji oblik radnog prostora (nalik na modifikovanu prizmu), kako je to uobičajeno za serijske mašine.

Paralelni mehanizam pn101 nastao je kao rezultat projekta Troosne paralelne mašine, koji je rađen u saradnji sa LOLA Sistemom. Uporedo sa gradnjom industrijskog prototipa napravljen je i fizički model troosne mašine sa paralelnom kinematikom u razmeri 1:5, pokazan na slici 1. U početku je model korišćen samo za testiranje paralelnog mehanizma. Kasnije je dodat pogon na glavnom vretenu i na pogonskim osama pomoćnog kretanja. Modelu treba još dodati upravljanje da bi postao mašina, što je i predmet ovog rada. Upravljanje je realizovano na bazi EMC sistemskog softvera, koji radi pod Linux operativnim sistemom. Može se koristiti i za višeosno upravljanje, pa je pored troosnog iskorišćeno i za konfigurisanje petoosnog upravljanja.

1. UVOD

Mašine postoje da bi se koristile za neku transformaciju kretanja radi obavljanja planiranog posla. Radi njih zato postoje mehanizmi i nekakvo upravljanje i programiranje. Deo geometrije i kinematike mašine, koji potiče od mehanizma, ne može biti korigovan. Ako je upravljanje programabilno, onda se njime mogu podešavati neki elementi geometrije i kinematike mašine. Kada mehanizam mašine ima paralelnu kinematiku, tada su oblik i veličina radnog prostora slabost većine takvih mašina. Ipak, mehanizmi tipa Triaglide i Hexaglide primeri su mašina gde je radni prostor izdužen u pravcu

¹⁾ mr Saša Živanović, asistent (szivanovic@mas.bg.ac.yu), Mašinski fakultet, Kraljice Marije 16, 11120 Beograd, Zoran Dimić, dipl. ing. LOLA SISTEM PC MAA, (zdimic@yubc.net)

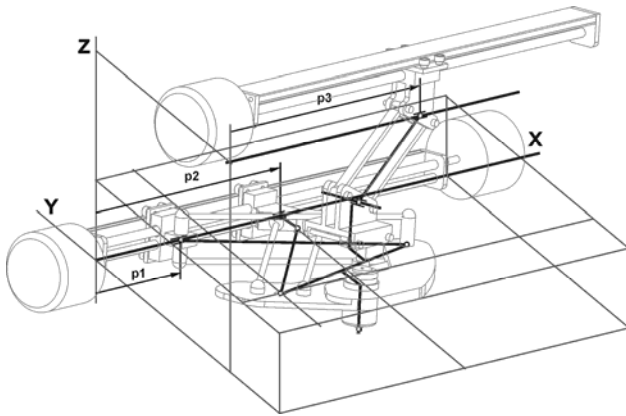
²⁾ Rađeno u okviru proteklog trogodišnjeg projekta MIS 3.02.0101 B Troosne paralelne mašine i tekućeg trogodišnjeg projekta TR6309B Petoosne paralelne mašine, u čijem finansiranju učestvuju Ministarstvo nauke i zaštite životne sredine Srbije i LOLA Sistem iz Beograda

2. KINEMATIČKI MODEL

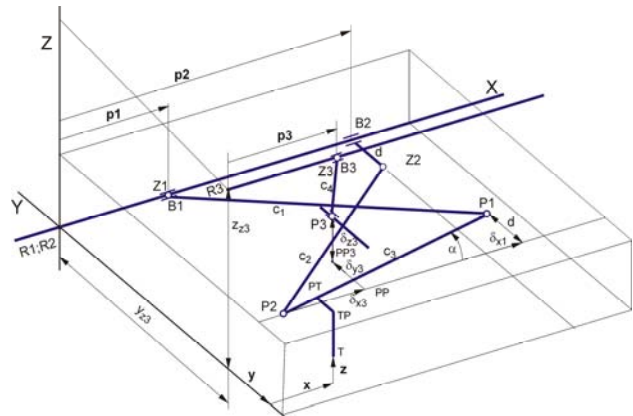
Platforma paralelnog mehanizma pn101 je pomoću dva para spojki sfernim zglobovima vezana za klizače p1 i p2. Sa trećim klizačem p3, platforma je vezana pomoću pasivnog translatorno obrtnog zgloba, slika 1. Komponente modela, da bi bile što lakše, su pretežno napravljene od aluminijuma (vođice, klizači, poklopac GV). Platforma je napravljena od plastike. Zavojna vretena su kupljena, dok je noseća struktura napravljena kao zavarena konstrukcija od šipkastog cevastog materijala kvadratnog poprečnog preseka.

Prve probe su vršene ručnim okretanjem zavojnih vretena, radi početnih provera stabilnosti i rada paralelnog mehanizma. Zatim su dodati i motori za ove tri translatorne ose. Razmatrane su varijante ugradnje koračnih motora (EKM) i motora jednosmerne struje. Kao konačno, usvojeno je rešenje sa koračnim motorima zbog lakše mogućnosti ostvarivanja upravljanja. Na pokretanoj platformi je postavljeno i osnaženo glavno vreteno, tako da model ima mogućnost obrade mekih materijala.

Geometrijski model sa slike 1 se za potrebe konkretnih izračunavanja, uprošćava na jednostavniji žičani kinematički model, prikazan na slici 2. Svaki par spojki se zamenjuje srednjom spojkom, a platforma linijom između tako osrednjenih zglobova, koji su obeleženi sa P1 i P2, slika 3. Pasivni translatorno obrtni zglob je takođe zamenjen linijama i klizačem.



Slika 2. Geometrijski i kinematički model



Slika 3. Kinematički model sa parametrima

Kinematički model sa parametrima za rešavanje inverzne i direktne kinematike dat je na slici 3. Koordinate karakterističnih tačaka mehanizma na bazi su : B1(p1; 0; 0); B2(p2; 0; 0); B3(p3; y_{z3}; z_{z3}); Z1=B1; Z2 (p2; -d; 0); Z3 = B3 ; Na pokretanoj platformi karakteristične tačke su: P1 (x_{p2} + δ_{x1}; y_{p2} + d; z_{p2}); P2 (x_{p2}; y_{p2}; z_{p2}); P3 (x_{p2} + δ_{x3}; y_{p2} + δ_{y3}; z_{p2} + δ_{z3}), gde je d = c₃ sin α .

Na osnovu kinematičkog modela mogu se uspostaviti sledeće geometrijske jednačine:

$$c_4^2 = (p_3 - x_{p3})^2 + (z_{z3} - z_{p3})^2; c_2^2 = (p_2 - x_{p2})^2 + (y_{p2} + d)^2 + z_{p2}^2; c_1^2 = (x_{p1} - p_1)^2 + (y_{p2} + d)^2 + z_{p1}^2 \quad (1)$$

Rešenje inverznog kinematičkog problema dobijeno je na osnovu kinematičkog modela i polaznih jednačina (1). Sva izračunavanja se vrše prema repernoj tački P2, od koje se lako može preračunati položaj vrha alata. Zadaje se: c₁, c₂, c₃, c₄, (α, d). Računa se sa (x_{p2}, y_{p2}, z_{p2}), ili sa (p₁, p₂, p₃). Neki od parametara mehanizma su: δ_{x1}, δ_{x3}, δ_{y3}, δ_{z3}, gde su (δ_{x1} = c₃ cos α); (δ_{y1} = d = c₃ sin α). Osnovni parametri realne mašine kao industrijskog prototipa su: c₁=1003mm, c₂=1026mm, c₃=1019mm, c₄=500mm. Odgovarajući parametri na realizovanom fizičkom modelu ove mašine su pet puta manji, odnosno: c₁=200.63mm, c₂=205.18mm, c₃=203.8mm, c₄=100mm.

Iz jednačina (1) i kinematičkog modela slede sledeće jednačine:

$$c_1^2 - c_2^2 = (x_{p1} - p_1)^2 + (y_{p2} + d)^2 + z_{p1}^2 - (p_2 - x_{p2})^2 - (y_{p2} + d)^2 - z_{p2}^2 \quad (2.1)$$

$$(z_{z3} - z_{p3})^2 = c_4^2 - (p_3 - x_{p3})^2 \quad (2.2)$$

$$c_2^2 = (p_2 - x_{p2})^2 + (y_{p2} + d)^2 + z_{p2}^2 \quad (2.3)$$

Rešenje direktnog kinematičkog problema (x_{p2}, z_{p2}, y_{p2}), dobijeno je na osnovu kinematičkog modela i jednačina 2.1, 2.2 i 2.3, respektivno. Rešenja IKP i DKP data su u Tablici 1.

Tablica 1. Rešenja IKP i DKP	
Rešenje Inverznog Kinematičkog Problema	Rešenje Direktnog Kinematičkog Pproblema
$p_3 = x_{p3} + \sqrt{c_4^2 - (z_{z3} - z_{p3})^2}$ $p_2 = x_{p2} + \sqrt{c_2^2 - [(y_{p2} + d)^2 + z_{p2}^2]}$ $p_1 = x_{p1} - \sqrt{c_1^2 - [(y_{p2} + d)^2 + z_{p1}^2]}$	$x_{p2} = \frac{s_2 - s_1^2 + p_2^2}{2(s_1 + p_2)}$ $z_{p2} = -s_4 - \sqrt{c_4^2 - (x_{p2} + s_3)^2}$ $y_{p2} = -d - \sqrt{c_2^2 - z_{p2}^2 - (x_{p2} - p_2)^2}$
$d = 30 \text{ mm}, \delta_{x3} = 100 \text{ mm}, \delta_{y3} = 5 \text{ mm}, \delta_{z3} = 55 \text{ mm},$ $z_{z3} = 120 \text{ mm}$	$s_1 = c_3 \cos \alpha - p_1, s_2 = c_1^2 - c_2^2, s_3 = \delta_{x3} - p_3,$ $s_4 = \delta_{z3} - z_{z3}, s_5 = s_2 - s_1^2 + p_2^2, s_6 = 2(s_1 + p_2)$

3. UPRAVLJANJE

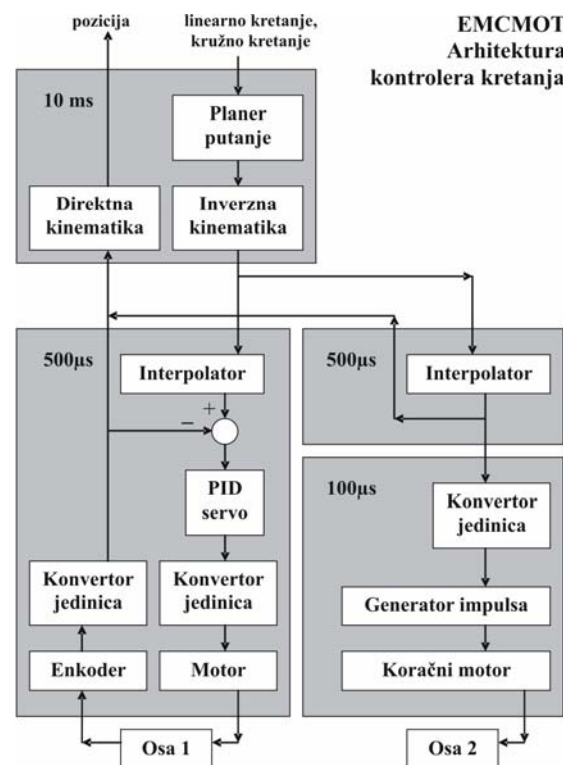
EMC sistemski softver [6] je izrađen na osnovi NIST-ove RCS (*Real-time Control System*) metodologije i programiran je korišćenjem NIST-ove RCS biblioteke. NIST-ova biblioteka olakšava instalaciju EMC sistemskog softvera na više UNIX i Microsoft platformi, obezbeđujući neutralni aplikativni programski interfejs (API) ka resursima operativnog sistema, kao što su deljene memorijske zone, semafori, tajmeri. EMC softver je napisan na C i C++ jezicima i može se instalirati na PC Linux, Windows NT i Sun Solaris operativne sisteme. U slučaju kada se upravlja konkretnim sistemom, tada je neophodno da EMC radi pod real-time verzijom Linux-a, koji obezbeđuje odgovarajući determinizam prilikom izvršavanja programa (tipično 200 ms). EMC sadrži četiri programska modula: kontroler kretanja (EMCMOT), kontroler diskretnih ulaznih/izlaznih signala, kontroler procesa koji ih koordiniše i kolekciju tekstualnih ili grafičkih korisničkih inerfejsa.

3.1 Kontroler kretanja EMCMOT

Kontroler kretanja, pokazan na slici 4, je napisan na jeziku C da bi se omogućilo lakše portovanje na real-time operativne sisteme. Kontrola kretanja uključuje kontrolu uzorkovanja pozicija osa, proračun sledeće tačke na trajektoriji, interpolaciju između uzorkovanih tačaka i proračun izlaznih vrednosti za motore. Kod servo sistema, izlaz se zasniva na PID kompenzacionom algoritmu.

Kontroler kretanja omogućava postavljanje softverskih granica radnih osa, interfejs ka graničnim i referentnim prekidačima radnih osa, PID servo kompenzaciju nultog, prvog i drugog reda, izbor maksimalne greške praćenja, izbor brzine i ubrzanja, pojedinačno pomeranje osa (kontinualno, inkrementalno, apsolutno), programabilnu direktnu i inverznu kinematiku.

Kontroler kretanja je projektovan prilično uopšteno. Inicijalizacione datoteke (sa istom sintaksom kao Microsoft Windos INI datoteke) se koriste za konfigurisanje parametara kao što su broj i tipovi osa (linearne ili obrtne), faktori skaliranja između uređaja u povratnoj sprezi (kao što su obrtni optički enkoderi, optičke merne letve i sl.) odabranih mernih jedinica za ose (npr. mm), pojačanje u servo petlji, trajanje servo i ciklusa planera putanje, kao i drugih sistemskih parametara. Kompleksna kinematika za robote i paralelne mašine se realizuje na jeziku C i povezuje sa postojećim sistemom kako bi se zamenile postojeće funkcije troosne ortogonalne kinematike. Takođe, realizovan na jeziku C, aplikativni programski interfejs (API), između kontrolera kretanja i spoljnog sveta, omogućava integraciju specifičnog korisničkog hardvera u EMC bez potrebe za bilo kakvom modifikacijom kontrolnog koda jezgra softvera.



Slika 4. Arhitektura kontrolera kretanja

Na slici 4 je dat prikaz upravljanja dvema osama, pri čemu osa 1 koristi servo motor, a osa 2 koristi koračni motor. Kod koračnog motora zadata pozicija je ujedno i vraćena vrednost u pozicionoj servo petlji softvera, tako da sistem praktično radi u otvorenoj petlji. Softverska rutina poredi izlaznu poziciju sa brojem akumuliranih impulsa i generiše upravljački koračni impuls ako je razlika veća od polovine impulsa. Kontroler kretanja je program koji se ciklično izvršava. Kada se upravlja stvarnom mašinom, kontroler kretanja zahteva real-time operativni sistem. U našoj instalaciji, koristili smo BDI 4.49 baziran na Debian Linux operativnom sistemu sa real-time ekstenzijom. Ovo daje sistemu determinističku upravljivost do oko 100 milisekundi. Kontroler kretanja koristi RT-Linux FIFO bafere (deljeni memorijski resurs) za prijem insrukcija ili slanje statusnih informacija i poruka o greškama.

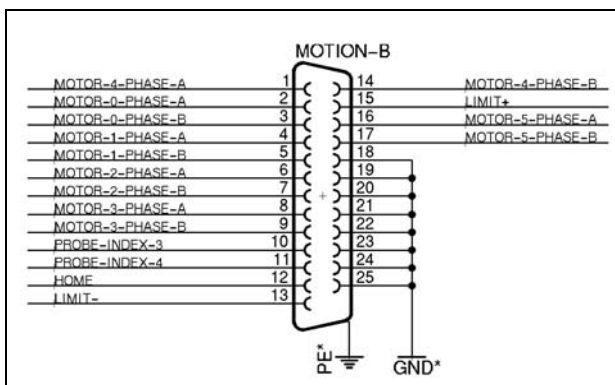
3.2 Kontroler diskretnih ulaznih/izlaznih signala EMCIO

Kontroler diskretnih ulaznih/izlaznih signala (slika 5) je napisan na jeziku C++, korišćenjem NIST-ove RCS biblioteke. Zasnovan je na hijerarhiji C++ klasa izvedenih iz NML_MODULE osnovne klase. Komunikacija između klasa ostvarena je korišćenjem NML-a (*Neutral Messaging Language*). Programski interfejs između diskretnog u/i kontrolera i spoljašnjeg sveta, napisan na jeziku C, omogućava priključenje korisničkog hardvera (sprežnih kartica za pogon mašine) bez potrebe za modifikacijom softvrskog jezgra. Programer mora implementirati funkcije (API) interfejsa za odgovarajuće korisničke kartice.

Diskretni u/i kontroler je implementiran kao hijerarhija kontrolera glavnog kretanja, kontrolera hlađenja i kontrolera pomoćnih uređaja (npr. estop, podmazivanje i sl.).

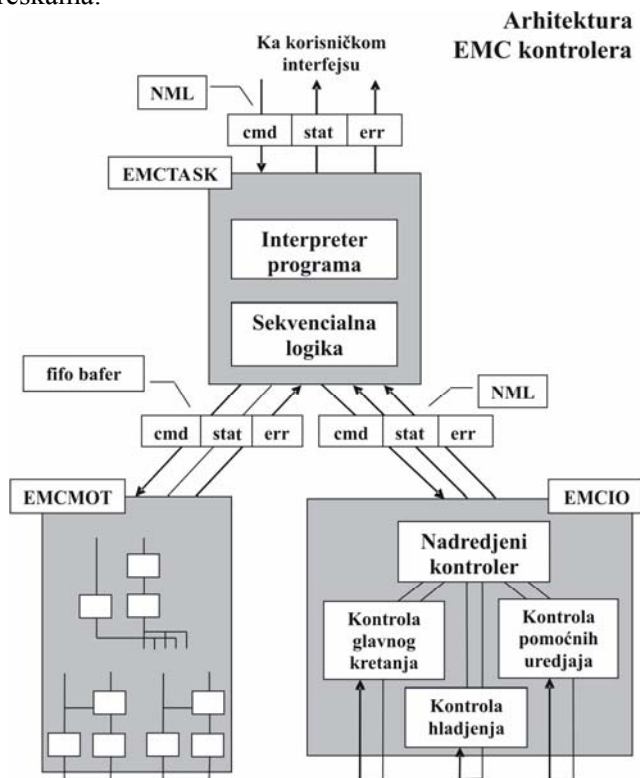
3.3 Kontroler procesa EMCTASK

Kontroler procesa (na vrhu slike 5) je projektovan slično diskretnom u/i kontroleru, korišćenjem NML_MODULE osnovne klase i RCS biblioteke. Odgovoran je za interpretiranje instrukcija G i M koda. Kontroler procesa koordinira aktivnosti kontrolera kretanja i diskretnog u/i kontrolera. Ove aktivnosti su sadržane u NC programu, odnosno nizu instrukcija G i M koda, koje kontroler procesa interpretira kao niz NML poruka i prosleđuje ih kontroleru kretanja ili diskretnom u/i kontroleru u odgovarajućim vremenskim trenutcima.



Slika 6. Veza koračnih motora sa paralelnim portom

povezan model mašine pn101 na paralelni port PC računara. EMC ima podršku za elektrokoračne motore preko paralelnog porta PC računara. Svaki paralelni port ima 12 bita za izlazne podatke i 5 bita za ulazne podatke.



Slika 5. Arhitektura EMC kontrolera

3.4 Eksterni programi

Pod eksternim programima se podrazumevaju programi kao što su grafički korisnički interfejsi (GUI) ili fleksibilni proizvodni sistemi (FSM), koji se koriste za pokretanje EMC-a. Eksterni programi komuniciraju sa EMC-om korišćenjem NML-a, slanjem poruka kao što su uključivanje mašine, prelazak na automatski režim, start programa, isključenje. Korisnički interfejsi mogu slati manuelne poruke, inicirane od operatera, kao što su pomeranje osa mašine u ručnom režimu ili slanje svih osa u referentnu poziciju.

3.5 Povezivanje koračnih motora

Na slici 6 prikazan je način na koji je

Izlazne linije paralelnog porta se koriste za upravljanje koračnim motorima. Prisustvo 12 bita za izlazne podatke znači da se u jednom trenutku može upravljati sa 6 koračnih motora. Ulazne linije paralelnog porta se koriste za priključenje prekidača krajeva hoda i referentnih pozicija. Prisustvo 5 bita za ulazne podatke znači da samo jedna osa može imati oba prekidača za krajeve hoda i jedan za referentnu poziciju. U slučaju troosnih sistema, može se učiniti kompromis tako što se pozitivni granični prekidači mapiraju na jedan ulaz, negativni na drugi, a referentni prekidači na treći ulaz. Postoji mogućnost priključenja još dva paralelna porta na računar, čime se dobija ukupno 36 izlaznih i 15 ulaznih linija.

3.6 Funkcije za inverznu i direktnu kinematiku

Ako se EMC koristi za upravljanje mašinama sa specifičnom kinematikom (roboti, mašine sa paralelnom kinematikom), potrebno je učiniti odgovarajuće izmene u jezgri EMC-ovog softverskog sistema. Izmene se odnose na zamenu standardnih, trivijalnih funkcija inverzne i direktne kinematike funkcijama specifičnim sa datu mašinu. Ovo podrazumeva realizaciju pomenutih funkcija na C programskom jeziku u nekoj korisničkoj datoteci (slika 7), uključenje datoteke u EMC, prevođenje i povezivanje softvera.

```

/*
 parkins.h
 Kinematics for 3 axis parallel
 machine
 Modification history:
 17-Oct-2006 created by Zoran Dimic
 */
#ifndef PARKINS_H
#define PARKINS_H

#define Xhome      210.6186
#define Yhome      (-163.5176)
#define Zhome      11.3364
#define d          30.00
#define c1          200.63
#define c2          205.18
#define c3          205.90
#define c4          100.00
#define pi          3.14159265358979323846
#define alfr        8.37742
#define alfr        ((pi/180.00)*alfr)
#define dx1         (c3*cos(alfr))
#define dy1         d
#define dx3         100.00
#define dy3         5.00
#define dz3         55.00
#define zz3         120.00
#define xr1         265
#define xr2         366
#define xr3         395
static char __attribute__((unused)) parkins_h[ ] =
"$Id: parkins.h,v 1.0 2006/10/17 10:25:42 paul Exp $";
#endif
/* PARKINS_H */

```

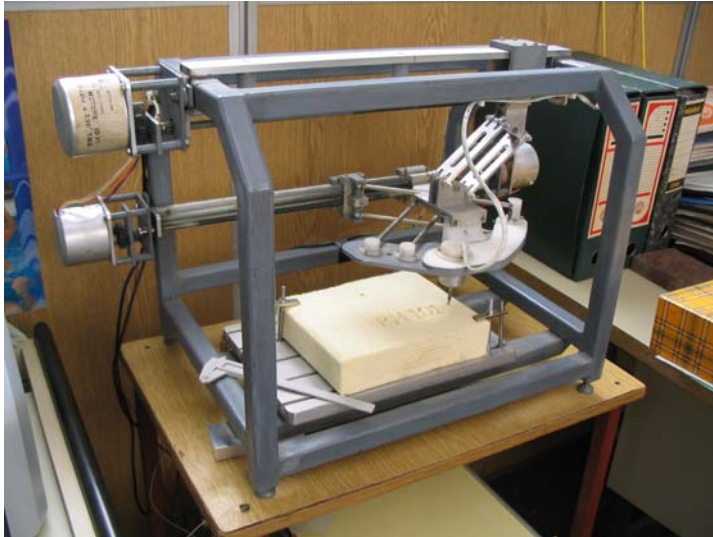
```

/*
 parkins.c
 Kinematics for 3 axis parallel
 machine
 Modification history:
 17-Oct-2006 created by Zoran Dimic
 */
#include "emcmot.h"          /* these decls */
#include "parkins.h"
#include <float.h>
#include <math.h>
/* ident tag */
#ifndef __GNUC__
#ifndef __attribute__
#define __attribute__(x)
#endif
#endif
static char __attribute__((unused)) ident[ ] =
"$Id: parkins.c,v 1.2 2005/08/12 11:37:42 paul Exp $";
int kinematicsForward(const double *joints,
                      EmcPose * pos,
                      const KINEMATICS_FORWARD_FLAGS * fflags,
                      KINEMATICS_INVERSE_FLAGS * iflags)
{
 double s1 = c3 * cos(alfr) - (xr1-(-1)*joints[0]);
 double s2 = c1*c1 - c2*c2;
 double s3 = dx3 - (xr3-(-1)*joints[2]);
 double s4 = dz3 -zz3;
 double s5 = s2 - s1*s1 + (xr2 -
 (-1)*joints[1])*(xr2-(-1)*joints[1]);
 double s6 = 2 * (s1+xr2-(-1)*joints[1]);
 pos->tran.x = s5/s6;
 pos->tran.z = -s4 - sqrt(c4*c4 - (pos->tran.x +
 s3)*(pos->tran.x + s3));
 pos->tran.y = -d - sqrt(c2*c2 -
 pos->tran.z*pos->tran.z -
 (pos->tran.x - (xr2-(-
 1)*joints[1]))*(pos->tran.x -
 (xr2-(-1)*joints[1])));
 pos->tran.x -= Xhome;
 pos->tran.y -= Yhome;
 pos->tran.z -= Zhome;
 pos->a = joints[3];
 pos->b = joints[4];
 pos->c = joints[5];
 return 0;
}
int kinematicsInverse(const EmcPose * pos,
                     double *joints,
                     const KINEMATICS_INVERSE_FLAGS * iflags,
                     KINEMATICS_FORWARD_FLAGS * fflags)
{
 double x,y,z;
 x = pos->tran.x + Xhome;
 y = pos->tran.y + Yhome;
 z = pos->tran.z + Zhome;
 joints[0] = -(xr1 - ((x + dx1) - sqrt(c1*c1 - ((y + d)*(y + d) + z*z)))));
 joints[1] = -(xr2 - (x + sqrt(c2*c2 - ((y + d)*(y + d) + z*z)))));
 joints[2] = -(xr3 - ((x + dx3) + sqrt(c4*c4 - ((z + dz3 - zz3)*(z + dz3
 - zz3)))));
 joints[3] = pos->a;
 joints[4] = pos->b;
 joints[5] = pos->c;

 return 0;
}

```

Slika 7. Datoteke sa definisanim funkcijama inverzne i direktne kinematike



Slika9. Fizički model pn101

upravljanje je neophodna implementacija funkcija inverzne i direktne kinematike. Performanse ovog softvera daju mogućnost da se isti upotrebi i za upravljanje petoosnom hibridnom mašinom tipa glodalice, čije je konfigurisanje takođe realizovano [8]. Sinteza upravljanja ovog modela troosne paralelne mašine bila je planirana i za sticanje iskustava u radu sistema EMC1 [6,7] u okruženju Linux da bi se implementacija na petoosnu mašinu mogla lakše realizovati.

5. LITERATURA

- [1] D. Milutinovic, M. Glavonjic, V. Kvirgic, S. Zivanovic, A New 3-DOF Spatial Parallel Mechanism for Milling Machines with Long X Travel, pp. 345-348, Annals of the Vol54/1, CIRP 2005.
- [2] Glavonjić, M., Živanović, S., Milutinović, D., Troosna paralelna mašina pn101, 31. JUPITER konferencija, 27. simpozijum NU - Roboti - FTS, Zbornik radova, ISBN 86-7083-508-8, str.3.1-3.5, Mašinski fakultet, Beograd, Zlatibor, april 2005.
- [3] Milutinović, D., Glavonjić, M., Kvirgić, M., Živanović, S., Novi paralelni mehanizam za glodalice sa dugačkom X osom, 31. JUPITER konferencija, 27. simpozijum NU - Roboti - FTS, Zbornik radova, ISBN 86-7083-508-8, str.3.6-3.11, Mašinski fakultet, Beograd, Zlatibor, april 2005.
- [4] Živanović, S., Konfigurisanje jedne troosne mašine sa paralelnom kinematikom, 30. Jubilarno savetovanje proizvodnog mašinstva SCG sa međunarodnim učešćem, Zbornik radova, ISBN 86-7776-009-1, str.119-124, Tehnički fakultet i Viša Tehnička škola Čačak, Vrnjačka banja, septembar 2005.
- [5] Dimić, Z., Sistemski softver za upravljanje troosnom glodalicom sa paralelnom kinematikom, 27. simpozijum NU - Roboti - FTS, Zbornik radova, ISBN 86-7083-508-8, str.3.12-3.15, Mašinski fakultet, Beograd, Zlatibor, april 2005
- [6] EMC - Enhanced Machine controller web site - www.linuxcnc.org
- [7] NIST - National Institute of Standards and Technology web site - www.nist.gov
- [8] M.Glavonjić, D. Milutinović, S. Živanović, Z. Dimić, Konfiguracija jedne hibridne petoosne mašine, 33. JUPITER konferencija, 29. simpozijum NU-Roboti-FTS, rad u štampi, Mašinski fakultet, Beograd-Zlatibor, maj 2007.

Saša Živanović, Zoran Dimić

CONTROLLING MODEL OF 3 AXIS PARALLEL KINEMATIC MACHINES BASED ON EMC SYSTEM SOFTWARE

Summary

This paper depicts structure of EMC software system for model of 3-axis parallel kinematics milling machine developed by Faculty of mechanical engineering.

Key words: controlling, parallel kinematics machine (PKM), stepper motors

4. ZAKLJUČAK

Model mašine sa paralelnom kinematikom pn101 je iskorišćen za testiranje i verifikaciju upravljanja na bazi EMC softverskog sistema, koji radi pod Linux operativnim sistemom. U cilju testiranja i verifikacije izvršena je obrada test kontura u obliku kvadrata, krugova i graviranja logoa pn101.

EMC sistemski softver, predstavljen u ovom radu, predstavlja univerzalan *open-source* softverski paket, te je kao takav izuzetno pogodan za konfigurisanje upravljanja, ne samo mašina sa trivijalnom (serijskom) kinematikom, već i specifičnih mašina sa paralelnom kinematikom, za čije