

Eksperimentalni razvojni sistem za mobilno robotsko platformo *

Peter ČEPON, Roman KAMNIK, Jernej KUŽELIČKI, Tadej BAJD, Marko MUNIH

Izvleček: Članek predstavlja konfiguracijo eksperimentalnega razvojnega sistema za mobilno robotsko platformo. Razvojni sistem je sestavljen iz nadzornega računalnika, razvojnega programskega okolja, ciljnega računalnika ter pogonske enote s pogonskim in usmerjevalnim kolesom. Na nadzornem računalniku potekata razvoj in nadzor izvajanja programske opreme, ciljni računalnik pa je namenjen izvajanju razvitih programov v realnem času na ciljni strojni opremi. Sporazumevanje med ciljnim računalnikom in pogonskimi moduli poteka preko vodila CAN. Razvojni sistem je zgrajen na osnovi programov Matlab-Simulink, Stateflow in xPC Target proizvajalca Mathworks. Razvojno okolje omogoča programiranje regulacijskih algoritmov v grafičnem načinu s pomočjo gradnje in povezovanja blokov. Okolje omogoča uporabniško prijazno delo s sistemom preko brezžične povezave, kar olajša eksperimentiranje z mobilno platformo in pohitri razvoj programske opreme.

Ključne besede: mobilna robotska platforma, vodilo CAN, razvojni sistem xPC Target, Stateflow, Simulink,

1 Uvod

Mobilni roboti danes prodirajo na vedno več področij uporabe. Zasledimo jih v industriji, znanosti, športu, vesoljski tehniki, zabavi, filmski industriji itd. Zaradi razširjenosti in napredka se pojavljajo nove zahteve,

kar odpira nova področja raziskav. Večanje avtonomnosti mobilnega robota zahteva vključevanje več senzorjev ter razvoj novih algoritmov vodenja in orientiranja v prostoru.

Cilj uporabe mobilnih robotov je nadomestiti človeka v nevarnem okolju ali mu omogočiti dostop do področij, ki mu zaradi različnih razlogov do sedaj niso bila dostopna. Mobilni roboti se glede na področje delovanja ločijo na vodne, zračne ter kopenske inteligentne naprave. Ne glede na način in okolje, v katerem delujejo, je njihovo bistvo, da so zmožni avtonomnega gibanja v nestrukturiranem okolju. Za tovrstno gibanje v prostoru mora mobilni robot poleg ustreznega pogona uporabljati senzorje, s katerimi zaznava prostor in se na osnovi te informacije ustrezno orientira in načrtuje svojo pot.

Razvoj mobilnega robota in algoritmov vodenja je dolgotrajen postopek, ki zahteva široko področje znanj. Ta vključujejo področja strojništva, elektronike in računalništva. Zaradi tega razvoj običajno poteka preko izdelave prototipa in uporabe razvojnih okolij. Razvojna okolja omogočajo

hiter razvoj algoritmov vodenja in preprosto vključevanje novih komponent. To omogoča eksperimentiranje s sistemom, kar skrajša čas, potreben za razvoj. Obstaja več pristopov k izgradnji prototipa in uporabi razvojnega okolja. Pri razvoju je pred izdelavo realnega sistema priporočljiva uporaba simulacijskih 2D ali 3D modelov mobilnih robotov ter simulacija delovanja v virtualnem računalniškem okolju. Primer tovrstnega orodja je simulacijsko orodje Webots proizvajalca Cyberbotics [6]. Orodje Webots omogoča izgradnjo 3D modela mobilnega robota, na katerega je mogoče namestiti poljubne senzorje in pogonske enote iz knjižnice modulov. Poleg tega orodje vsebuje knjižnico modelov komercialno dosegljivih mobilnih robotov, kot so Pioneer, Khepera, Hemisson, Aibo itd. Za vodenje simuliranega delovanja je uporabljen programski jezik C++. Simulacijsko orodje, ki je prav tako primerno za izgradnjo simulacijskih modulov, je programsko okolje Matlab-Simulink proizvajalca Mathworks, Inc. [1]. Orodje omogoča modeliranje in simuliranje delovanja dinamičnih sistemov z vrsto spremljajočih orodij za analizo in nadzor delovanja.

Peter ČEPON, univ. dipl. inž., doc. dr. Roman KAMNIK, univ. dipl. inž., Univerza v Ljubljani, Fakulteta za elektrotehniko; Dr. Jernej Kuželički, univ. dipl. inž., Iskra Avtoelektrika, d. d., Šempeter pri Gorici Prof. dr. Tadej Bajd, univ. dipl. inž., prof. dr. Marko Munih, univ. dipl. inž., Univerza v Ljubljani, Fakulteta za elektrotehniko

* Delo je bilo sofinancirano v okviru programskega financiranja ARRS, Analiza in sinteza gibanja pri človeku in stroju P2-0228, in Ciljnega raziskovalnega programa MORS in ARRS, Znanje za varnost in mir 2004–2010, projekt Mobilni robotski sistem za izvidniške, raziskovalne in reševalne namene M2-0116.

Na drugi strani je na tržišču moč dobiti kar nekaj realnih mobilnih robotov, ki so namenjeni raziskovalnemu delu in so zasnovani tako, da jih je možno tudi nadgrajevati [3], [4], [5]. Komercialno dosegljivi roboti so različnih velikosti in namembnosti. Med znana podjetja, ki ponujajo komercialne kolesne mobilne robote, sodita podjetji K-TEAM in MobileRobots. Izdelki podjetja K-TEAM so miniaturni mobilni robotki Hemisson, Kheperalll in Koalall. Izdelki podjetja MobileRobots pa so mobilni roboti večjih velikosti, kot so Pioneer 3, PatrolBot in Seekur. Roboti so z ozirom na izbrano konfiguracijo lahko opremljeni z različnimi senzorji za zaznavanje okolice.

Cilj tega dela je predstavitev eksperimentalne mobilne robotske platforme, ki je zasnovana na osnovi pogonskih sistemov, namenjenih premikanju ter usmerjanju električnih viličarjev. Sistem vodenja, ki je zgrajen na osnovi vgrajenega računalnika in xPC Target operacijskega sistema, je zasnovan tako, da omogoča uporabniško prijazno eksperimentiranje in razvoj algoritmov vodenja. V delu je predstavljena konfiguracija razvojnega sistema eksperimentalne robotske mobilne platforme in preliminarni rezultati delovanja.

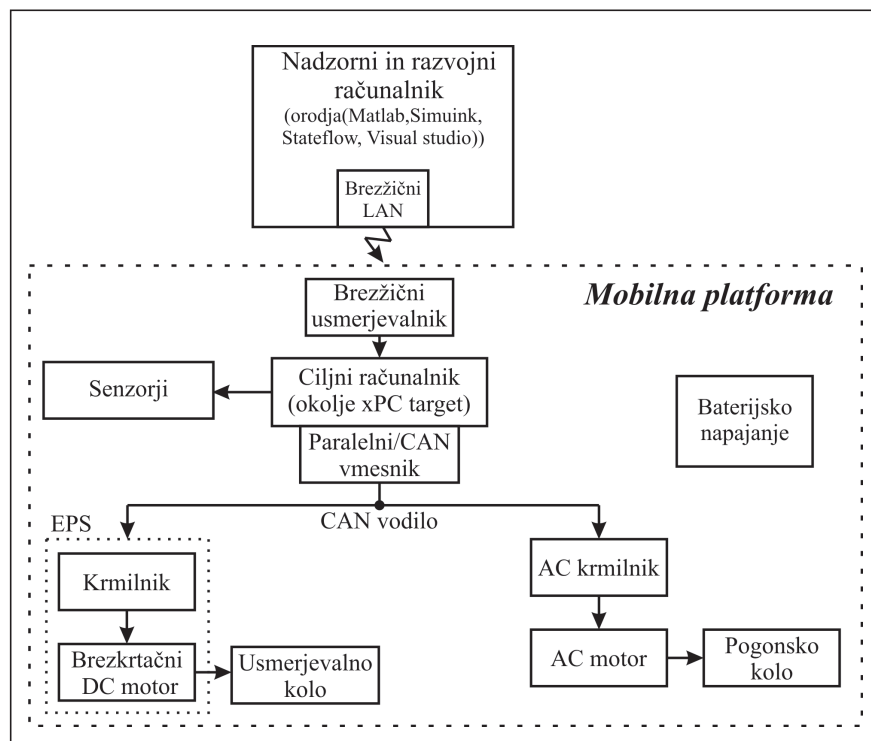
■ 2 Konfiguracija eksperimentalnega razvojnega sistema za robotsko mobilno platformo

Eksperimentalna razvojna platforma je zgrajena na osnovi modulov, ki so v blokovni shemi prikazani na *sliki 1*, na fotografijah pa na *sliki 2*.

Pogonski sistem platforme tvorijo pogonski moduli, ki se uporabljajo za pogon in usmerjanje električnih viličarjev. Pogonski modul je proizvod podjetja Iskra Avtoelektrika. Za sam pogon je uporabljen trifazni asinhronski AC-motor, ki ima vgrajena inkrementalni kodirnik in zavoro. Fotografijo asinhronskega motorja, reduktorja in pogonskega kolesa prikazuje *slika 2 b*. Regulacija vrtenja AC-motorja je izvedena s pomočjo AC-krmilnika, ki je zasnovan na DSP-procesorju in omogoča krmiljenje v koordinatnem sistemu polja. Krmilnik

motorja z nadrejeno enoto komunicira preko komunikacijskega protokola CANopen.

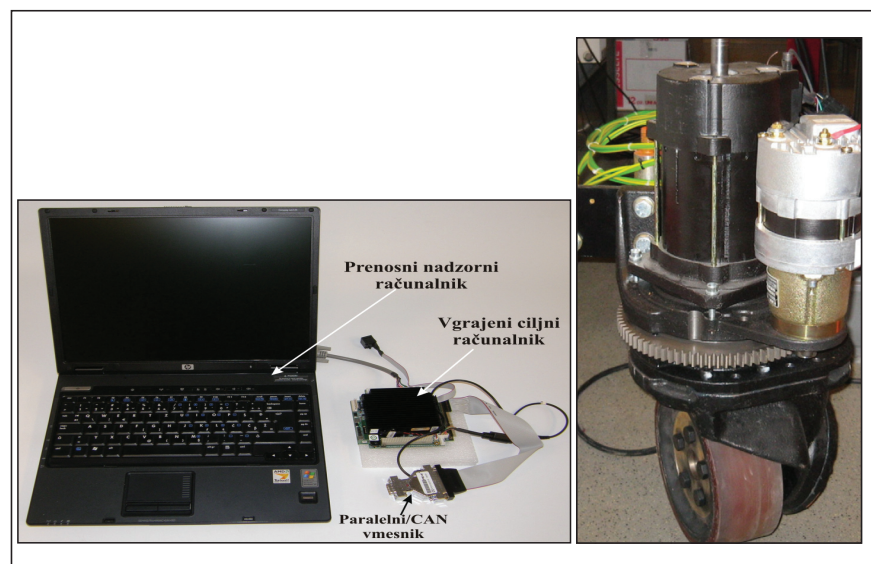
Za nadzor in krmiljenje AC-krmilnika in sistema EPS je uporabljen ciljani računalnik PC/104 PCM-



Slika 1. Konfiguracija sistema vodenja mobilne robotske platforme

Poleg pogonskega kolesa je na *sliki 2 b* prikazan tudi sistem EPS (ang. Electric Power Steering), namenjen usmerjanju pogonskega kolesa. Sistem EPS tvorijo brezkrtačni DC-motor, reduktor in krmilnik, ki je vgrajen v samo ohišje motorja. Krmilnik EPS z nadrejenim računalnikom prav tako komunicira preko vodila CAN.

3380 dimenzij 108 mm x 115 mm proizvajalca Advantech, ki je konfiguriran na osnovi strojne opreme vgrajenega računalnika. Delovanje v realnem času je bilo doseženo z uporabo operacijskega sistema xPC Target proizvajalca Mathworks [8]. Orodje omogoča izvajanje programskih aplikacij v realnem času,



Slika 2. Fotografija a - sistema vodenja, b - pogonskega sistema

ki so razvite v grafičnem okolju Matlab-Simulink. Programiranje v grafičnem načinu na osnovi povezovanja blokov poteka v celoti na razvojnem računalniku. Ko je shema vodenja razvita, je prevedena v kodo za izvajanje. Koda za izvajanje je zatem preko brezžične TCP/IP-povezave naložena na ciljni računalnik, ki se nahaja na mobilni platformi. Sporazumevanje med krmilnikom motorjev in ciljnim računalnikom poteka s pomočjo protokola CANopen preko vodila CAN [2]. CAN (ang. Controller Area Network) je serijsko vodilo, ki je osnovano na ISO OSI sedemnivojskem sistemu. Za uspešno komuniciranje so uporabljeni le trije nivoji. Ti so: prvi fizični, drugi podatkovni in sedmi aplikacijski. Najvišja hitrost prenosa podatkov znaša 1 Mbit/s. Zaradi robustnosti in možnosti povezav naprav, ki delujejo v realnem času, je vodilo CAN pogosto v uporabi v industrijskih okoljih. Protokol CANopen nadgrajuje fizični in podatkovni nivo v aplikacijskem nivoju. Standard CANopen definira pogoje pravilnega delovanja, pošiljanja in naslavljanja sporočil ter uporabo 11-bitnega ali 29-bitnega identifikatorja. Komunikacija preko vodila CAN poteka tako, da se vsaka priključena naprava prijavi na vodilo s svojim prepoznavnim naslovom. Zatem gospodar vodila (ang. master) pošlje na vodilo sporočilo, s katerim ga postavi v operacijsko stanje. Ko je vodilo v operacijskem stanju, gospodar vodila pošilja na vodilo sporočilo dolžine do največ osem bajtov skupaj z 11-bitnim identifikatorjem, ki določa, kateri napravi je sporočilo namenjeno. Naslovljena naprava se odzove s povratnim sporočilom [2]. Za povezavo ciljnega računalnika in vodila CAN je bil uporabljen vmesnik CAN, ki je priključen na paralelna vrata ciljnega računalnika.

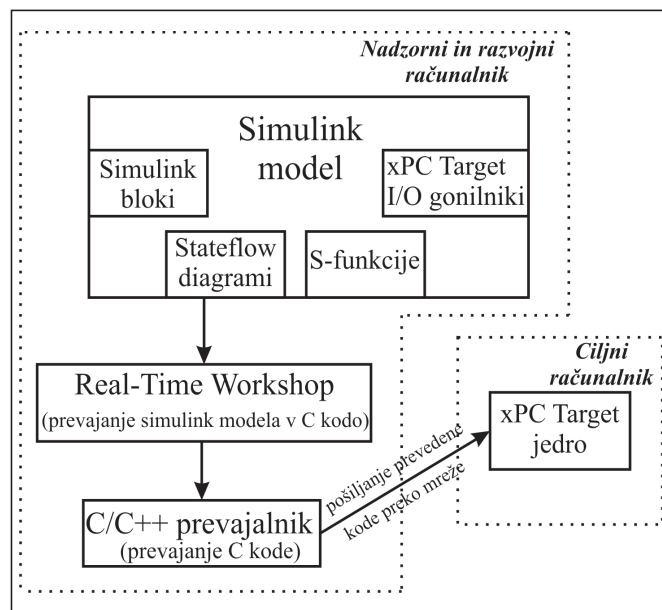
Na vgrajeni računalnik je možno priključiti različne periferne naprave. Te so lahko priključene preko paralelnega vodila, vodila CAN, vodila firewire ali vodila PC/104.

Za baterijsko napajanje mobilne robotske platforme sta uporabljeni dve akumulatorski bateriji tipa Dry-

fit proizvajalca TAB, d. o. o., ki sta med seboj povezani zaporedno, kar omogoči napetost 24 V za napajanje pogonov.

3 Programska arhitektura razvojnega sistema

Razvoj programske opreme poteka na razvojnem računalniku v grafičnem okolju. Ko je programska oprema razvita in prevedena v izvršljivo kodo, je preko brezžične povezave naložena na ciljni računalnik. Ta na osnovi operacijskega sistema xPC Target v realnem času opravlja vodenje in nadzor pogojskega in usmerjevalnega kolesa.



Slika 3. Potek razvoja programa s pomočjo razvojnega sistema

Programska arhitektura razvojnega sistema je prikazana na *sliki 3*, kjer vidimo, da poteka programiranje v grafičnem načinu v okolju Simulink, kjer z uporabo blokov iz knjižnic Simulinka, orodja diagramov Stateflow, blokov xPC Target z I/O-gonilniki in lastnih funkcij, zgradimo funkcionalni model Simulink. Ta je s pomočjo orodja Real-Time Workshop nato preveden v C-kodo, ki je s pomočjo prevajalnika C/C++ prevedena v kodo za izvrševanje. Ta koda je preko brezžične TCP/IP-povezave posredovana ciljnemu računalniku, na katerem teče xPC Target jedro za izvrševanje aplikacije v realnem

času. Ko je izvršljiva koda naložena na ciljni računalnik, sta možna zagon aplikacije v realnem času ter spremljanje poteka signalov na ciljnim in razvojnem računalniku. Delovanje sistema xPC Target je osnovano na principu izvorno-ciljnega (ang. host-target) delovanja, kar omogoča oddaljen nadzor nad delovanjem ciljnega računalnika [8]. Na razvojnem računalniku je tako mogoče na daljavo preko brezžične povezave nadzirati potek signalov ter spreminjati in iskati optimalne vrednosti parametrov v realnem času.

3.1 Prikaz primera programa

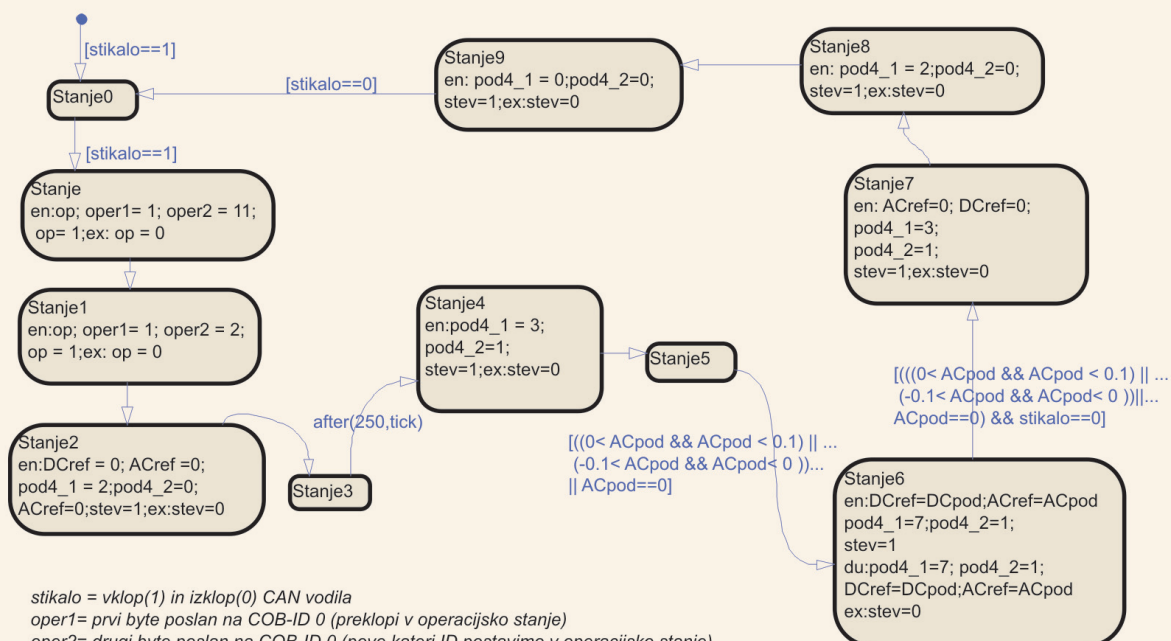
Prikaz programa v grafični obliki, ki je zgrajen z uporabo knjižnice blokov Simulink in orodja diagramov poteka Stateflow, je prikazan na *sliki 4*.

Okolje Simulink je namenjeno modeliranju in simulaciji dinamičnih zveznih sistemov. Simulacijska shema je zgrajena grafično s povezovanjem blokov iz knjižnic Simulinka. Orodje Stateflow pa je namenjeno

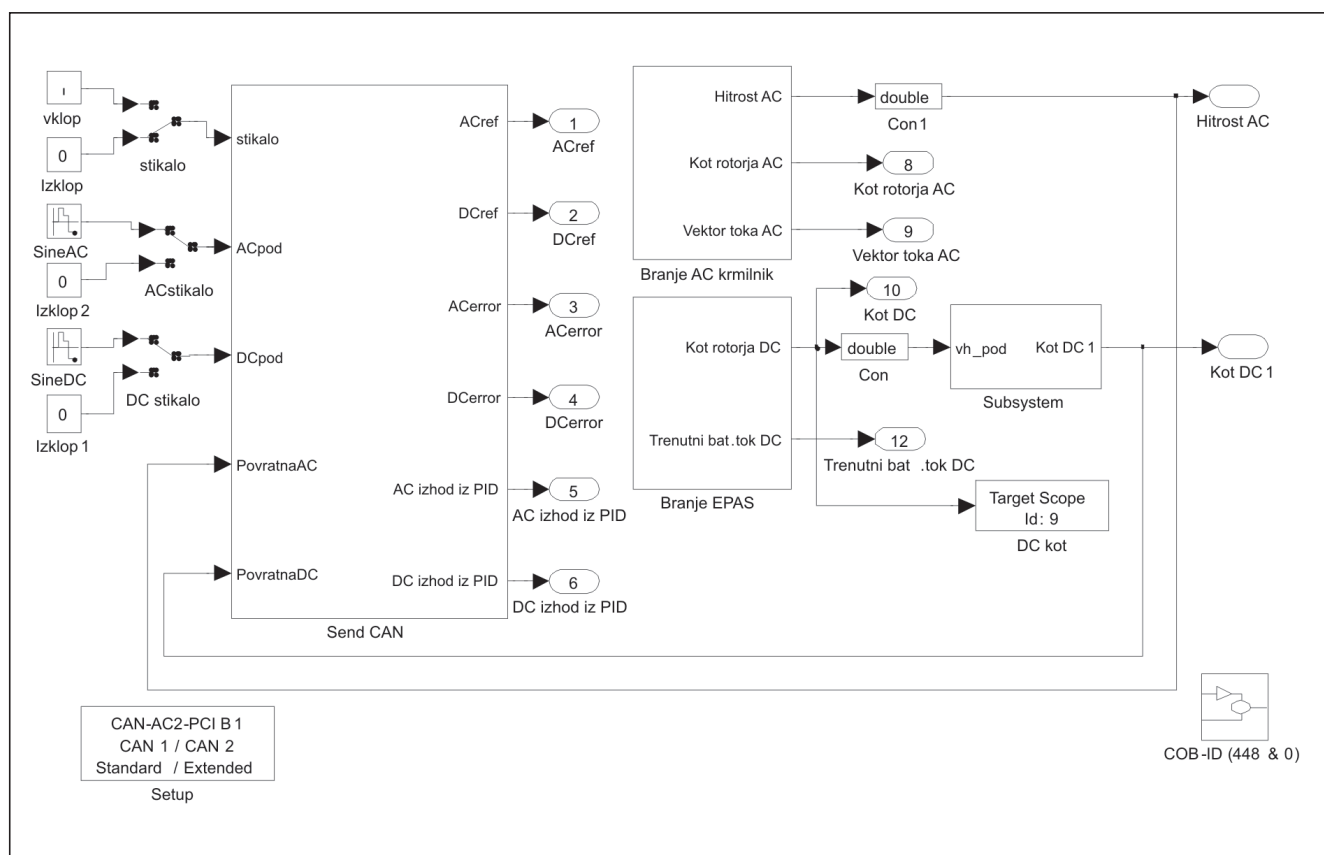
modeliranju in simulaciji diskretnih sistemov in dogodkov. Omogoča grafično zasnovo bločnih diagramov, pri katerih s pomočjo določenih dogodkov prehajamo med diskretnimi stanji sistema.

Na *sliki 4* je prikazan primer programa hitrostne regulacije asinhronskega pogonskega motorja in pozicijske regulacije brezkrtačnega usmerjevalnega motorja pri sinusnem referenčnem signalu. Rezultati vodenja obeh motorjev so prikazani na *sliki 5*. Graf na *sliki 5 a* predstavlja referenčni in dejanski položaj usmerjevalnega motorja, graf na *sliki 5 b* pa referenčno in dejansko hitrost asinhronskega pogonskega motorja.

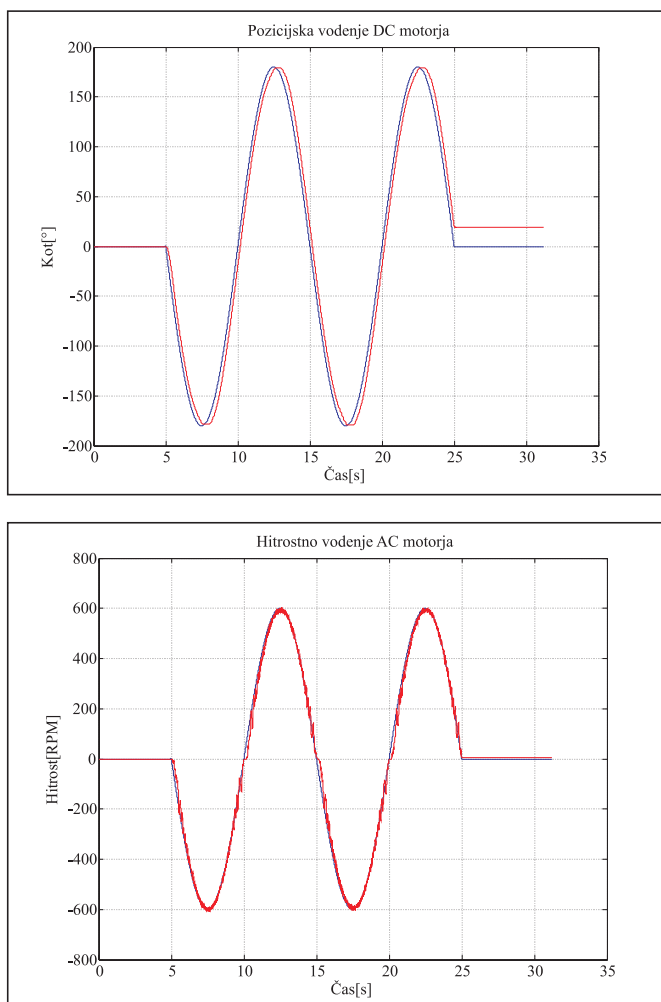
Potek vklop in posiljanja na CAN vodilo



stikalo = vklop(1) in izklop(0) CAN vodila
 oper1= prvi byte poslan na COB-ID 0 (preklopi v operacijsko stanje)
 oper2= drugi byte poslan na COB-ID 0 (pove kateri ID postavimo v operacijsko stanje)
 op = prozimo CAN vodilo za posiljanje na COB-ID 0
 ACref=referenca za AC krmilnik (izhod)
 DCref=referenca za EPAS (izhod)
 ACpod=referencni podatek za AC krmilnik (vhod)
 DCpod=referencni podatek za EPAS (vhod)
 pod4_1=prvih osem bitov flags word(vklop stikala in vklop močnostne stopnje)
 pod4_2=drugi osem bitov flags word(vklop močnostne stopnje DC)
 stev=prozimo CAN vodilo za posiljanje referenc na CAN vodilo
 after počaka 1s in nato začne s posiljanjem



Slika 4. Prikaz a – poteka diagramov Stateflow, b – uporabe grafičnega povezovanja blokov Simulink



Slika 5. a – rezultat pozicijske regulacije brezkrtačnega DC-motorja, b – rezultat hitrostne regulacije AC- asinhronskega motorja

4 Zaključek

V delu je prikazana konfiguracija eksperimentalne mobilne robotske platforme,

programa v grafičnem okolju, uporaba programskih struktur in orodij okolja Matlab, nadzor delovanja, spreminjanje parametrov, spremljanje in zajemanje signalov ciljnega sistema na daljavo ter preprosto vključevanje novih naprav.

ki omogoča hiter in uporabniku prijazen razvoj algoritmov vodenja. Platforma je zasnovana na osnovi komercialno dosegljivih pogonskih modulov ter sistema vodenja, ki ga predstavljata dva računalnika, medsebojno brezžično povezana. Prvi računalnik, ki je lahko prenosni, je namenjen razvoju algoritmov vodenja in nadzoru delovanja, drugi, ki je lahko vgrajeni računalnik, nameščen na ciljnem sistemu, pa izvajanju algoritmov v realnem času. Prednosti, ki jih omogoča tak sistem, so prenosljivost, uporabniško prijazna zasnova

Literatura

- [1] M. Bongiovanni, An experimental framework for rapid prototyping of mobile robot controllers, International Conference Towards Autonomous Robotic Systems (TAROS), pp. 21–27, London 2005.
- [2] M. Farsi, M. B. M. Barbosa, CANopen implementation: applications to industrial networks, Research Studies Press Ltd. 2000.
- [3] G. Dudek, M. Jenkin, Computational Principles of Mobile Robotics, Cambridge University Press, Cambridge 2000.
- [4] R. Siegwart, I. R. Nourbakhsh, Introduction to Autonomous Mobile Robots, MIT Press, Cambridge 2004.
- [5] R. R. Murphy, Introduction to AI Robotics, MIT Press, Cambridge 2000.
- [6] M. Oliver, WebotsTM: Professional Mobile Robot Simulation International Journal of Advanced Robotic Systems, Vol. 1, No. 1, pp. 39–42, 2004.
- [7] J. Borenstein, H. R. Everett, L. Feng, Where Am I? Sensors and Methods for Mobile Robot Positioning, University of Michigan, Michigan 1996.
- [8] xPC Target User's Guide, Matlab & Simulink, The MathWorks, Inc., 2007.

Experimental Development System for Mobile Robotic Platform

Abstract: In this paper an experimental development system for mobile robotic platform is presented. The development system incorporates the main controller, the drive unit with drive and steering wheel, the drive controller and development environment. The software development and supervision of execution is performed on the host main computer. The target embedded controller running xPC Target real time operating system is used for execution of the developed program in real time on a target system. The target controller communicates with the drive units via CAN communication. The development system is based on Mathworks Matlab tools Simulink, Stateflow and xPC Target. This configuration allows the development of control algorithm in graphical mode by building and connecting functional blocks. In this way the development system is built providing user friendly graphical software development environment, optimal tuning of parameters, acquisition and logging of signals, and easy incorporation of new devices.

Keywords: mobile robotic platform, CAN bus, xPC Target, experimental development system, Stateflow, Simulink,