

NAPREDNI MEHATRONSKI PRISTOP K NAČRTOVANJU ODPRTEGA KRMILJA STROJEV

Aleš Hace, Aleš Polič, Karel Jezernik

Inštitut za robotiko
Fakulteta za elektrotehniko, računalništvo in informatiko
Univerza v Mariboru
Smetanova ulica 17, 2000 Maribor
tel: 02 2207307
fax: 02 2207315
email: ales.hace@uni-mb.si

Povzetek: Članek posega na področje proizvodne avtomatizacije v okviru fleksibilnih proizvodnih sistemov. Predstavljen je nov mehatronski pristop k snovanju krmilja strojev, ki uvaja integriran in interdisciplinarni način celovitega načrtovanja strojev, kjer so mehanski, elektronski in informacijski vidiki proizvodnega sistema obravnavani sinergijsko. V članku je opisan krmilnik rezalnega stroja, ki je zasnovan na distribuirani arhitekturi krmilja in z novimi pristopi k snovanju krmilnih funkcij. Zgrajen je na arhitekturi osebnega računalnika z odprtim operacijskim sistemom realnega časa QNX. Široko razširjeno grafično okolje Microsoft Windows služi kot vmesnik človek-stroj. Krmilnik in uporabniški vmesnik sta povezana z Ethernet TCP/IP protokolom, in se tako lahko povezujeta tudi na pisarniški nivo omrežja avtomatizirane proizvodnje. Za opis logičnih krmilnih funkcij je predstavljena nova metoda, ki temelji na matričnem opisu Petrijevih mrež. Omogoča simulacijo in analizo sistema s čimer se lahko bistveno skrajša razvojni cikel izdelave krmilja stroja, obenem pa je kompatibilna s standardnimi načini za programiranje logičnih krmilnikov. Krmilje je tako bolj zanesljivo in učinkovito ter prijazno do uporabnika.

Ključne besede: proizvodna avtomatizacija, CNC/PLC, operacijski sistemi realnega časa, aplikacija sistemov vodenja, dogodkovni sistemi, Petrijeve mreže

1. Uvod

Sodobni stroji so tehnološko zelo zahtevni, visoko avtomatizirani, v prihodnosti pa se bodo morali bolje prilagoditi proizvodnji-na-zahtevo. Zato se v zadnjem času raziskave in razvoj na področju proizvodnih sistemov usmerjajo k takšnim rešitvam, ki bodo izboljšale rekonfigurabilnost, fleksibilnost, hitrost in točnost, z namenom skrajšanja razvojnih ciklov izdelkov in odzivnih časov na nove zahteve trga. Za realizacijo teh ciljev je potrebno razviti:

- inteligentne proizvodne postroje z novimi mehatronskimi moduli, ki se bodo sposobni prilagoditi na procesno-orientirane proizvodne zahteve
- rekonfigurabilne inteligentne krmilne arhitekture in sisteme za krmiljenje takih postrojov
- metode in "plug-in" enote za komunikacijo med postroji
- inteligentne vmesnike človek-stroj za enostavnejše upravljanje postrojov
- mrežna orodja za porazdeljeno proizvodno avtomatizacijo
- inženirska orodja za konfiguriranje krmilnih sistemov, simulacijska orodja in okolja, komunikacijske algoritme, protokole in jezike

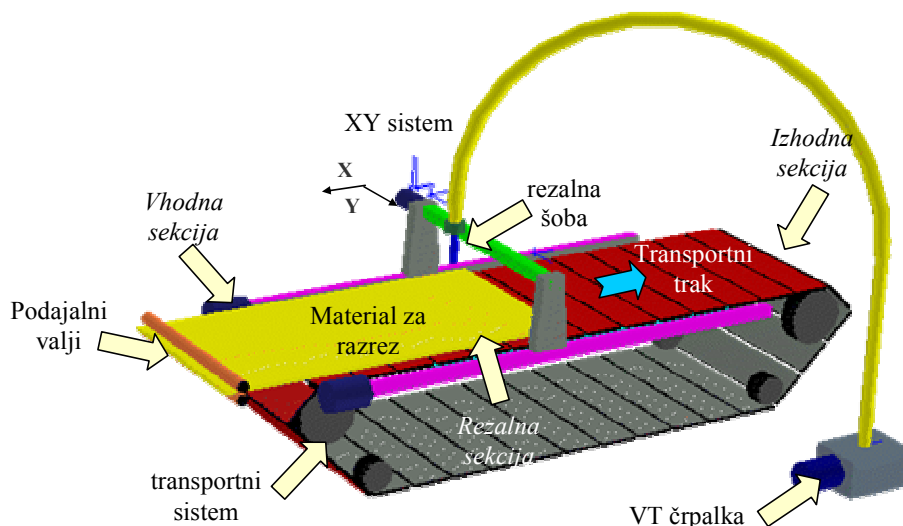
Naštete komponente lahko razdelimo na komunikacijske module, module za vodenje procesov in krmilne module ter module z dodatnimi procesnimi funkcijami. Dodatne procesne funkcije lahko vključujejo tako strojne kot programske rešitve za realizacijo večfunkcijskih strojev, ki se bodo znali prilagoditi na zahtevano nalogo. Krmilni moduli vsebujejo odločitveno logiko, usmerjanje transporta materiala, polizdelkov in izdelkov, nadzor in regulacijo procesov, proizvodne informacije ter dodatno še komunikacijske funkcije. Tako se lahko stroj prilagaja na spremembe proizvodnih parametrov, pri čemer se proizvodna strategija lahko izbira tudi v skladu z drugimi stroji. Odprta distribuirana

arhitektura vodenja mora torej podpirati funkcije vodenja strojev, komunikacije med stroji in optimizacije proizvodnje. Uporabniški vmesnik pa mora operaterju omogočati nadzor procesa, logistično podporo, možnost simulacije proizvodnih procesov, intervencije med izvajanjem procesa ter e-asistenco in e-vzdrževanje. Nova shema proizvodne avtomatizacije mora biti konfigurirana, izvedena in testirana v industrijskih aplikacijah.

Mehatronika uvaja nov, integriran in interdisciplinarni pristop k celovitemu načrtovanju izdelkov, kjer so mehanski, elektronski in informacijski vidiki proizvodnega sistema obravnavani sinergijsko. Predstavlja osnovo za nove inteligentno krmiljene naprave z napredno funkcionalnostjo, višjo zanesljivostjo in boljšo učinkovitostjo. Članek se metodološko osredotoča na praktično inženirsko platformo za načrtovanje, konfiguriranje, izvedbo in delovanje inteligentnih proizvodnih sistemov. Je aplikativno orientiran in uvaja mehatronski pristop k načrtovanju CNC krmilnika [5] in logičnih krmilnih funkcij [12]. K načrtovanju krmilja stroja pristopa kot sledi: najprej je predstavljena tehnologija tipičnega sodobnega rezalnega stroja in princip delovanja ter glavne strojne funkcije. Sledi zasnova grafičnega uporabniškega vmesnika za učinkoviti nadzor in upravljanje stroja, določena je arhitektura aparaturne opreme za krmilje stroja, na koncu pa so razviti so algoritmi vodenja stroja in predstavljeni nekateri rezultati.

2. Rezalni stroj

Nova tehnologija rezanja z vodnim curkom je v primerjavi z obstoječimi tehnologijami veliko bolj fleksibilna in zato bolj prilagodljiva sodobnim tržnim razmeram, omogoča pa tudi hitro izdelavo prototipov in razvoj popolnoma novih izdelkov. Obravnavan rezalni stroj je namenjen razrezu usnja, tekstila in sintetičnih materialov, uporaba te tehnologije pa je mogoča tudi na drugih področjih kot je kamnoseštvo, steklarstvo, lesarstvo, papirništvo, gumarska industrija, obdelava pločevine, prehrabena industrija, itd.

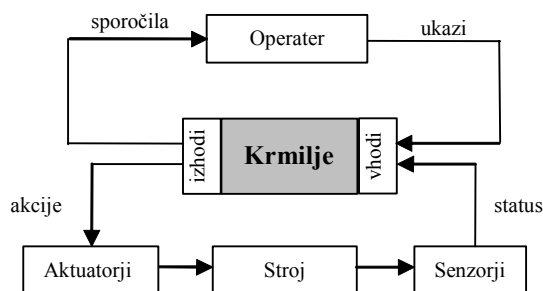


Slika 1: Rezalni stroj

Stroj sestavljajo trije poglavitni deli: transportni sistem, XY sistem in visokotlačni rezalni sistem. Shematsko je stroj prikazan na sliki 1. Transportni sistem podaja material v rezalno sekcijo in ga fiksira v času rezanja. Sestavljata ga transportni trak in podajalni valji. Podajalni valji dozirajo material za razrez na transportni trak. Transportni trak transportira material v rezalno sekcijo, odrezke pa iz rezalne v izhodno sekcijo, kjer jih operator pobere in odloži na paleto. Taka zasnova transportnega sistema omogoča kontinuirano podajanje materiala v rezalno sekcijo. Omogočeno je tudi rezanje delov, ki so daljši od rezalne sekcije stroja. XY sistem predstavljata medsebojno pravokotno nameščeni osi, ki ju poganjata visokozmogljiva servomotorja. X-os sestavljata dve vzporedno ležeči vreteni, ki ju poganjata ločena servomotorja. Vreteno y-osi leži na obeh vretenih x-osi. Rezalna šoba je nameščena na y-osi, z visokotlačno hidravlično črpalko pa je povezana preko visokofleksibilne cevi. Takšna razporeditev koordinatnih osi omogoča optimalen dostop do rezanega materiala v vsaki poziciji rezalne šobe.

Osnovni režim obratovanja stroja je avtomatski cikel. Funkciji transporta materiala in rezanja se izvedeta sekvenčno znotraj posameznega cikla. Med rezanjem stoji operator pri izhodni sekciji in pobira odrezke predhodnega rezanja. Po zaključku rezanja in po spraznitvi izhodne sekcije, se izvede nova faza transporta materiala.

3. Krmilnik



Slika 2: Splošna krmilna shema

Krmilnik stroja skupaj s konstrukcijsko zasnovo stroja mora zagotavljati želeno funkcionalnost, hkrati pa mora omogočiti varno in učinkovito upravljanje stroja (slika 2). Krmilje tako vsebuje več funkcij, ki jih lahko v grobem razdelimo na tehnološke funkcije, funkcije upravljanja in pomožne funkcije. Med tehnološke funkcije štejemo tiste, ki so povezane z elementarnim namenom stroja kot je vrtanje, rezanje, varjenje ali zgolj pozicioniranje. Med pomožne funkcije uvrščamo tiste splošnega namena kot je nadzor delovanja stroja, varnost ljudi in opreme, diagnostika napak in odpravljanje motenj v delovanju. Funkcije upravljanja so usmerjene k operaterju v smislu učinkovite uporabe in upravljanja stroja. Namenjene so komunikaciji človek-stroj in so navadno izvedene na tekstovnih ali grafičnih operacijskih panelih z dodanimi krmilnimi gumbi. Operacijski panel je priključen na programabilni logični krmilnik (Programmable Logic Controller - PLC) ali računalniški numerični krmilnik (Computer Numerical Control - CNC), ki izvajata algoritme vodenja stroja. Ti se lahko razdelijo na vrednostno- in dogodkovno- orientirane algoritme. Vrednostno-orientirani algoritmi (numerično krmilje) kot je pozicioniranje so navadno izvedeni na CNC. Dogodkovno-orientirani algoritmi (logično krmilje), ki vključujejo funkcije kot so start/stop, so navadno vgrajeni na PLC. Implementacija tako zahtevne in obsežne strukture krmilnih funkcij je mogoča le s kompleksno krmilniško arhitekturo. Pri gradnji zanesljivih in zmogljivih strojev z zahtevno tehnologijo, visoko stopnjo avtonomnosti

ter prijaznim upravljanjem, je potrebno uporabiti integriran pristop k načrtovanju krmilja. V svetu še vedno prevladujejo sposobni, a zaprti CNC/PLC sistemi, vendar pa se v zadnjih letih v praksi vse bolj uveljavljajo tudi krmilja, ki uporabljajo tehnologijo osebnih računalnikov (Personal Computer - PC). Programska oprema za nadzorni vmesnik človek-stroj (Human-Machine Interface - HMI), za upravljanje stroja, logične krmilne procedure in regulacijske algoritme, je zgrajena modularno ter na voljo za sprejemljivo ceno. Sistemski integratorji lahko enostavno kombinirajo strojne in programske komponente za gradnjo učinkovitih, standardnih in odprtih rešitev krmilnih sistemov, s tem pa se izognejo uporabi dragih zaprtih sistemov z omejeno funkcionalnostjo. Pri načrtovanju odprtih sistemov, ki uporabljajo standardne strojne in programske rešitve, so manjši stroški, takšen sistem je razširljiv in nadgradljiv ter omogoča izboljšave v prihodnosti. Koristi vključujejo ne le nižje stroške, pač pa tudi boljšo funkcionalnost sistema in komunikacijsko povezavo s podjetjem. Razen tega je mogoče skrajšati razvojni cikel, z uporabo simulacijskih orodij in orodij za diagnosticiranje pa se skrajša tudi čas zastojev v proizvodnji, s čimer se poveča produktivnost in zmanjšajo vzdrževalni stroški.

3.1. Vmesnik človek-stroj

Grafični uporabniški vmesnik (Graphical User Interface - GUI), ki je predstavljen na sliki 3, je glavni del HMI na rezalnem stroju. GUI je implementiran na ekranu osebnega računalnika krmilnega sistema in združuje vse prikaze ter funkcije, ki so potrebne za upravljanje stroja.



Slika 3: Grafični uporabniški vmesnik

Namen GUI je, da zagotovi uporabniku prijazen nadzor nad delovanjem stroja. Uporabniku posreduje vse koristne strojne informacije na ekranu kontrolne plošče kot so:

- splošne informacije, ki vključujejo režim in način obratovanja stroja (avtomatsko/ročno, izvajanje/ustavitev), ime in stanje aktualnih tehnoloških procedur, ter ostale informacije o stanju stroja
- informacije o hitrosti rezalne glave, ki vključuje želeno in dejansko vrednost
- podatki o posameznih oseh kot so položaj in druge statusne informacije
- podatki o programu (ime programa, aktualni blok programa, status programa)
- podatki o orodju (curek odprt / zaprt)
- podatki o stanju procedur na PLC (avtomatsko obratovanje stroja zagnano / v teku / ustavljeno)
- podatki o dodatnih oz. pomožnih napravah na stroju (visokotlačna črpalka, napajanje stroja, varnostni krog)
- razne druge informacije (izklop v sili, stanje na mreži, briši napako)

- razna sporočila, ki vključujejo več nivojev alarmov in druga sporočila operaterju, ki se prikažejo sproti kot nastanejo ustrezni dogodki na stroju; za ponovni zagon po napaki na stroju mora operater alarme zbrisati ročno.

GUI omogoča nadzor, programiranje in konfiguriranje stroja, podpira pa tudi večfunkcijske programske krmilne gumbе. V osrednjem področju ekrana kontrolne plošče vključuje polje, v katerem izriše geometrijo kontur rezanja. Ko se rezalni program naloži, se načrtovana kontura rezanja prikaže v celoti. Posamezni segmenti, ki se režejo, in tisti, ki so že bili odrezani, so označeni z različno barvo. Hkrati se na načrtovani konturi s križcem označuje dejanski položaj rezalne glave. Položaj rezalne glave se osvežuje s hitrim ciklom osveževanja.

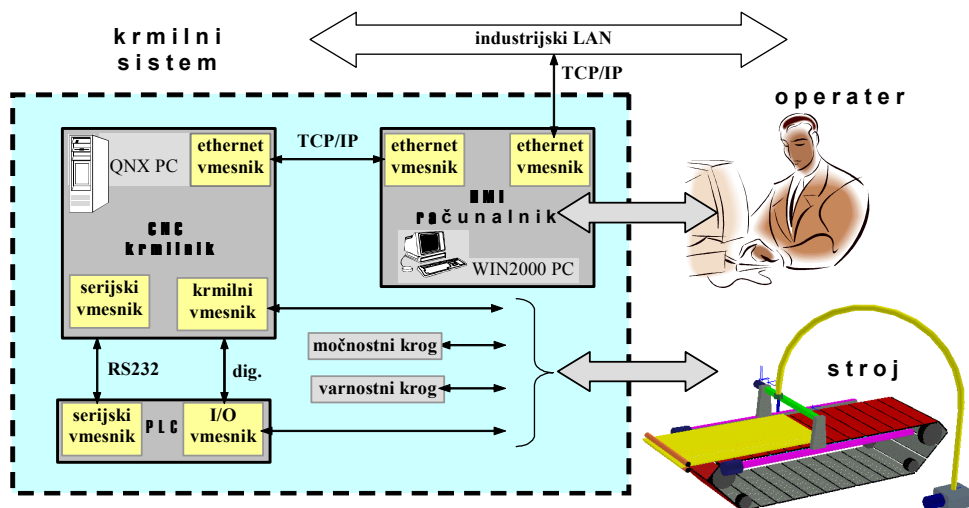
Struktura oken uporabniškega vmesnika je krmiljena interaktivno glede na trenutno stanje stroja oziroma procedure. V primeru spremembe režima obratovanja GUI samodejno ustrezno zamenja okna. GUI vsebuje različna okna, nekatera med njimi se pojavljajo na vmesniku dinamično. Okno procedure se pojavi na ekranu samodejno, ko se procedura na stroju zažene. Okno pokaže vse korake procedure med izvajanjem. Operater je tako ves čas natančno voden skozi proceduro, prikažejo pa se tudi navodila za morebitne ročne posege operaterja. Takšen pristop naredi delovanje stroja transparentno, omogoča pa tudi hitro odkrivanje napak in ponoven zagon stroja v primeru zastoja.

Uporabniško prijazno upravljanje olajša določanje napak na stroju, s tem pa je tudi vzdrževanje lažje. Identifikacija napak je mogoča s pomočjo sporočil, ki jih GUI posreduje operaterju. Sporočila o napakah so prikazana na lahko razumljiv in korekten način. Medtem GUI nadaljuje spremljanje in beleženje vseh novih alarmov. Informacije o stanju stroja se zajemajo ciklično, s periodo osveževanja krajšo od sekunde, kar omogoča sprotno posredovanje sporočil in primerno hiter odziv na stroju.

3.2. Strojna oprema krmilnika

V kolikor želimo vgraditi CNC/PLC funkcionalnost na osebni računalnik (PC), mora ta delovati v realnem času. Še več, za izvedbo servo regulacijskih algoritmov se mora PC odzivati v strogem realnem času. Odziv PC mora biti tako logično in povrh tega še časovno pravilen – odzivati se mora pravočasno in deterministično. PC operacijski sistem, ki omogoča delovanje v realnem času, je QNX [14]. QNX operacijski sistem je primeren tudi za uporabo pri aplikacijah v robotiki in avtomatizaciji [4], o čemer so že poročali številni avtorji [1], [2], [7], [15], saj ima fleksibilno arhitekturo, ki zagotavlja robustno in zanesljivo implementacijo vgrajenih sistemov. Zelo natančni programski časovniki omogočajo delovanje v strogem realnem času na poceni in razširjeni strojni opremi. Tako je lahko z uporabo sodobnih enoprocesorskih osebnih računalnikov, ki se dobijo na tržišču, mogoč razvoj takšnih sistemov, ki imajo v istem računalniku vgrajene krmilne algoritme, mrežno podporo in grafični uporabniški vmesnik. S takšno arhitekturo krmilnika je mogoče zamenjati tradicionalno večprocesorsko arhitekturo "gostitelj/DSP", ki se pogosto uporablja v aplikacijah sistemov vodenja. Takšen sistem je drag in ni tako fleksibilen, programska oprema pa je zelo kompleksna. Prednosti enoprocesorskega sistema tako vključujejo nižje stroške in manjšo kompleksnost sistema, večjo fleksibilnost in nadgradljivost. V predstavljeni aplikaciji upoštevamo še združljivost z obstoječo CAD/CAM programsko opremo razvito v Microsoft Windows okolju, zato je krmilni računalniški sistem zasnovan z dvema osebnima računalnikoma (slika 4):

- HMI računalnik z Microsoft Windows okoljem na katerem teče grafični uporabniški vmesnik in komunikacijska podpora za povezavo z obstoječo tovarniško CAD/CAM programsko opremo .
- CNC krmilnik je osebni računalnik z QNX operacijskim sistemom za deterministično izvajanje CNC/PLC funkcij v strogem realnem času.



Slika 4: Krmilni sistem

CNC/PLC programski paket je vgrajen na osebнем računalniku in obsega regulacijske algoritme za XY sistem, osnovne nadzorne rutine za vodni curek, krmilne procedure in komunikacijske module. Končna instalacija sistema je načrtovana v smislu vgrajenih sistemov brez tipkovnice, miške in ekrana. Računalnik z nameščenim Microsoft Windows okoljem ima vgrajeno opremo z grafičnim uporabniškim vmesnikom, gosti pa tudi CAD/CAM programsko opremo z zunanjo Ethernet povezavo na tovarniško lokalno omrežje. Na uporabniški vmesnik so priključene še klasične vhodno/izhodne enote. Računalnika sta med seboj povezana s standardnimi Fast Ethernet vmesniškimi karticami. CNC krmilni računalnik je opremljen z vhodno/izhodno vmesniško kartico za aplikacije servo regulacijskih algoritmov, ki predstavlja krmilni vmesnik. Krmilni vmesnik povezuje CNC krmilnik s strojem. Priključen je na servo regulatorje motorjev in na zunanji PLC, ki je dodan za razširitev sistema z dislocirano inteligentno I/O periferijo. Glavne naloge PLC so krmiljenje transportnega sistema in visokotlačne črpalke. RS-232 povezava omogoča prenos večje množice podatkov med CNC krmilnikom in zunanjim PLC. Močnostni in varnostni algoritmi krmilja so izvedeni z relejsko logiko.

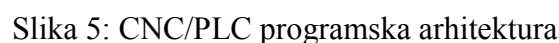
3.3. Programska oprema

3.3.1 *Računalniški numerični krmilnik*

CNC izvaja različne naloge potrebne za vodenje stroja: osnovne krmilne logične procedure, časovno kritične položajne regulacijske zanke, pospeševanje/zaviranje, NC generiranje poti, nalaganje in interpretiranje programov, vmesnik za programiranje stroja, grafični uporabniški vmesnik, upravljanje z datotečnim sistemom, procesiranje podatkov in mrežna podpora. S pomočjo zunanjega PLC skrbi za usklajeno delovanje XY sistema, transportnega sistema in visokotlačne črpalke. Vse uporabniško pomembne informacije, ki so povezane s strojem, morajo biti prenesene do uporabnika. Tako CNC posreduje uporabniškemu vmesniku informacije o položaju in hitrosti rezalne glave, o izvajanju programa rezanja, razna sporočila in alarme. Nadalje krmilnik vzdržuje informacijski most, ki povezuje zunanji PLC z uporabniškim vmesnikom, s čimer je omogočen prenos podatkov o delovanju transportnega sistema in visokotlačne črpalke. Da je mogoče izvajati vse te naloge, mora programska oprema CNC krmilnika delovati v pogojih realnega časa, pri čemer so pogoji strogega realnega časa povezani s servo regulacijskimi algoritmi, medtem ko se informacijska podpora uporabniškega vmesnika lahko izvaja v mehkem realnem času. Programski procesni model odjemalec-strežnik omogoča izvedbo takšne arhitekture procesov, ki ustreza naštetim zahtevam. Ta model je v QNX mogoče zelo enostavno uporabiti s pomočjo komunikacijske sheme, kjer je strežni proces blokiran medtem, ko čaka na sporočilo od odjemalca. Predlagana arhitektura procesov krmilnika je strukturirana kot kaže slika 5. Procesi so načrtovani tako, da izvajajo naslednje naloge:

- **Strežnik krmilnega vmesnika** (hardware server) implementira nizko nivojske procedure za dostop do I/O vmesniške kartice in tako zagotavlja zajemanje digitalnih in analognih signalov.
- **Interpolator, XY regulator, Krmilna logika** (path planner, setpoint buffer, XY controller, control logic) skrbijo za izvedbo uporabniških nalog: interpolator na osnovi

- **Podatkovni strežnik** (data server) omogoča ostalim procesom uporabo skupnih podatkov, ki so shranjeni v skupnem deljenem pomnilniku.
- **Uporabniški vmesnik** (user interface) izvaja uporabniške funkcije kot je posredovanje informacij operaterju, ročni vnos krmilnih komand, vnos in interpretacija uporabniškega programa, na podlagi katerega se krmili interpolator, XY regulator in krmilna logika, ter nadzor sistema; podpira vhodno/izhodne naprave kot so tipkovnica, miška in monitor, nadalje pa tudi mrežni protokol aplikacijskega nivoja, ki je zgrajen na TCP/IP skladu.

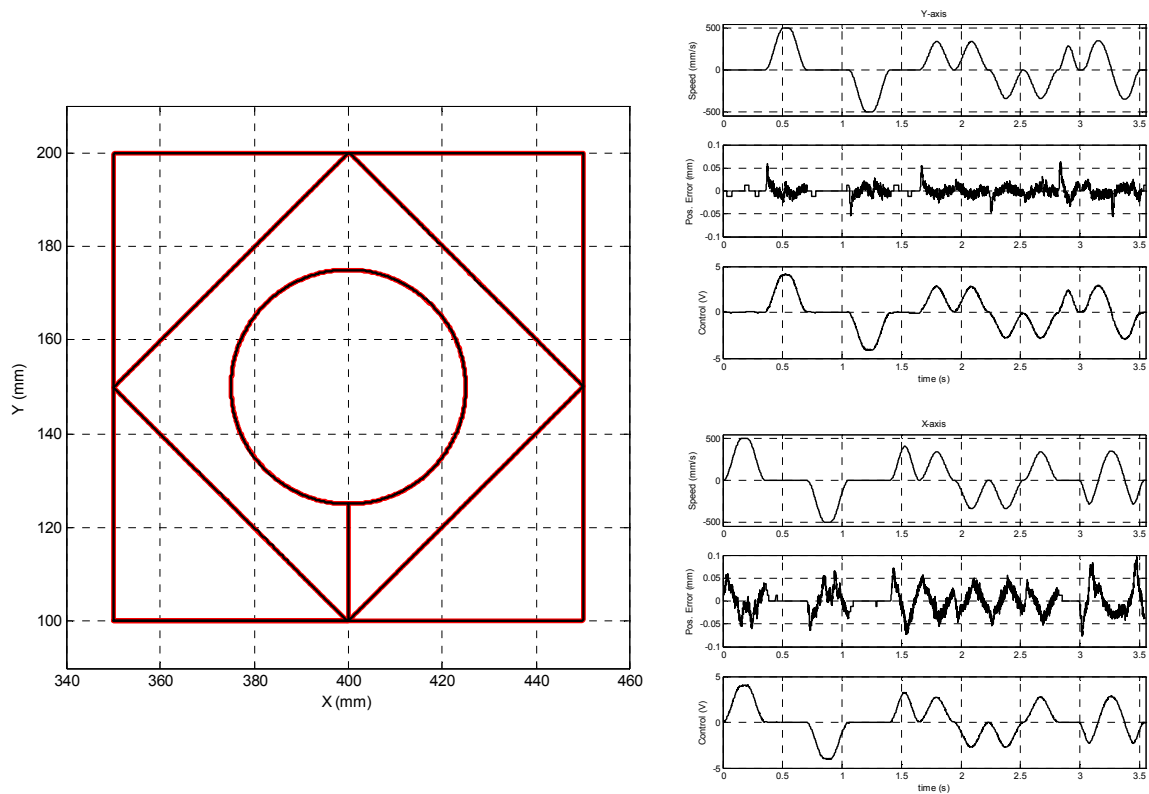


- **TCP/IP odjemalec/strežnik** (TCP/IP client/server) skrbita za mrežno Ethernet povezavo z HMI računalnikom. TCP/IP protokol omogoča povezavo več oddaljenih računalnikov in hkrati relativen varen prenos podatkov. Protokol aplikacijskega nivoja, ki določa pomen komunikacijskega sporočila, je v osnovi ukazni interpreter. Podatki se izmenjujejo na zahtevo strežnika na HMI računalniku ciklično. Tako se zagotovi kontinuirano osveževanje podatkov na uporabniškem vmesniku. Za primerno kratko periodo in dovolj hitro odzivnost sistema na posredovanje operaterja pa mora biti informacijski paket čim manjši glede na pasovno širino komunikacijskega kanala. Zato je k periodičnemu informacijskemu paketu, ki vsebuje osnovne informacije o statusu stroja, dodana še dogodkovno krmiljena informacijska shema. Ta zajema krajša sporočila, ki imajo pogosto višjo prioriteto in vsebujejo informacijo o drugih dogodkih na stroju (npr.: napake, alarmi), in morajo biti samodejno obdelana ali posredovana operaterju.
- **Pošiljatelj in sprejemnik serijskega komunikacijskega kanala (serial comm. sender/receiver)** komunicirata z zunanjim PLC preko serijske RS232 povezave. PLC odgovarja na zahteve pošiljatelja, in se odziva z ustreznimi sporočili, ki jih pošilja sprejemniku. Sporočila so sestavljena iz spremenljivk s predločenim pomenom, da so lahko čim krajša, saj je hitrost prenosa po serijskem kanalu relativno nizka.

Eksperimentalni rezultati

CNC krmilnik je bil izveden na osebnem računalniku z AMD ATHLON 800MHz Thunderbird procesorjem na GA-7IXE4 matični plošči z 64MB RAM. Na HMI računalnik je bil priključen z Fast Ethernet (100Mbps) povezavo, z RS232 serijsko povezavo pa na zunanji PLC. Na tem sistemu so se lahko regulacijski algoritmi položajne zanke izvajali s kratko periodo 200 μ s, tipična nastavitev je bila 1ms, kar je omogočalo visoko točnost položajnega vodenja pri visokih rezalnih hitrostih do 1m/s. Procedure krmilne logike so se izvajale s

periodo 4ms, kar je dovolj hitro za relativno počasne procese, ki jih nadzirajo logične krmilne funkcije. Podatki na uporabniškem vmesniku so se ciklično osveževali s periodo 0.5s. Na to je vplivala relativno počasna serijska povezava z zunanjim PLC, ki je s 19200 Baudov prenosne hitrosti podatkov omejila komunikacijsko periodo na 200ms.



Slika 6: Regulacijski rezultati

Na sliki 6 so prikazani rezultati položajne regulacije rezalnega sistema. Izhod položajnega regulatorja na CNC krmilniku je bila hitrostna komanda servomotorjem. Regulacijski algoritem je bil v tem primeru računan v 250 μ s časovnih intervalih. Na diagramu levo je prikazana testna kontura v XY ravnini, desna diagrama pa trajektorije posameznih osi z želeno hitrostjo zgoraj, sledilnim položajnim pogreškom v sredini in regulacijskim izhodnim signalom spodaj.

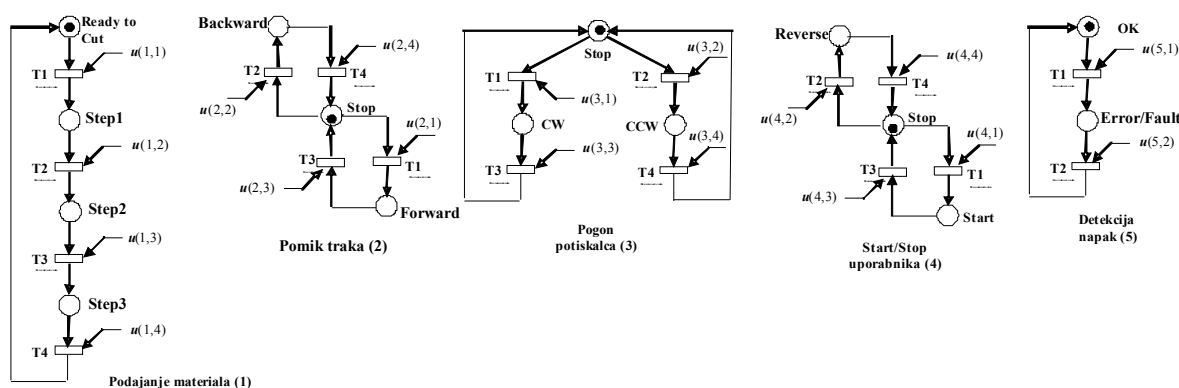
3.3.2 *Načrtovanje programirljive krmilne logike*

Osnovni namen PLC je izvajanje dogodkovno-orientiranih logičnih funkcij kot so vklop/izklop stroja, izbor avtomatskega cikla oziroma ročnega upravljanja, start/stop cikla, nadzor aktuatorjev, detekcija napak in spremljanje obratovalnih pogojev. Klasično se k programiranju krmilnih funkcij pristopi na osnovi časovnega diagrama izvajanja tehnoloških funkcij stroja in na osnovi krmilne sheme elektroožičenja [9]. S tem so podane tehnološke funkcije stroja in arhitektura njegovega krmilja. Implementacija funkcij za upravljanje, nadzor in varovanje stroja ter funkcij za samodejno zaustavljanje, diagnozo in odpravo napak, pa je prepuščena presoji in izkušnjam programerja. Zaganjanje takšnih strojev se ponavadi lahko prične šele po končanju vseh strojnih in elektromontažnih del. Krmilna logika se v obliki programa naloži na PLC, nakar se prične s testiranjem osnovnih funkcij v ročnem režimu obratovanja, sledi testiranje avtomatskih funkcij in na koncu še optimizacija delovanja stroja.

Za skrajšanje razvojnega cikla načrtovanja krmilnih funkcij je bil razvit nov pristop k obravnavi dogodkovnih sistemov [11], [12], [13], ki omogoča načrtovanje in analizo pravilnosti zasnovanih krmilnih funkcij ter simulacijo zmogljivosti ciljnega sistema še pred izvedbo vključujoč mehanske, elektronske in informacijske komponente. Na ta način je mogoče preveriti konstrukcijske in funkcionalne rešitve ciljnega sistema že v zgodnji fazi razvojnega cikla, kar bistveno skrajša trajanje in zmanjša stroške projekta v primerjavi z ugotavljanjem funkcionalnih pomanjkljivosti že zgrajenega ciljnega sistema. Metoda temelji na predpostavki, da logične krmilne funkcije predstavimo kot dogodkovne sisteme, za katere so značilna specifična stanja. Sistem v vsakem trenutku zavzame le eno diskretno stanje, prehod v novo stanje pa se zgodi ob pojavu dogodka pri izpolnjenih pogojih. Stanja krmilnih funkcij so pogosto povezana s stanji aktuatorjev, ki jih funkcija krmili ali pa s stanji delovnega procesa. Prehod krmilne funkcije med stanji se zgodi ob pojavu dogodka, npr.

pritisek na gumb, aktiviranje senzorja ali doseganje mejne vrednosti. Novo stanje, ki ga bo funkcija zavzela po pojavu dogodka, je odvisno od trenutnega stanja funkcije in dogodka ter od trenutnega stanja preostalih funkcij na stroju in od vhodnih krmilnih signalov.

Petrijeve mreže (Petri Nets - PN) [5], [10] so med pogostejše uporabljenimi orodji za obravnavo dogodkovnih sistemov [16], [17]. Ena od njihovih prednosti je, da jih je mogoče hitro pretvoriti v enega od standardiziranih načinov za programiranje PLC-jev [3]. Petrijeve mreže so v osnovi bipartitni graf, ki premore dve vrsti vozlišč, ponavadi predstavljenimi s krogi in pravokotniki. Stanja funkcij prikazujejo krogi, trenutno aktivno stanje označuje žeton. Pravokotniki simbolizirajo dogodke, ki povzročijo prehod med stanji. Dopustne prehode med stanji funkcije ponazarjajo usmerjene linije. Dodatna linija, povezana v pravokotnik, predstavlja zunanje pogoje, ki vplivajo na prehod med stanji ob pojavu dogodka [9]. Nekaj primerov PN, ki predstavljajo krmilne funkcije rezalnega stroja s slike 1, prikazuje slika 7.



Slika 7: Petrijeve mreže krmilnih funkcij procedure transporta materiala

Strukturo grafa PN lahko predstavimo s pomočjo incidenčne matrike M [10]. Stolpci matrike M predstavljajo stanja krmilne funkcije, medtem ko vrstice predstavljajo dogodke. Komponente matrike $M(i,j)$ lahko zavzamejo vrednosti:

$$\mathbf{M}(i, j) = \begin{cases} +1, & j\text{-stanje je posledica } i\text{-dogodka} \\ 0, & j\text{-stanje ni povezano z } i\text{-dogodkom} \\ -1, & j\text{-stanje je pogoj za } i\text{-dogodek} \end{cases} \quad (1)$$

Incidenčne matrike PN s slike 7 predstavlja (2).

$$\mathbf{M}_1 = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 1 \end{bmatrix} \mathbf{M}_2 = \begin{bmatrix} 1 & -1 & 0 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \mathbf{M}_3 = \begin{bmatrix} 1 & -1 & 0 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \mathbf{M}_4 = \begin{bmatrix} 1 & -1 & 0 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \mathbf{M}_5 = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (2)$$

Stanje sistema po k dogodkih predstavlja vektor $\mathbf{m}(k)$. Njegove komponente so razpoložljiva stanja funkcije, pri čemer velja:

$$\mathbf{m}(k) = \begin{cases} 1, & \text{stanje aktivno} \\ 0, & \text{stanje ni aktivno} \end{cases} \quad (3)$$

Vektorje stanja PN s slike 7 predstavlja (4). Ime komponente ustreza prostoru PN s slike 7.

$$\mathbf{m}_1 = \begin{bmatrix} RdyToCut \\ Step1 \\ Step2 \\ Step3 \end{bmatrix}, \mathbf{m}_2 = \begin{bmatrix} Stop \\ Forward \\ Backward \end{bmatrix}, \mathbf{m}_3 = \begin{bmatrix} Stop \\ CW \\ CCW \end{bmatrix}, \mathbf{m}_4 = \begin{bmatrix} Stop \\ Start \\ Reverse \end{bmatrix}, \mathbf{m}_5 = \begin{bmatrix} OK \\ Error / Fault \end{bmatrix} \quad (4)$$

Podobno kot trenutno stanje krmilne funkcije $\mathbf{m}(k)$ so s pomočjo vektorjev zapisani tudi dogodki in zunanji pogoji, ki vplivajo na sistem. Komponente vektorja $\mathbf{x}(k)$ predstavljajo dogodke za krmilno funkcijo, komponente vektorja $\mathbf{u}(k)$ pa predstavljajo zunanje pogoje za pojav dogodkov.

$$\mathbf{x}(k) = \begin{cases} 1, & \text{dogodek je} \\ 0, & \text{dogodka ni} \end{cases}, \mathbf{u}(k) = \begin{cases} 1, & \text{pogoj je izpolnjen} \\ 0, & \text{pogoj ni izpolnjen} \end{cases} \quad (5)$$

Vektorje dogodkov za PN s slike 7 predstavlja (6), zunanje pogoje zanje pa opisuje (7).

$$\mathbf{x}_1 = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix}, \mathbf{x}_2 = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix}, \mathbf{x}_3 = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix}, \mathbf{x}_4 = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix}, \mathbf{x}_5 = \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} \quad (6)$$

$$\mathbf{u}_1 = \begin{bmatrix} 1 < time < 3 \\ time > 3 \\ pos > 5 \\ time > 7 \end{bmatrix}, \mathbf{u}_2 = \begin{bmatrix} m_{3,2} \\ m_{3,3} \\ m_{3,1} \\ m_{3,1} \end{bmatrix}, \mathbf{u}_3 = \begin{bmatrix} m_{1,3} \& m_{4,2} \& pos < 10 \\ m_{1,3} \& m_{4,3} \& pos > 0 \\ m_{4,1} \mid m_{1,3} \& pos > 5 \\ time > 7 \end{bmatrix}, \mathbf{u}_4 = \begin{bmatrix} m_{5,1} \\ 6 < time < 6.5 \\ m_{5,2} \\ time > 7 \end{bmatrix}, \mathbf{u}_5 = \begin{bmatrix} 3.8 < time < 5.5 \\ m_{4,3} \end{bmatrix} \quad (7)$$

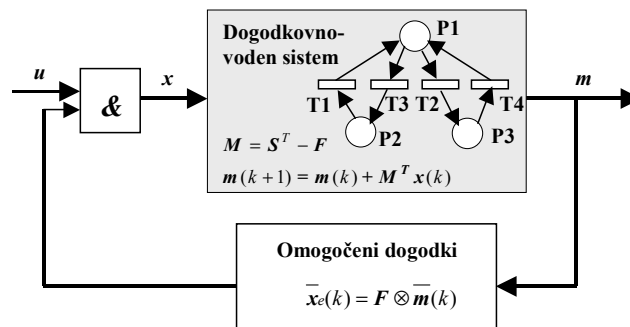
Predstavljen pristop omogoča formalen opis prehajanja stanj logičnih krmilnih funkcij. Formalizem je sestavljen iz dveh delov: najprej se glede na trenutno stanje krmilne funkcije $\mathbf{m}(k)$ določeni vsi dogodki $\mathbf{x}_e(k)$, ki jih trenutno stanje omogoča [8], [16]. Iz nabora omogočenih dogodkov je nato s pomočjo pogojev $\mathbf{u}(k)$ izbran en dogodek $\mathbf{x}(k)$, kot kaže enačba (8), ki bo spremenil stanje krmilne funkcije [12].

$$\mathbf{x}(k) = (\bar{\mathbf{F}} \oplus \mathbf{m}(k)) \& \mathbf{u}_{ec}(k) \quad (8)$$

Matrika $\mathbf{F}$ je del incidenčne matrike $\mathbf{M}$, ki predstavlja pogoje za prehajanje v novo stanje. Novo stanje krmilne funkcije $\mathbf{m}(k+1)$ je nato izračunano z enačbo (9) glede na trenutno stanje $\mathbf{m}(k)$ in dogodek določen z vektorjem $\mathbf{x}(k)$.

$$\mathbf{m}(k+1) = \mathbf{m}(k) + \mathbf{M}^T \mathbf{x}(k) \quad (9)$$

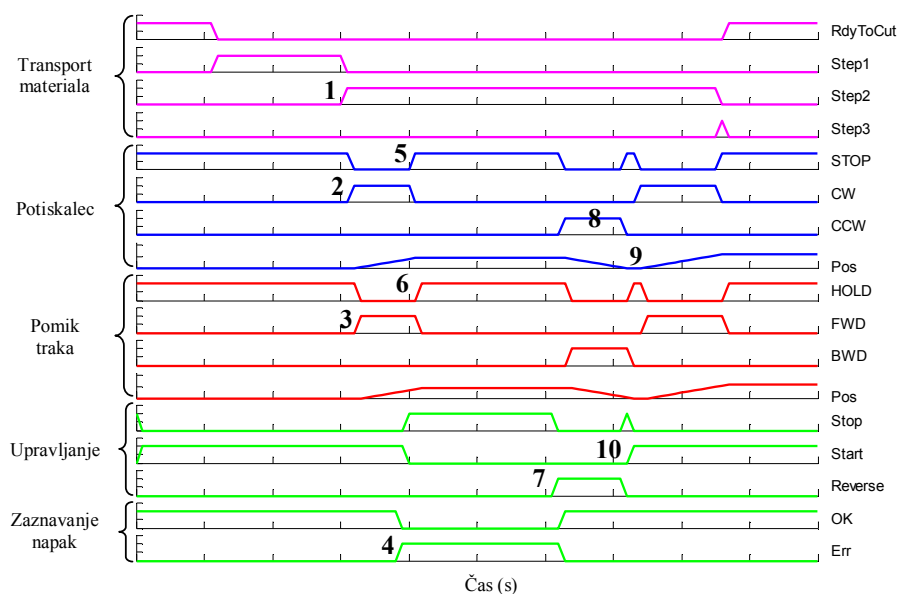
Znak $\oplus$ predstavlja Boolov skalarni produkt. Računa se kot navadni skalarni produkt, le da sta množenje in seštevanje zamenjana z logičnima ALI in IN. Shematično je formalizem prikazan na sliki 8.



Slika 8: Modularni logični krmilnik

Modularni logični krmilnik [12] omogoča algebrski način določanja prehajanja stanj dogodkovno vodenega sistema glede na strukturo sistema $\mathbf{M}$, trenutno stanje $\mathbf{m}(k)$, trenutne dogodke $\mathbf{x}(k)$ in trenutne zunanje pogoje $\mathbf{u}(k)$.

Rezultati



Slika 9: Časovni diagram stanj krmilnih funkcij

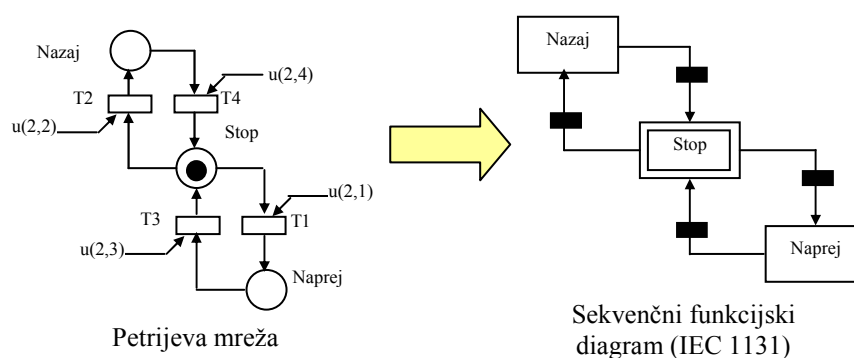
Ideja matričnega modularnega logičnega krmilnika je ilustrirana na primeru procesa transporta materiala, ki zajema naslednje krmilne funkcije:

- **Transport materiala** predstavlja osnovno tehnološko funkcijo. Sestoji iz izhodiščnega stanja »Ready to Cut« in treh sekvenčno izvršenih korakov.
- **Potiskalec** je uporabljen za pomikanje transportnega traka. Implementiran je kot elektromotor s tremi specifičnimi stanji: (i) vrtenje v urni smeri (pomik naprej), (ii) vrtenje v protiurni smeri (pomik nazaj) in (iii) mirovanje.
- **Pomik traka** predstavlja funkcijo pomikanja transportnega traka. Funkcija lahko zavzame tri stanja: transportni trak se pomika naprej, nazaj ali pa miruje.
- **Upravljanje** je uporabniško orientirana funkcija za upravljanje procesa.
- **Zaznavanje napak** je nadzorna funkcija za zagotavljanje varnosti uporabnika in stroja.

Z uporabo incidenčnih matrik M_1 do M_5 iz enačbe (2), enačbe izbora dogodkov (8) in enačbo prehajanja stanj (9), je obravnavan sistem mogoče simulirati s programskim paketom Matlab.

Namen simulacije je analiza delovanja zasnovanega logičnega krmilja in odziva logičnega krmilja na kritične dogodke. Slika 9 predstavlja diagram prehajanja stanj posameznih krmilnih funkcij med procesom transporta materiala v stroj. Diagram opisuje potek naslednjih krmilnih funkcij: *Transport materiala*, *Potiskalec*, *Pomik traka*, *Upravljanje* in *Zaznavanje napak*. Spremenljivki *Pos* iz krmilnih funkcij *Potiskalec* in *Pomik traka* sta zvezni in označujeta trenutna položaja potiskalca in transportnega traka. Dovoljeni prehodi med posameznimi stanji so razvidni s slike 7.

Procedura prikazana na sliki 9 je naslednja: v koraku 2 (1) se prične potiskalec vrteti v urni smeri (2), kar pomakne transportni trak naprej (3). Ob pojavu napake (4) se mora potiskalec takoj ustaviti (5), s čimer se zmanjša možnost strojeloma. Posledično se ustavi tudi transportni trak (6). Napako odpravi operator s sprostitvijo potiskanja (7). Pri tem se potiskalec (8) in transportni trak pomakneta nazaj v izhodišče (9). Ko je napaka odpravljena, lahko operator ponovno požene avtomatsko izvajanje transporta materiala (10).



Slika 10: Pretvorba Petrijeve mreže v SFC

Za programiranje logičnih krmilnikov so uporabljeni jeziki v skladu z IEC 1131 standardom, predvsem kontakti načrt in sekvenčni funkcijski diagram (angl. Sequential Function Chart - SFC). Predstavljena metoda je kompatibilna z uporabo standardiziranih programskih jezikov. PN s slike 7 je mogoče enostavno pretvoriti v SFC [3], [9]. Prostori Petrijeve mreže v tem primeru kar odgovarjajo korakom SFC-ja, tranzicije Petrijeve mreže pa odgovarjajo tranzicijam SFC-ja. Primer pretvorbe Petrijeve mreže v sekvenčni funkcijski diagram

prikazuje slika 10. Koraki v SFC-ju so prikazani z okvirčki, izhodiščni korak pa je označen z dvojnim okvirčkom.

4. Sklep

V tem članku je predstavljen mehatronsko orientiran pristop k načrtovanju odprtega krmilja stroja, ki sinergijsko obravnava mehanske, elektronske in informacijske vidike proizvodnega sistema. Na začetku sta določena tehnologija stroja in princip delovanja. Nadalje je predstavljen primerni uporabniški vmesnik, ki omogoča učinkovito obvladovanje tehnologije stroja in uporabniku prijazno upravljanje stroja. Aparaturna oprema krmilja temelji na običajni PC tehnologiji, dopolnjeni z učinkovitim operacijskim sistemom za delo v realnem času in Ethernetni komunikacijski tehnologiji. Takšen koncept omogoča implementacijo različnih krmilnih algoritmov, uporabniškega vmesnika, in mrežnih opravil na enem samem procesorju. Vrednostno-orientirani algoritmi za pozicioniranje so implementirani na klasični način: interpretiranje gibanja, interpolacija in končno regulacija pozicioniranja. Za dogodkovno-orientirane algoritme je uveden nov matrični pristop. Modularni logični krmilnik omogoča ločeno načrtovanje in analizo logičnih krmilnih funkcij, njihovo modularno povezljivost in ponovno uporabnost v drugih sistemih. Tak pristop omogoča analizo vrednostno- in dogodkovno- orientiranih algoritmov še preden je ciljni sistem dejansko zgrajen. To je še posebej pomembno pri načrtovanju dragih in namenskih sistemov ter strojev, kjer podrobnih testiranj ni mogoče izvesti na ciljnem sistemu. Na področju običajne strojegradnje prinaša predlagani pristop koristi za načrtovalca stroja in za končnega uporabnika: omogoča skrajšanje razvojnega cikla ter integrirano načrtovanje tehnologije stroja in njegovega krmilja, ki pa je bolj zanesljivo, učinkovito in prijazno do uporabnika.

5. Literatura

- [1] Arutunian, E.B., D.R. Meldrun, N.A. Friedman, & S.E. Moody (1998). Flexible Software Architecture for User-Interface and Machine Control in Laboratory Automation. *BioTechniques* 25-4 (1998), pp. 698-705.
- [2] Bezi, I., & G. Tevesz, (1994) PC based robot controller for advanced control algorithms. *Proc. of the Joint Hungarian - British International Mechatronics Conference*, Budapest, Hungary, 1994.
- [3] Frey G., M. Minas, K.H. John, (2001). Integration von Petrinetzen in den Steuerungsentwurf nach IEC61131, *Proc. of the SPS/IPC/Drives*, Nürnberg, Germany, 2001, pp. 197-205
- [4] GM Powertrain Group, Manufacturing Engineering Controls Council (1996). Open, modular architecture controls at GM Powertrain. *Whitepaper available at http://www.arcweb.com/omac/Techdocs/Open_at_GMPTG.pdf*
- [5] Hace, A., K. Jezernik (2003). The Open CNC Controller for a Cutting Machine. *Proc. of IEEE ICIT 2003*, Maribor, Slovenia, pp. 1231-1236
- [6] Jensen, K. (1997). Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Volume 1. Basic Concepts. Monographs in Theoretical Computer Science, 2nd corrected printing. *Springer-Verlag*.
- [7] Loffler, M.S., V.K. Chitrakaran, & D.M. Dawson (2002). QMotor 3.0 and the QMotor Robotic Toolkit: A PC-Based Control Platform. *IEEE Control Systems Magazine* 22-3 (2002), pp.12-26.
- [8] Mireless J. Jr. & F.L. Lewis (2001). Intelligent Material Handling: Development and Implementation of a Matrix-Based Discrete-Event Controller. *IEEE Trans. on Industrial Electronics*. 48-6 (2001), pp. 1087-1097.

- [9] Park, E., D.M. Tilbury, & P.P. Khargonekar (1998). A Formal Implementation of Logic Controllers for Machining Systems using Petri Nets and Sequential Function Chart. *Proc. of the 1998 Japan-USA Symposium on Flexible Automation*, Otsu, Japan.
- [10] Peterson J.L. (1981). Petri Net Theory and the Modeling of Systems, *Prentice-Hall*, New Jersey.
- [11] Polič, A (2003). Matrični opis diskretnega dogodkovnega sistema. *Zbornik tretje konference AIG'03*, Maribor, Slovenija, pp. 179-184
- [12] Polič, A., A. Hacı, K. Jezernik (2003). Matrix based logic controller. *Proc. of IEEE ICIT 2003*, Maribor, Slovenia, pp. 370-374
- [13] Polič, A., A. Hacı, K. Jezernik (2003). System specification, performance analysis and control implementation using matrix-based approach. *Proc. of 12th International Workshop on Robotics in Alpe-Adria-Danube Region*, Cassino, Italy, 2003
- [14] QSSL, Kanata, Ontario K2M 1W8 Canada [On-line]. Available at: <http://www.qnx.com>
- [15] Sukhatme, G.S., J.E. Montgomery, & M.J. Matarić (1999). Design and Implementation of a Mechanically Heterogeneous Robot Group. *Proc. of the SPIE conference*, Boston, Massachusetts 1999.
- [16] Tacconi D.A. & F.L. Lewis (1997). A New Matrix Model for Discrete Event systems: Application to Simulation, *IEEE Trans. on Control Systems*, 17-Oct. (1997), pp. 62-71.
- [17] Zhou, M., F. DiCesare & A. A. Desroches (1992). A Hybrid Methodology for Synthesis of Petri Net Models for Manufacturing systems. *IEEE Trans. on Robotics and Automation*, 8-3 (1992), pp. 350-361.