Matrični opis diskretnega dogodkovnega sistema

Aleš Polič
Univerza v Mariboru
Fakulteta za Elektrotehniko, Računalništvo in Informatiko
Inštitut za robotiko
Smetanova ulica 17, SI-2000 Maribor
ales.polic@uni-mb.si

MATRIX BASED DISCRETE-EVENT SYSTEM DESCRIPTION APPROACH

Abstract: Functions of machine control can be described by their discrete states and the influence of the state of one function to the other functions is governed by discrete event supervision system such as PLC. This article presents a matrix based formal approach to description of such a control function, which is specified, analyzed and straightforward converted to standardized PLC code.

1 Uvod

Krmilja strojev sestavljajo krmilne funkcije, ki služijo spremljanju in upravljanju stroja s strani operaterja, povezavi stroja z okoljem, krmiljenju in regulaciji aktuatorjev na stroju in seveda izvajanju tehnoloških operacij in funkcij, ki jim je stroj namenjen. Vsaka izmed krmilnih funkcij lahko v določenem trenutku zavzame eno izmed svojih karakterističnih stanj. Karakteristični stanji posamezne funkcije sta v splošnem vsaj dve: opazovana krmnilna funkcija se v danem trenutku lahko izvaja ali pa ne. Prehajanje funkcije med stanji (tranzicija) je odvisno od trenutnega stanja opazovane funkcije in od stanja drugih funkcij v sistemu.

Klasično se k programiranju krmilnih funkcij pristopi na osnovi časovnega diagrama izvajanja tehnoloških funkcij stroja in na osnovi krmilne sheme elektroožičenja. 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, ki jo programer pripravi se v obliki programa naloži na PLK, nakar se prične s testiranjem osnovnih funkcij v ročnem režimu obratovanja, sledi testiranje avtomatskih funkcij in na koncu še optimizacija delovanja stroja.

2 Specifikacija krmilne logike

Čeprav so ravno funkcije za upravljanje, nadzor in varovanje stroja ter funkcije za samodejno zaustavljanje, diagnozo in odpravo napak tiste, ki ponavadi določajo kvaliteto krmilja, in zavzamejo večino vse krmilne logike (tudi do 90%) [1], ni najti nobenega pravega pristopa za specificiranje delovanja teh funkcij. Da je stvar vredna obravnave, je mogoče razvideti že iz standarda IEC 1131, ki govori o jezikih za programiranje PLK-jev. Med njimi je ponujen tudi t.i. sekvenčni diagram (SFC), ki omogoča razčlenitev programa PLK (krmilnih funkcij) v organizacijske enote v obliko medsebojno povezanih tranzicij in korakov. Tranzicija vsebuje prehodne pogoje, korak pa sestavljejo opravila, ki se izvedejo v okviru tega koraka. Takšen pristop že omogoča specifikacijo krmilne logike, vendar pa se pri večjih, kompleksnejših in visokoavtomatiziranih strojih za obvladovanje kompleksnosti krmilmnih funkcij pojavijo zahteve po sistematični specifikaciji in formalni analizi krmilij sistemov.

Sistematična specifikacija in formalna analiza krmilne logike zahteva uporabo matematično formalnih orodij. Krmilna logika strojev vsebuje značilnosti diskretnih dogodkovnih sistemov (DDS). Eno od pogosteje uporabljenih orodij za načrtovanje, simulacijo in analizo DDS so petrijeve mreže (PM). Čeprav se PM izkažejo za zelo uporabno orodje, pa se kljub vsemu pojavljata dve težavi: (i) modeliranje sistema z uporabo PM je prepuščeno izkušenosti in iznajdljivosti posameznega snovalca ter (ii) nepopolnost dinamičnega opisa takega modela [12].

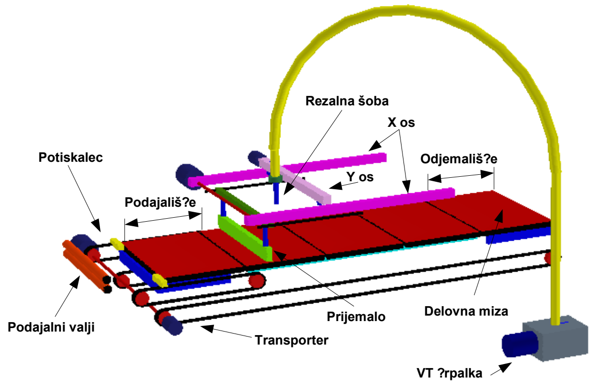
Ta članek predstavlja matrični način specifikacije DDS. Osnova matrične specifikacije je matrična predstavitev PM, ki poleg matričnega opisa strukture DDS omogoča tudi opis njegovih dinamičnih lastnosti. Takšen opis je dopolnjen z enačbo diskretnega dogodkovnega regulatorja, ki omogoča popoln dinamičen opis DDS.

Kot DDS je obravnavana krmilna funkcija menjave miz transportnega sistema stroja za razrez materialov z vodnim curkom. V prvem delu je predstavljen stroj za razrez materialov z vodnim curkom, sledi kratek uvod v teorijo petrijevih mrež (PM), v nadaljevanju pa je prikazan matrični pristop k modeliranju DDS in dinamični dogodkovni regulator. Za obravnavo zmogljivosti sistema so v matrično specifikacijo vključene časovne lastnostmi sistema. Dobljen model sistema je uporabljen za analizo časa cikla obravnavanega transportnega sistema.

3 Stroj

Stroj za razrez materialov z vodnim curkom je namenjen razrezu usnja, tekstila in sintetičnih materialov. Sestavljen je iz treh poglavitnih delov: transportnega sistema za podajanje materiala, dvoosne koordinatne mize in rezalnega sistem z visokotlačno črpalko in šobo (slika 1).

Transportni sistem podaja material v rezalno področje in ga drži v času rezanja. Sestavljen je iz petih delovnih miz, ki se pomikajo kot tekoči trak, naprej s pomočjo potiskalca, nazaj pa s pomočjo transporterja. Dviganje in spuščanje transportnih miz med transporterjem in potiskalcemje izvedeno s pomočjo dvižnih vilic, ki jih pomikajo pnevmatski cilindri. Držanje materiala v fazi rezanja je izvedeno s pomočjo prijemala. Material se dostavlja v dveh oblikah: plošče se polagajo neposredno na podajališče, bale pa se podajajo skozi podajalne valje.



Slika 1 : Stroj

4 Petrijeve mreže

PM so orodje za obvladovanje diskretnih sistemov. So bipartitni graf, ki ga opisuje množica. $\{\boldsymbol{P}, \boldsymbol{T}, \boldsymbol{I}, \boldsymbol{O}, \boldsymbol{m}\}$. Vozlišča grafa predstavljajo prostori $\boldsymbol{P}$, ki so praviloma prikazana s krogi, in tranzicije $\boldsymbol{T}$, ki jih prikazujejo pravokotniki. Vozlišča so medsebojno povezana z usmerjenimi povezavami, prikazanimi s puščicami. Vhodne povezave $\boldsymbol{I}$ vodijo iz prostorov k tranzicijam, izhodne povezave $\boldsymbol{O}$ pa vodijo od tranzicij v prostore. Stanje mreže, opišemo s porazdelitvijo žetonov po prostorih. Žetonov v posameznem prostoru prikazuje lokalno stanje tega prostora, število in razporeditev vseh žetonov v sistemu pa predstavlja stanje celotne mreže.

PM na sliki 2 prikazuje funkcijo menjave miz transportnega sistema. Prostori PM označeni s S predstavljajo opravila, ki se izvedejo znotraj te funkcij, prostori označeni z R pa predstavljajo resurse, ki so potrebni za izvajanje teh opravil. Žeton v prostoru S pomeni, da se opisano opravilo izvaja, žeton v prostoru R pa pomeni, da je opisan resurs razpoložljiv. Opravila in resursi vključeni v obravnavano funkcijo so razvidni iz tabel 1 in 2.

Dinamične lastnosti PM so prikazane s prehajanjem žetonov med prostori. Žeton se pomakne iz vhodnega prostora v izhodni prostor s proženjem tranzicije, ki povezuje oba prostora. Tranzicijo je mogoče sprožiti, če je omogočena. To pomeni, da vsak od njenih vhodnih prostorv vsebuje žeton. Pri proženje tranzicije se žetoni iz iz vhodnih prostorov pobrišejo, v izhodne prostore pa se pomaknejo novi žetoni.

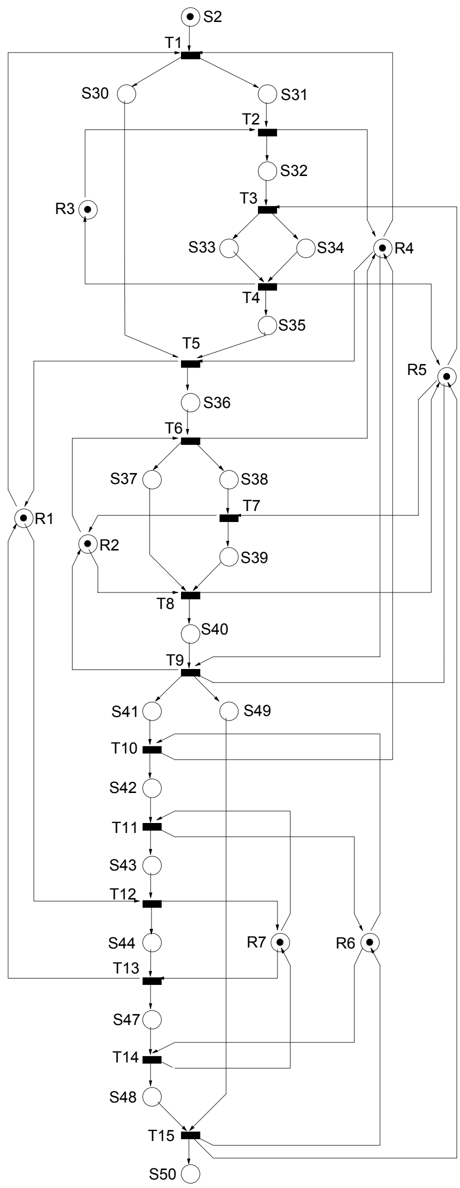
Opravilo	Opis
S30	Pomik x-osi v pozicijo prijemanja prijemala
S31	Pomik potiskalca v izhodišče
S32	Spuščanje sprednjih dvižnih vilic
S33	Vračanje delovne mize s transporterjem
S34	Dviganje sprednjih dvižnih vilic
S35	Brez operacije
S36	Potiskanje delovnih miz s potiskalcem
S37	Vračanje potiskalca v izhodišče
S38	Spuščanje zadnjih dvižnih vilic
S39	Pomik vrnjene delovne mize na zadnje dvižne vilice
S40	Dviganje zadnjih dvižnih vilic
S41	Potiskanje vrnjene delovne mize
S42	Sproščanje materiala, odklop prijemala od del. miz
S43	Prklop prijemala na x-os
S44	Vračanje prijemala v izhodišče
S47	Odklop prijemala od x-osi
S48	Priklop prijemala na del. mize, prijemanje materiala
S49	Vračanje transporterja v izhodišče

Tabela 1 : Opravila

Opravilo	Opis
S30	Pomik x-osi v pozicijo prijemanja prijemala
S31	Pomik potiskalca v izhodišče
S32	Spuščanje sprednjih dvižnih vilic
S33	Vračanje delovne mize s transporterjem
S34	Dviganje sprednjih dvižnih vilic
S35	Brez operacije
S36	Potiskanje delovnih miz s potiskalcem

Tabela 2 : Resursi

Graf PM je mogoče ekvivalentno opisati s pomočjo algebrajskih enačb [2]. V ta namen sta uvedeni matriki **S** in **F**, katerih stolpci ustrezajo prostorom PM, vrstice pa ustrezajo tranzicijam v petrijevi mreži. Matriko **F** imenujemo matrika predhodnih stanj. Člen **F**(*i,j*) ima vrednost 1, če je prostor v stolpcu *j* vhodni prostor tranzicije v stolpcu *i*, v nasprotnem primeru je vrednost člena **F**(*i,j*) enaka 0. Matriko **S**(*i,j*) imenujemo matrika naslednjih stanj. Člen **S**(*i,j*) ima vrednost 1, če je prostor v stolpcu *j* izhodni prostor tranzicije v stolpcu *i*, sicer je vrednost člena **S**(*i,j*) enaka 0. Če je PM čista [2] lahko matriki ***S*** in ***F*** združimo v incidenčno matriko **M**, s čemer je struktura sistema opisana z eno samo matriko (1)



Slika 2 : Petrijeva mreža

$$\mathbf{M} = \mathbf{F} - \mathbf{S} \tag{1}$$

Stanje sistema po *k* proženjih tranzicij lahko opišemo z vektorjem **m**(*k*). Komponente vektroja **m**(*k*) predstavljajo prostore v PM, vrednost posamezne komponente pa predstavlja število žetonov v pripadajočem prostoru. S pomočjo incidenčne matrike lahko zapišemo izraz za določanje naslednjega stanja sistema s pomočjo enačbe prehajanja stanj (2)

$$\mathbf{m}(k+1) = \mathbf{m}(k) + \mathbf{M}^T \mathbf{x}(k) \tag{2}$$

m(*k*) je pri tem vektor, ki opisuje trenutno stanje sistema, **m**(*k+1*) je vektor, ki opisuje novo stanje sistema, **M** je incidenčna matrika,

$\mathbf{x}(k)$ pa je vektor proženja tranzicij. Komponente vektorja $\mathbf{x}$ predstavljajo tranzicije v PM. Vrednost komponente vektorja proženja je 1, če naj se v danem trenutku tranzicija sproži, sicer je vrednost komponente enaka 0.

Učinkovito uporabo PM za potrebe načrtovanja krmilne logike sistemov omejujeta predvsem dve pomanjkljivosti: (i) modeliranje sistema s pomočjo grafa PM je precej zamudno in zahteva veliko izkušenj in (ii) vektor $\mathbf{x}$ v enačbi prehajanja stanj (2) ni enoumno določen.

Za reševanje problema modeliranja sistema je mogoče najti več prispevkov [3,4]. Najpogosteje so omenjeni pristopi »top-down«, »bottom-up« ali pa kombinacija obeh. Vendar pa vsi ti pristopi pa temeljijo na grafih PM, kar še vedno zahteva precej izkušenj za snovanje uporabnih modelov.

Za rešitev problema določanja vektorja $\mathbf{x}$ so uporabljeni pristopi kot je nadzorni krmilnik (angl supervisory control) [5, 6]. Slednji temelji na uvedbi dodatnih prostorov v mreži, pri čemer nadzorni krmilnik dodaja in odvzema žetone iz prostorov, s čemer preprečuje proženje tranzicij, ki bi povzročile neželjena stanja.

5 Matrični model DDS

Problem modeliranja DDS je poenostavljen z uporabo matričnega pristopa [5]. Problem modeliranja je iz načrtovanja grafa PM prenešen na sestavljanje incidenčne matrike sistema **M**, ki jo sestavljata matriki **F** in **S** (1). Stolpci matrik **F** in **S** ustrezajo prostorom PM, vrstice obeh matrik pa ustrezajo tranzicijam PM. Prostori PM ustrezajo stanjem DDS, tranzicije pa pomenijo pogoje za prehajanje med stanji DDS.

Stolpce matrik **F** in **S** lahko opišemo z množico prostorov PM $\boldsymbol{P} = \{\boldsymbol{PI}, \boldsymbol{v}, \boldsymbol{r}, \boldsymbol{PO}\}$. ***PI*** je množica vhodnih prostorov petrijeve mreže DDS (S2 na sliki 2), ***v*** je množica opravi, ki jih mora DDS izvesti (tabela 1), ***r*** je množica resursov, ki so potrebni za izvajanje opravil (tabela 2), ***PO*** pa je množica izhodnih prostorv PM DDS (S50 na sliki 2).

Vrstice matrik **F** in **S** predstavljajo pogoje za prehajanje DDS med stanji. Zapišemo jih lahko z množico tranzicij PM ***T***.

Potem, ko je pomen stolpcev in vrstic matrik **F** in **S** določen, je potrebno določiti strukturo PM oz. DDS. Matrika **F** je matrika predhodnih stanj sistema. Člen $\mathbf{F}(i,j)$ ima vrednost 1, če je prostor v stolpcu j vhodni prostor k tranziciji v prostoru i in torej predstavlja pogoj za proženje omenjene tranzicije. Matrika **S** je matrika naslednjih stanj sistema. Člen $\mathbf{S}(i,j)$ ima vrednost 1, če s proženjem tranzicije v stolpcu i sistem preide v prostor iz stolpca j. Če tranzicija in prostor nista povezana ima pripadajoči člen matrike **F** ali **S** vrednost 0. Matrika F ima za obravnavani sistem takšno obliko:

$$F = \left[\begin{array}{c:cccccccccccccccccc:ccccccc:c}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right] \quad (3)$$

Glede na pomen stolpcev lahko dobljeno matriko **F** razčlenimo kot podaja (4)

$$\mathbf{F} = \left[\mathbf{F}_u \;\vdots\; \mathbf{F}_v \;\vdots\; \mathbf{F}_r \;\vdots\; \mathbf{F}_y\right] \quad (4)$$

$\mathbf{F}_u$ je matrika vhodov in predstavlja začetne pogoje. Matrika sekvence opravil $\mathbf{F}_v$ določa sekvenco izvajanja posameznih opravil. Matrika resursov $\mathbf{F}_r$ določa resurse potrebne za izvajanje posameznih opravil. $\mathbf{F}_y$ je matrika sestavljena iz samih ničel in je potrebna za uskladitev dimenzij matrike **F** z dimenzijami matrike **M**. Na podoben način se določi matrika **S**, ki jo po analogiji lahko razčlenimo

$$\mathbf{S} = \left[\mathbf{S}_u \;\vdots\; \mathbf{S}_v \;\vdots\; \mathbf{S}_r \;\vdots\; \mathbf{S}_y\right] \quad (5)$$

$\mathbf{S}_v$ je matrika sprožanja novih opravil, $\mathbf{F}_r$ je matrika sproščanja resursov. $\mathbf{S}_y$ je matrika izhodov DDS in predstavlja rezultat izvajanja DDS npr. izdelek ali končanje neke funkcije. $\mathbf{S}_u$ je matrika sestavljena iz samih ničel in je potrebna za uskladitev dimenzij matrike **S** z dimenzijami incidenčne matrike **M**

6 Dinamični dogodkovni krmilnik DDS

Problem enoumnega določanja vektorja proženja tranzicij $\mathbf{x}$ in s tem predvidljivega obnašanja DDS je rešljiv z uvedbo diskretnega dogodkovnega regulatorja (DDR) [5,6]. Vektor proženja tranzicij $\mathbf{x}(k)$ se računa s pomočjo matrike predhodnih stanj sistema $\mathbf{F}$ in vektorja trenutnega stanja sistema $\mathbf{m}(k)$

$$x(k) = \overline{F} \oplus m(k) = \left[\overline{F_u \mid F_v \mid F_r \mid F_y}\right] \oplus \left[PI \mid v \mid r \mid PO\right]^T \quad (6)$$

Tako določen vektor $\mathbf{x}(k)$ dopolni enačbo prehajanja stanj PM (2) tako, da je omogočen enoumen zapis določanja naslednjega stanja sistema $\mathbf{m}(k+1)$ na podlagi poznavanja trenutnega stanja sistema $\mathbf{m}(k)$ in strukture sistema, ki jo opisuje incidenčna matrika $\mathbf{M}$.

Enačba DDR (6) se za razliko od enačbe prehajanja stanj PM (2) ne računa po pravilih standardne matrične algebre. Zgornja prečna črta pomeni logično negacijo, znak $\oplus$ pa pomeni logično operacijo, ki je podobna matričnemu množenju, vendar pa sta množenje in seštevanje komponent zamenjana z logičnima »in« in »ali«.

Za analizo zmogljivosti obravnavanega sistema je potrebno poznati trajanja posameznih opravil. Trajanje posameznemu opravilu je prirejeno tako, da se ob aktiviranju posameznega stanja žeton v pripadajočem prostoru zadrži za čas izvajanja tega opravila. Po preteku tega časa obravnavan prostor omogoči izhodne tranzicije. Opisani algoritem je izveden tako, da je posamezen prostor razdeljen v dva dela. Vektor stanja sistema $\mathbf{m}(k)$ je razdeljen v dva vektorja: $\mathbf{m}_i(k)$ predstavlja vsa pravila, ki se trenutno izvajajo in resurse, ki so trenutno zasedeni, $\mathbf{m}_f(k)$ pa predstavlja zaključena opravila in sproščene resurse [5].

$$\mathbf{m}(k) = \mathbf{m}_i(k) + \mathbf{m}_f(k) \quad (7)$$

Ob proženju tranzicije se žeton pomakne v prostor $\mathbf{m}_i(k)$, ki označuje izvajanje opravila, po preteku časa trajanja opravila pa se žeton pomakne v prostor $\mathbf{m}_f(k)$, ki označuje, da je opravilo končano in omogoča proženje izhodne tranzicije.

Zaradi razcepa vektorja $\mathbf{m}(k)$ na dva dela $\mathbf{m}_i(k)$ in $\mathbf{m}_f(k)$ je potrebno ustrezno preoblikovati tudi enačbo prehajanja stanj PM (2), ki razpade na dva dela

$$\mathbf{m}_i(k+1) = \mathbf{m}_i(k) + \mathbf{Sx}(k) \quad (8)$$

$$\mathbf{m}_f(k+1) = \mathbf{m}_f(k) - \mathbf{Fx}(k) \quad (9)$$

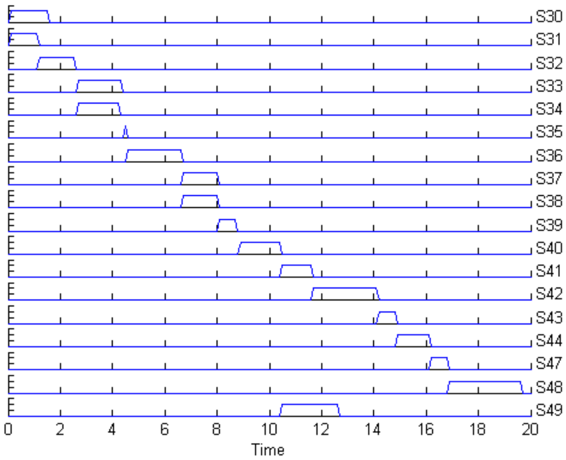
(8) opisuje vsa opravila ki se pričnejo po k-tem proženju tranzicij PM in pri tem aktivirane resurse skupaj s tistimi opravili, ki se že izvajajo oz resursi, ki so že aktivni. Pomik žetona iz prostora $\mathbf{m}_i(k)$ v $\mathbf{m}_f(k)$ se izvede po poteku časa izvajanja pripadajočega opravila. (9) predstavlja po k-tem proženju tranzicij končana opravila in sproščene resurse.

7 Simulacija transportnega sistema

Z uporabo matrik $\mathbf{F}$ in $\mathbf{S}$, enačbe prehajanja stanj PM (2) in enačbe DDR (6), lahko z uporabo programskega paketa Matlab izvedemo simulacijo delovanja in analizo zmogljivosti delovanja transportnega sistema stroja za vodni razrez.

Namen simulacije je določitev pričakovanega časa cikla izvajanja sekvence menjave miz. Čas cikla menjave miz je pomemben z vidika ocene ustreznosti konstrukcijske rešitve glede na kapaciteto stroja.

Čas cikla menjave miz z vgrajeno krmilno logiko po izvorni specifikaciji je znašal 55s. Z optimizacijo sekvence izvajanja opravil in povečanjem hitrosti aktuatorjev pa je bil čas cikla skrajšan na 19.7s. Časovni diagram izvajanja optimizirane sekvence menjave miz je prikazan na sliki 3. Optimizacija sekvence menjave miz je bila izvedena s paralelnim izvajanjem medsebojno neodvisnih opravil, ki sprikazana v tabeli 3. Opravila v levem stolpcu tabele 3 se izvajajo paralelno z opravili v desnem stolpcu tabele 3, kar je razvidno tudi s slike 2.



Slika 3

S30	S31, S32, S33, S34, S35
S33	S34
S37	S38, S39
S49	S41, S42, S43, S44, S47, S48

Tabela 3: Paralelna opravila

8 Zaključek

V tem članki je prikazan matrični pristop k specificiranju, modeliranju in dinamičnem opisu DDS. Matrični pristop temelji na matričnem zapisu PM in omogoča specificiranje in modeliranje sistema v matrični obliki brez zamudnega snovanja grafa PM. Enačba prehajanja stanj (2) takega modela je dopolnjena z enačbo DDR (6), ki v vsakem trenuku določi ustrezen vektor proženja tranzicij **x** glede na strukturo sistem in glede na trenutno stanje sistema. Razširitev matričnega zapisa z vključitvijo časa omogoča spremljanje zmogljivosti obravnavanega sistema.

Metoda je uporabljena na primeru snovanja krmilja transportnega sistema stroja vodni razrez materialov. Z dobljenim modelom so bile izdelane simulacije delovanja sistema. Glede na rezultate simulacij so bile v krmilno shemo vnešene spremembe s katerimi je bilo doseženo bistveno skrajšanje časa cikla menjave miz. Rezultat simulacije optimiranega krmilja je primerljiv z dejansko izmerjenim časom menjave miz na stroju, ki znaša 18s, kar utemljeje uporabnost predlagane metode.

Matrična predstavitev modela sistema je ekvivalentna grafu PM [2]. Zaradi te lastnosti je iz matrične predstavitve modela sistema dobljeno PM mogoče enostavno konvertirati v SFC programsko kodo, ki je eden od standardiziranih programskih jezikov industrijskih PLK-jev. Pristop se je izkazal za uporabnega na področju specifikacije in implementacije logičnih krmilij. Cilji nadaljnjih raziskav so približati pristop k snovanju krmilij hierarhičnih, distribuiranih in hibridnih sistemov.

9 Literatura

[1] E. Park, D.M.Tilbury, P.P. Khargonekar, *A Formal Implementation of Logic Controllers for Machining Systems using Petri Nets and Sequental Function Chart,* The 1998 Japan-USA Symposium on Flexible Automation, pp683-690, Otsu, Japan

[2] Peterson J. L., *Petri Net Theory and the Modeling of Systems*, New Jersey: Prentice-Hall, 1981

[3] K. Jensen, *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use.* Volume 1, Basic Concepts. Monographs in Theoretical Computer Science, Springer-Verlag, 2nd corrected printing, 1997

[4] M. Zhou, F. DiCesare and A. A. Desroches, *A Hybrid Methodology for Synthesis of Petri Net Models for Manufacturing systems*, IEEE Trans. on Robotics and Automation, vol. 8, no. 3, 1992 , pp. 350-361

[5] Diego A. Tacconi and Frank L. Lewis. *A New Matrix Model for Discrete Event systems: Application to Simulation*, IEEE Contr. Syst., vol. 17, Oct. 1997 , pp. 62-71

[6] Mireless J. Jr., Lewis L. Frank, *Intelligent Material Handling: Development and Implementation of a Matrix-Based Discrete-Event Controller*, IEEE Trans. on Industrial Electronics, vol. 48, no. 6, 2001, pp. 1087-1097