Dogodkovno voden pristop k snovanju logičnih krmilnih sistemov

Aleš Polič, Karel Jezernik
Univerza v Mariboru
Fakulteta za elektrotehniko, računalništvo in informatiko
Inštitut za robotiko
Smetanova ulica 17, SI-2000 Maribor
ales.polic@uni-mb.si, karel.jezernik@uni-mb.si

EVENT DRIVEN APPROACH FOR PRODUCTION SYSTEMS LOGIC CONTROL DESIGN

Abstract: This paper investigates the potentials of a new matrix based approach for description of event driven systems in the field of production systems control logic design. The relevance of the new academic developed approach for the implementation in the industrial logic control design problems is discussed and illustrated with an example.

1 Uvod

Razvoj novih tehnologij ter zahteve po vedno večji učinkovitosti, prilagodljivosti in zanesljivosti predvsem pa potrebe po večji avtonomnosti proizvodnih sistemov, strojev in naprav zahtevajo sistematičen pristop k obravnavi takih sistemov. Proizvodni sistemi so običajno inženirski sistemi, katerih namen je izvajanje predpisane proizvodne oz. tehnološke funkcionalnosti z zahtevano zmogljivostjo. Funkcionalnost in zmogljivost sta pogojeni s primerno konstrukcijsko zasnovo sistema ter ustrezno izbiro in namestitvijo aktuatorjev in senzorjev. Končne lastnosti sistema pa so določene šele s primerno zasnovo in izvedbo krmilja. Poleg same tehnološke funkcionalnosti mora krmilje proizvodnega sistema zagotavljati še nekatere pomožne funkcije npr. varovanje opreme in uporabnika, povezavo z okoljem, funkcije za daljinsko upravljanje ipd.

Eden od pogostejših problemov, ki se v praksi pojavi ob načrtovanju krmilij sistemov je način specifikacije zahtevane funkcionalnosti sistema. Običajno je podana tehnološka funkcionalnost v obliki Ganttovega diagrama, ki predpisuje sekvenco izvajanja tehnoloških operacij posameznih aktuatorjev ter pogoje za prehode med posameznimi koraki te sekvence. Izvedba pomožnih funkcij pa ponavadi ni podrobneje specificirana, zato je njena implementacija pogosto prepuščena programerjem. Bistvena težava omenjenega pristopa je, da podane specifikacije ni mogoče sistematično preverit, prav tako pa ni mogoče vnaprej preverjat zasnovanega krmilja. Morebitne pomanjkljivosti tako načrtovanega sistema so zato pogosto odkrite šele v poznih fazah projekta, kar pa je običajno povezano z visokimi stroški odpravljanja takšnih pomanjkljivosti.

Poleg vodenja zveznih veličin je običajno večji del krmilja proizvodnih sistemov namenjen zagotavljanju ustreznega odziva sistema na procesne dogodke in pogoje (npr. senzor aktiviran, nivo dosežen, gumb pritisnjen ipd.). Slednji so predstavljeni z logičnimi signali in povzročajo prehod sistema med diskretnimi stanji (gretje vklopljeno, vrtenje v desno, ipd.). Število diskretnih stanj sistema je omejeno, v normalnem delovanju pa so dovoljeni le prehodi med določenimi stanji. Prehajanje procesa med diskretnimi stanji ob pojavu procesnih dogodkov je pogosto označeno kot *dogodkovno vodenje sistema*, sistem ki vključuje dogodkovno vodenje pa je *diskretni dogodkovni sistem* (DDS). Krmilje, ki obvladuje dogodkovno dinamiko v sistemu je v praksi označeno kot *logično krmilje*.

V primerjavi z zveznimi sistemi je teorija, ki bi opisovala modeliranje, analizo in vodenje DDS bistveno manj razvita. Najpogosteje uporabljen pristop k vodenju DDS je predstavljen v [13], in predpostavlja da DDS generira dogodke spontano, sam od sebe, nadzorni krmilnik pa te dogodke le spremlja in jih omejuje glede na vnaprej predpisane zahteve. DDS je v tem primeru modeliran kot generator dogodkov in je predstavljen s pomočjo avtomata stanj, celotna teorija pa temelji na uporabi formalnih matematičnih jezikov. Bistvena pomanjkljivost te metode je neupoštevanje dejstva, da v praksi proces oz. sistem dogodkov načeloma ne generira sam od sebe, ampak se le odziva na dogodke, ki jih generirajo logični krmilni vhodi.

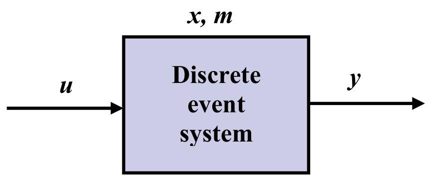
Poleg omenjenega pristopa je na področju vodenja DDS pogosto mogoče zaslediti uporabo Petrijevih mrež [6],[12]. Petrijeva mreža je v osnovi bipartitni graf, iz teorije grafov pa je poznano da je strukturo takega graf mogoče opisati s pomočjo matrik [6]. Matrični opis sistema je bistveno bližji klasični teoriji sistemov, kot pa formalni matematični jeziki in teorija grafov, pomanjkljivost matričnega opisa pa je predvsem nepopoln opis dinamike DDS.

V tem članku je predstavljena nova shema za modeliranje, simulacijo in analizo DDS. Strukturo DDS predstavljajo diskretna stanja in dogodki ter je zapisana v matrični obliki. Pojav dogodkov v sistemu je pogojen s trenutnim stanjem sistema ter trenutnimi vhodi v sistem in je zapisan s pomočjo logične enačbe. Dogodki sprožijo spremembo diskretnega stanja sistema, kar je zapisano s pomočjo rekurzivnega zakona, posameznemu diskretnemu stanju sistema pa je prirejena slika izhodnih signalov.

2 Diskretni dogodkovni sistemi

2.1 Karakteristične veličine

V tem članku obravnavani diskretni dogodkovni sistemi so procesi s končnim številom diskretnih stanj in dogodkovno proženim prehajanjem med njimi. Diskretna stanja procesa so določena s specifičnimi lastnostmi procesa npr. motor se vrti v levo, ventil je odprt, gretje je izklopljeno, avtomatski režim teče, ipd. V vsakem trenutku lahko proces zavzame le eno izmed diskretnih stanj, prehajanje med njimi pa prožijo dogodki.



Slika 1: Diskretni dogodkovni

Množico diskretnih stanj dogodkovnega sistema lahko predstavimo z logičnim vektorjem $\boldsymbol{m}$, katerega komponente predstavljajo posamezna diskretna stanja in imajo naslednji pomen:

$$\boldsymbol{m}(k)=\begin{cases}1, & \text{aktivno stanje}\\ 0, & \text{neaktivno stanje}\end{cases}. \tag{1}$$

Prehajanje procesa med diskretnimi stanji prožijo dogodki. Posamezen dogodek se zgodi, ko je izpolnjen pogoj, ki je za ta dogodek predpisan. Množico dogodkov dogodkovnega sistema lahko predstavimo z logičnim vektorjem $\boldsymbol{x}$, katerega komponente predstavljajo posamezne dogodke in imajo naslednji pomen:

$$\boldsymbol{x}(k)=\begin{cases}1, & \text{dogodek je}\\ 0, & \text{dogodka ni}\end{cases}. \tag{2}$$

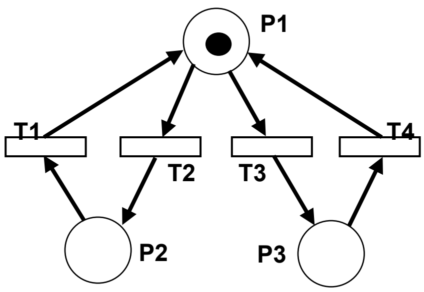
V praksi je posamezen dogodek ponavadi določen s kombinacijo več pogojev, prav tako pa je znotraj posameznega diskretnega stanja treba krmiliti več aktuatorjev. Ker gre v obeh primerih za logične signale sta uvedena še vhodni vektor $\boldsymbol{u}$ in izhodni vektor $\boldsymbol{y}$. Komponente vektorja $\boldsymbol{u}$ predstavljajo vhodne signale sistema, komponente vektorja $\boldsymbol{y}$ pa predstavljajo izhodne signale sistema.

$$\boldsymbol{u}(k)=\begin{cases}1, & \text{signal prisoten}\\ 0, & \text{signal ni prisoten}\end{cases} \tag{3}$$

$$y(k)=\begin{cases}1, & \text{izhod aktiven}\\ 0, & \text{izhod neaktiven}\end{cases} \tag{4}$$

2.2 Petrijeve mreže

Strukturo DDS predstavljajo povezave med diskretnimi stanji in dogodki. Pogosto je predstavljena s pomočjo grafa Petrijeve mreže [5],[9]. Petrijeva mreža je v osnovi bipartitni graf. Premore dve vrsti vozlišč: stanja DDS so prikazana s prostori (krogi), dogodki DDS pa so prikazani s tranzicijami (pravokotniki). Dovoljene prehode med stanji ponazarjajo puščice. Primer grafa Petrijeve mreže s tremi prostori in štirimi tranzicijami je prikazan na sliki 2.



Slika 2: Petrijeva mreža

2.3 *Matrična predstavitev strukture*

Strukturo grafa Petrijeve mreže lahko predstavimo s pomočjo matrik [6]. V ta namen sta uvedeni matriki $\boldsymbol{S}$ in $\boldsymbol{F}$. Matrika $\boldsymbol{F}$ predstavlja vse povezave iz prostorov k tranzicijam Petrijeve mreže. Stolpci matrike $\boldsymbol{F}$ predstavljajo prostore, vrstice pa predstavljajo tranzicije. Člen $\boldsymbol{F}(i,j)$ ima vrednost 1, če je prostor v stolpcu j vhodni prostor tranzicije v vrstici i, sicer je vrednost komponente 0. V dogodkovnem sistemu to pomeni, da stanje, ki ga predstavlja prostor v stolpcu j predstavlja pogoj za dogodek ki ga predstavlja tranzicija v vrstici i.

$$F(i,j) = \begin{cases} 1, & j\text{-state is a condition for } i\text{-event} \\ 0, & j\text{-state is not related to } i\text{-event} \end{cases} \tag{5}$$

Matrika $\boldsymbol{S}$ predstavlja vse povezave iz tranzicij v prostore Petrijeve mreže. Stolpci matrike $\boldsymbol{S}$ predstavljajo tranzicije, vrstice pa predstavljajo prostore grafa Petrijeve mreže. Člen $\boldsymbol{S}(i,j)$ ima vrednost 1, če je prostor v vrstici i izhodni prostor tranzicije v stolpcu j, sicer je vrednost komponente 0. V dogodkovnem sistemu to pomeni, da pojav dogodka, ki ga predstavlja tranzicija v stolpcu j postavi stanje ki ga predstavlja prostor v vrstici i.

$$S(i,j) = \begin{cases} 1, & i\text{-state is a consequence of j-event} \\ 0, & i\text{-state is not related to } j\text{-event} \end{cases} \tag{6}$$

$$\boldsymbol{F} = \begin{array}{c} \\ \mathrm{T1} \\ \mathrm{T2} \\ \mathrm{T3} \\ \mathrm{T4} \end{array} \begin{array}{c} \begin{array}{ccc} \mathrm{P1} & \mathrm{P2} & \mathrm{P3} \end{array} \\ \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array} \qquad \boldsymbol{S} = \begin{array}{c} \\ \mathrm{P1} \\ \mathrm{P2} \\ \mathrm{P3} \end{array} \begin{array}{c} \begin{array}{cccc} \mathrm{T1} & \mathrm{T2} & \mathrm{T3} & \mathrm{T4} \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{array} \tag{7}$$

Matriki $\boldsymbol{F}$ in $\boldsymbol{S}$ lahko združimo v enotno matriko $\boldsymbol{M}$, ki ji rečemo incidenčna matrika sistema (8).

$$\boldsymbol{M} = \boldsymbol{S}^T - \boldsymbol{F} \tag{8}$$

$$\boldsymbol{M}(i,j) = \begin{cases} +1, & j\text{-stanje je posledica } i\text{-dogodka} \\ 0, & j\text{-ni povezano z } i\text{-dogodkom} \\ -1, & j\text{-stanje je pogoj za } i\text{-dogodek} \end{cases} \tag{9}$$

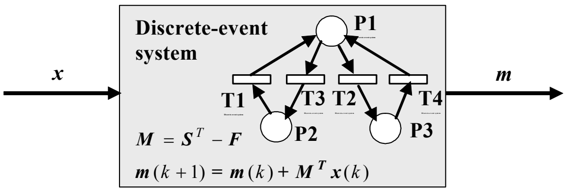
Slika 2 prikazuje graf Petrijeve mreže s tremi prostori in štirimi tranzicijami. Vhodni prostor tranzicije T1 je prostor P2, izhodni prostor tranzicije T1 pa je prostor P1. Podobno lahko določimo vhodne in izhodne prostore še za preostale tranzicije.

3 Dinamika dogodkovnega sistema

Dinamika dogodkovnega sistema opisuje vpliv pojavljanja posameznih dogodkov na prehajanjem dogodkovnega sistema med stanji. Ob poznani strukturi DDS, opisani z matriko $\boldsymbol{M}$, in poznanem trenutnem stanju DDS, opisanim z vektorjem $\boldsymbol{m}(k)$, je novo stanje po pojavu dogodka, opisanega z vektorjem $\boldsymbol{x}(k)$, določeno z diferenčno enačbo prehajanja stanje (10) [6].

$$\boldsymbol{m}(k+1) = \boldsymbol{m}(k) + \boldsymbol{M}^{\boldsymbol{T}}\boldsymbol{x}(k) \tag{10}$$

Indeks k v (10) označuje k-ti pojav dogodka v DDS. Produkt $\boldsymbol{M}^{\boldsymbol{T}}\boldsymbol{x}(k)$ določi spremembo stanj, ki zajema ukinitev preteklih aktivnih stanj in postavitev novih. V primerjavi s teorijo zveznih in časovno-diskretnih sistemov, opis sistema z enačbo (10) predstavlja neke vrste odprtozančno vodenje (slika 3). Dogodki za prehajanje med stanji niso natančno določeni.

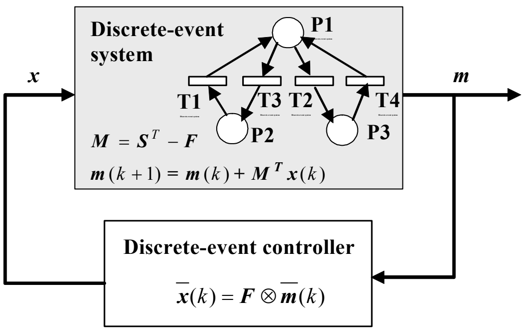


Slika 3: Odprtozančni model DDS

Prve poskuse zapisa dinamike DDS modeliranega v matrični obliki najdemo v [4],[10]. Dogodki, ki prožijo prehajanje DDS med diskretnimi stanji so določeni glede na trenutno stanje sistema ***m(k)*** z enačbo (11).

$$\overline{\boldsymbol{x}_e}(k) = \boldsymbol{F} \otimes \overline{\boldsymbol{m}}(k) \qquad (11)$$

Enačba (11) se računa po pravilih boolove algebre. Znak $\otimes$ predstavlja matrični boolov produkt, ki se računa podobno kot običajni matrični produkt, le da sta množenje oz seštevanje zamenjana z logično konjunkcijo (IN) oz. disjunkcijo (ALI) [10]. Vrstice matrike ***F*** predstavljajo diskretna stanja sistema, ki morajo biti aktivna za pojav dogodka s katerim je omenjena vrstica povezana. Boolov matrični produkt $\boldsymbol{F} \otimes \overline{\boldsymbol{m}}(k)$ preveri aktivnost potrebnih stanj DDS in vrne 1, v kolikor so potrebna stanja aktivna oz. 0 v kolikor niso. Formalizem je shematično prikazan na sliki 4.

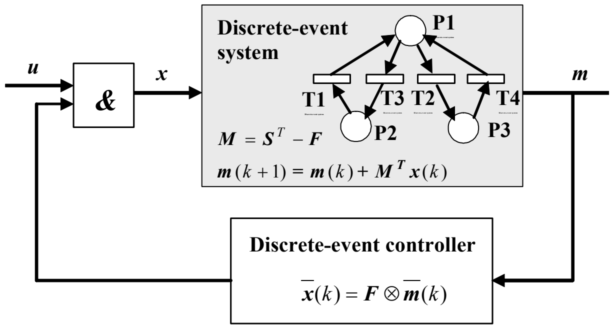


Slika 4: Določanje dogodkov na podlagi povratne informacije o stanju DDS

Opis, predstavljen na sliki 4, določa dogodke v sistemu na podlagi informacije o stanja sistema. Pomanjkljivost takšnega opisa pa je, da ne upošteva pogojev za dogodke, ki izvirajo iz vhodnih logičnih signalov. V ta namen je bila predlagana shema, kjer so dogodki določeni tako z upoštevanjem trenutnega stanja sistema ***m*** kot tudi krmilnih vhodov ***u*** (12).

$$\boldsymbol{x}(k) = \overline{\left(\boldsymbol{F} \otimes \overline{\boldsymbol{m}}(k)\right)} \& \boldsymbol{u} \qquad (12)$$

V (12) so dogodki določeni iz dveh delov: $\overline{\left(\boldsymbol{F} \otimes \overline{\boldsymbol{m}}(k)\right)}$ predstavlja pogoje ki izvirajo iz trenutnega stanja sistema, ***u*** pa predstavlja pogoje, ki izvirajo od zunaj. Oba dela sta združena z logično konjunkcijo, formalizem pa



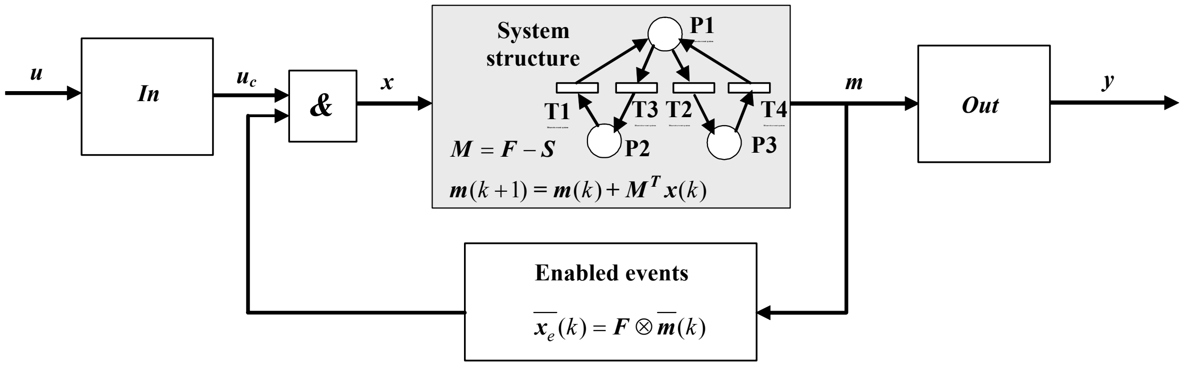
Slika 5: Matrični model DDS

je prikazan na sliki 5.

Zaradi uskladitve dimenzije vhodnega vektorja ***u*** z dimenzijo vektorja dogodkov ***x*** je uvedena vhodna transformacijska matrika ***In*** (13). Njeni stolpci predstavljajo vhodne signale ***u***, vrstice pa dogodke ***x***. Posamezna vrstica pomeni potrebno kombinacijo vhodnih signalov za izpolnitev pogoja za pripadajoč dogodek. Komponente matrike ***In***, ki predstavljajo vhode, kateri na dogodek v pripadajoči vrstice ne vplivajo, so označene z *x*. i-ta komponenta vektorja $\boldsymbol{u_c}$ ima vrednost 1, če se vhodni vektor ujema s pripadajočo vrstico, sicer je njena vrednost 0.

$$\boldsymbol{u}_c(k) = f(\boldsymbol{In}, \boldsymbol{u}(k)) \qquad (13)$$

Na podoben način je uvedena izhodna transformacijska matrika ***Out***, ki vektorju trenutnega stanja sistema ***m*** priredi izhodni vektor ***y***. Stolpci matrike D predstavljajo diskretna stanja ***m***, njene vrstice pa izhode ***y***. Posamezen stolpec predstavlja kombinacijo izhodnih signalov glede na pripadajoče diskretno stanje. Izhodni vektor ***y*** se računa iz trenutnega stanja sistema ***m*** in izhodne matrike



Slika 6: Matrični model DDS

Out s pomočjo matričnega boolovega produkta (15).

$$\boldsymbol{y}(k) = \boldsymbol{Out} \otimes \boldsymbol{m}(k) \tag{15}$$

Kot rezultat je predstavljena nova shema za opisovanje DDS, ki je predstavljena na sliki 6. Strukturo DDS predstavljajo diskretna stanja in dogodki, zapisana pa je v matrični obliki. Pojav dogodkov v sistemu je pogojen s trenutnim stanjem sistema ter trenutnimi vhodi v sistem in je zapisan s pomočjo logične enačbe. Dogodki sprožijo spremembo diskretnega stanja sistema, kar je zapisano s pomočjo rekurzivnega zakona. Posameznemu diskretnemu stanju sistema je prirejena slika izhodnih signalov.

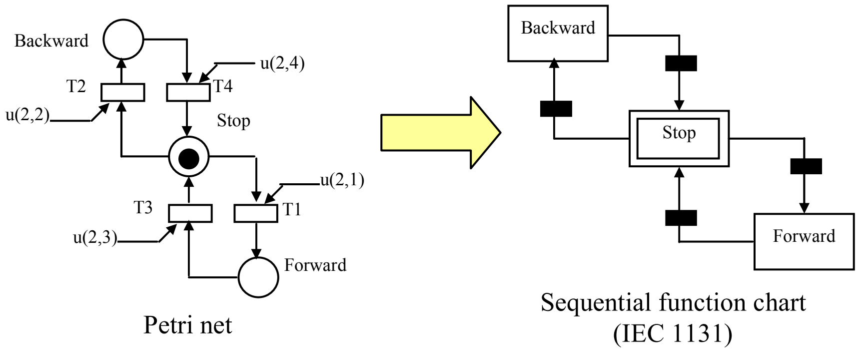
4 Zgled

Primer uporabe matričnega opisa DDS je prikazana na enostavnem zgledu modeliranja krmiljenja transportnega traka s konstantno hitrostjo. Transportni trak lahko miruje, se pomika naprej ali pa nazaj. Tej funkcionalnosti pripišemo tri diskretna stanja in jih zapišemo s pomočjo logičnega vektorja $\boldsymbol{m}$. Prehajanje med temi stanje se zgodi ob dogodkih $\boldsymbol{x}$, ki so v danem primeru naslednji: vklop naprej (T1), naprej izklop (T3), vklop nazaj (T2), nazaj izklop (T4). Seveda je pojav vsakega od teh dogodkov določen z izpolnitvijo predpisanih pogojev npr. vklop naprej se lahko zgodi ko transportni trak miruje in je pritisnjen gumb za vklop. Mirovanje transportnega traka je v tem primeru pogoj, ki izvira iz trenutnega diskretnega stanja transportnega traka (torej notranji pogoj), gumb za vklop pa je vhod (torej zunanji pogoj). Vhodi sistema so zapisani z vektorjem $\boldsymbol{u}$, izhodi $\boldsymbol{y}$ pa so v danem primeri kar logični krmilni signali za vklop pogonskega motorja. Vektorji in matrike modeliranega sistema so predstavljeni s (16), struktura sistema pa je grafično prikazana s Petrijevo mrežo na sliki 7.

$$\boldsymbol{M} = \begin{bmatrix} 1 & -1 & 0 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}, \boldsymbol{m} = \begin{bmatrix} Stop \\ Levo \\ Desno \end{bmatrix}, \boldsymbol{x} = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix}$$
$$\boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \boldsymbol{In} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{16}$$

5 Zaključek

Predstavljen matrični pristop k modeliranju DDS na področju snovanja logičnih krmilij. Z vidika raziskovalnega dela je predstavljen matematično formalen opis sistemov z inherentno dogodkovno dinamiko, ki je zelo podoben opisu zveznih oz. časovno diskretnih sistemov v prostoru stanj. Omogočeno je modeliranje sistemov, odprte pa so možnosti za njihovo analizo, simulacijo ter snovanje vodenja. Z vidika praktične uporabe predstavljen pristop omogoča neposredno implementacijo tako opisanega krmilja na programirljivih logičnih krmilnikih (PLK) [14], digitalnih signalnih procesorjih (DSP) in drugih programirljivih logičnih napravah (PLD, FPGA) [15]. Modeliran proces in krmilje je mogoče simulirati in preveriti njegovo delovanje vnaprej, dobljeno krmilje oz. algoritem pa v obliki matrik oz. grafični obliki enostavno prenesti na ciljni sistem. Graf Petrijeve mreže je



Slika 7: Pretvorba Petrijeve mreže v SFC

namreč podoben sekvenčnemu funkcijskemu diagramu (SFC), ki je eden izmed standardnih jezikov za programiranje PLK-jev. Pretvorba med Petrijevo mrežo in sekvenčnim funkcijskim diagramom je prikazana na sliki 7.

6 Literatura

[1] Bryan E. Graham, "Control Logic Requirements for Complex Manufacturing Systems," in *NSF Workshop on Logic Control for Manufacturing systems,* 2000

[2] 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

[3] 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*.

[4] 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.

[5] Park, E., D.M. Tilbury, & P.P. Khargonekar (1998). A Formal Implementation of Logic Controllers for Machining Systems using Petri Nets and Sequental Function Chart. *Proc. of the 1998 Japan-USA Symposium on Flexible Automation*, Otsu, Japan.

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

[7] Polič, A (2003). Matrični opis diskretnega dogodkovnega sistema. *Zbornik tretje konference AIG'03*, Maribor, Slovenija, pp. 179-184

[8] Polič, A., A. Hace, K. Jezernik (2003). Matrix based logic controller. *Proc. of IEEE ICIT 2003*, Maribor, Slovenia, pp. 370-374

[9] Polič, A., A. Hace, 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

[10] 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.

[11] 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.

[12] T. Murata, "Petri nets: Properties, analysis and applications," Proc. IEEE, vol. 77, Apr., 1989, pp. 541-580

[13] P.J. Ramadge, W. M. Wonham , "The control of discrete event systems," Proceedings of IEEE, 77(1) pp 81-98, 1989

[14] A. Polič, K. Jezernik, "A Matrix based approach to design of logic control functions for a production system," "2nd IEEE Conference on Industrial Informatics INDIN 2004, Berlin, Germany

[15] A. Polič, M. Čurkovič, K. Jezernik, "Matrix based design of current source inverter with FPGA implementation," Proc. of IEEE IECON'04, 2004, Busan, Korea.