ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1984

Seria: AUTOMATYKA z. 74 Nr kol.810

Zbigniew Banaszak

Instytut Cybernetyki Technicznej
Politechniki Wrocławskiej

ALGORYTMY SYNTEZY SIECIOWYCH MODELI PROCEDUR STEROWANIA DYSKRETNYMI PROCESAMI PRODUKCYJNYMI

Streszczenie. W pracy przedstawiono koncepcję automatycznej syntezy sekwencyjnych procedur sterowania przebiegami procesów technologicznych w dyskretnych systemach produkcyjnych. Główny nacisk położono na, będące podstawą koncepcji, algorytmy wyznaczania deterministycznych i bezblokadowych sieciowych modeli procedur sterowania procesami sekwencyjnymi, realizującymi w każdym cyklu jedno zadanie.

1. Wstęp

Rozwój automatyzacji dyskretnych systemów produkcyjnych jest ściśle uwarunkowany postępem w zakresie budowy i eksploatacji komputerowych systemów sterowania obiektami rozproszonymi. Sterowanie procesami technologicznymi w takich systemach wymaga odpowiednio przygotowanego oprogramowania, tzn. odpowiednich procedur monitorowania i diagnostyki pracy systemu, procedur koordynacji współpracy procesów przebiegających współbieżnie, procedur sterowania przepływem detali, narzędzi, itp.
Wraz ze wzrostem złożoności realizowanych procedur sterowania jak również koniecznością częstej ich wymiany i/lub modyfikacji, powodowanych zachodzącymi zmianami przebiegu procesów technologicznych, pojawia się konieczność korzystania z komputerowo wspomaganych systemów programowania. Fakt ten wskazuje na rosnącą potrzebę automatyzacji etapu przygotowania produkcji związanego z opracowaniem programów grupowego sterowania obrabiarek, centrów obróbkowych CNC, itp.

W niniejszej pracy przedstawiono pewną koncepcję automatycznej syntezy sekwencyjnych procedur sterowania przebiegami procesów technologicznych w dyskretnych systemach produkcyjnych. Zasadniczą cechą uzyskiwanych procedur jest możliwość określania cyklu produkcyjnego dla zadanego ciągu technologicznego, co między innymi pozwala na organizację procesów przepływowych z nawrotami.
Podstawą koncepcji są algorytmy pozwalające na uzyskiwanie deterministycznych i bezblokadowych modeli sieciowych sekwencyjnych procedur sterowania. Uzyskane rezultaty stanowią punkt wyjścia dla budowy algorytmów wyznaczających modele sieciowe procedur sterowania procesami sekwencyjnymi przebiegającymi współbieżnie, jak również stanowią podstawę dla budowy procedur symulacyjnego szeregowania zadań.

W pracy przedstawiono budowę i działanie algorytmów syntezy sieci Petriego modelujących procedury sterowania dla zadanej klasy procesów sekwencyjnych, omówiono koncepcję budowy systemów sterowania sekwencyjnymi, dyskretnymi procesami produkcyjnymi oraz scharakteryzowano zasadnicze kierunki przyszłych prac w tym zakresie.

2. Algorytmy syntezy żywych i bezkonfliktowych sieci Petriego

Język sieci Petriego jest jednym z powszechniej stosowanych obecnie języków modelowania procesów dyskretnych, w szczególności procesów przebiegających współbieżnie [5]. Sieciowa reprezentacja modelu procesu jest wygodnym zapisem umożliwiającym badanie takich jego własności jak: występowanie konfliktów, występowanie blokad, cykliczności procesu, itd. W praktyce, oznacza to przenoszenie się pewnych własności modelowanego procesu na określone własności strukturalne modelującej go sieci. Niektóre z tych własności są warunkami wystarczającymi istnienia określonych cech modelowanych procesów.
Z uwagi na ograniczoną objętość opracowania pominięto całkowicie teoretyczne podstawy sieci Petriego odsyłając czytelnika do obszernej literatury [7,8].

W pracy przedstawiono dwie różne procedury syntezy sieci Petriego $PN = (P,T,C,M_0)$ o zadanych własnościach funkcjonalnych. Rozważanymi dalej własnościami są: bezpieczność, bezkonfliktowość oraz żywotność sieci. Przedmiotem rozważanych procedur są sieci będące modelami deterministycznych procesów cyklicznych, w których nie występują blokady.

Rozważmy niżej sformułowany problem syntezy sieci Petriego o zadanych własnościach funkcjonalnych: Dla zadanej sekwencji σ przejść ze zbioru T określ sieć zgodną, żywą i bezkonfliktową $PN = (P,T',C,M_0)$, $T \subset T'$, taką że sekwencja σ', dla której zachodzi $\delta(M_0, \sigma') = M_0$, jest sekwencją modelującą sekwencję σ.
Oznacza to, że tak określona sieć winna cyklicznie odtwarzać zadaną sekwencję zdarzeń σ poprzez realizację modelującej sekwencji paleń σ'.
Synteza czystej i bezpiecznej sieci Petriego o wymienionych wyżej własnościach sprowadza się do wyznaczenia macierzy incydencji C oraz znakowania początkowego M_0. Określenie tych parametrów sieci równoważne jest wyznaczeniu macierzy procesu D [1], tzn. macierzy określonej przez następujące warunki:

i. $U_\sigma C = 0$, $U_\sigma C = \sum_{i=1}^{L(\sigma)} D[j_i]$

ii. $U_{j_i} C = D[j_i]$

gdzie: $L(\sigma)$ - oznacza liczbę elementów σ

$U_6 = U_{j_1} + U_{j_2} + \ldots + U_{j_k}$ – jest wektorem paleń sekwencji 6, gdzie j_i-ta współrzędna wektora U_6 oznacza liczbę pojawień się przejścia t_{j_i} w sekwencji 6,

$D[j_i]$ – jest j_i-tym wierszem macierzy D.

Konkurencyjne metodologie wyznaczania macierzy D prezentują dwa niżej przedstawione, oparte na różnych strukturach organizacyjnych, algorytmy syntezy sieci Petriego.

2.1 Algorytm LCFS – 1

Poniższe przedstawienie algorytmu zawiera jedynie werbalny opis jego zasadniczych kroków. Szczegółowy opis algorytmu przedstawiono w pracy [1].

Niech $6 = t_{j_1} t_{j_2} \ldots t_{j_n}$ będzie zadaną sekwencją przejść zbioru T. Poszukiwanymi parametrami są: macierz incydencji C oraz znakowanie początkowe M_0 sieci $PN = (P,T,C,M_0)$ modelującej zadaną sekwencję przejść 6.

krok 1. Wyznacz macierz $D = [d_{ij}]_{g \times a}$, której każdy wiersz zawiera po jednym elemencie równym odpowiednio 1 i -1. Wszystkie pozostałe elementy są równe 0. Niezerowe elementy macierzy D zorganizowane są w ten sposób, że jeżeli w danym wierszu macierzy znajduje się element równy 1 wówczas w odpowiadającej mu kolumnie wiersza bezpośrednio po nim następującego znajduje się element równy -1. W przypadku elementu równego 1 znajdującego się w ostatnim wierszu macierzy D, w odpowiadającej mu kolumnie wiersza pierwszego znajduje się element równy -1.

Ponadto, istnieje wzajemnie jednoznaczne odwzorowanie $\varphi : T \longrightarrow WM$, gdzie $WM \subset \{D[i] \mid i \in N\}$, takie że jeżeli $t_{j_i} = t_{j_k}$ oraz $j_i \neq j_k$, wówczas $\varphi(t_{j_i}) = D[a] = \varphi(t_{j_k}) = D[c]$, $a \neq c$.

Spełnione są również poniższe założenia:

i. Każde dwa wiersze $D[a]$, $D[c]$ macierzy D, takie że $D[a], D[c] \in \overline{WM}$ lub $D[a] \in WM$, $D[c] \in \overline{WM}$, gdzie $\overline{WM} = \{D[i] \mid i \in N\} \setminus WM$, traktowane są jako wiersze różne.

ii. Wiersze $D[a], D[c]$ są sobie równe, wtedy i tylko wtedy, gdy $D[a], D[c] \in WM$ i $\varphi^{-1}(D[a]) = \varphi^{-1}(D[c])$.

iii. Przyjmuje się, że $D[1] \in \overline{WM}$.

krok 2. Sprawdź czy każda kolumna macierzy D zawiera tylko jeden element równy -1. Jeżeli tak, wówczas przejdź do kroku 4, w przeciwnym wypadku rozszerz macierz D do macierzy D' według poniższego schematu powtarzanego dla kolumn macierzy, w których występują elementy równe -1 należące do co najmniej dwóch różnych wierszy macierzy D.

Dla danej kolumny, z wierszy macierzy, które w danej kolumnie zawierają element równy -1, wyznacz uporządkowaną rodzinę podzbiorów równych sobie wierszy. Porządek podzbiorów określony jest przez porządek występowania wierszy w macierzy D. Dla wszystkich elementów pierwszego podzbioru rodziny: rozszerz macierz D o liczbę kolumn równą pomniejszonej o jeden liczności elementów rodziny, wpisując w nowo utworzonych kolumnach wierszy, odpowiadających elementom pierwszego podzbioru, element równy 1, ponadto, w każdej z nowo utworzonych kolumn, w kolejności wyznaczonej uporządkowaniem zbiorów, wpisz element równy -1 dla każdego elementu należącego do kolejnego podzbioru rodziny.
Powyższe postępowanie powtórz dla rodziny podzbiorów pomniejszonej o element pierwszy, aż do momentu gdy liczność rodziny osiągnie wartość równą 1. Dla każdej z nowo wprowadzonych kolumn dokonaj sprawdzenia czy pierwszy i ostatni element (niezerowy element kolumny) są sobie równe.
W przypadku gdy elementy te równe są 1 , wówczas do odpowiedniej kolumny wiersza pierwszego macierzy wpisz element równy -1, w przeciwnym wypadku wpisz element równy 1. Przejdź do kroku 3.
krok 3. Sprawdź czy każda z nowo wprowadzonych w kroku 2 kolumn macierzy D' nie zawiera następujących po sobie (rozdzielonych jedynie elementami równymi 0) elementów równych 1 lub -1. Jeżeli tak, wówczas przejdź do kroku 4, w przeciwnym wypadku rozszerz macierz D' do macierzy D'' według poniższego schematu.
Dla każdej z nowo utworzonych kolumn macierzy D' wyznacz zbiór par wierszy, w których występują elementy równe -1 rozdzielone tylko elementami zerowymi. Jeżeli zbiór ten nie jest zbiorem pustym, wówczas rozszerz macierz D' wstawiając, dla aktualnie rozważanej kolumny, w wybranych wierszach należących do zbioru $\overline{WM}$, elementy równe 1 rozdzielające elementy par zbioru równe -1.
Następnie, dla aktualnie rozważanej kolumny wyznacz zbiór par wierszy, w których występują elementy równe 1 rozdzielone tylko elementami zerowymi. Jeżeli zbiór ten nie jest zbiorem pustym, wówczas dla każdego jego elementu powtórz poniższą procedurę.
Rozszerz macierz D' wstawiając, dla aktualnie rozważanej kolumny, w wybranych wierszach należących do zbioru $\overline{WM}$ elementy równe -1 rozdzielające elementy par zbioru równe 1. Przejdź do kroku 4.
krok 4. W oparciu o macierz D'' (lub D') wyznacz znakowanie początkowe - wektor o liczbie współrzędnych określonej liczbą kolumn macierzy. Elementami niezerowymi wektora (równymi 1) są współrzędne odpowiadające kolumnom, w których jako pierwszy (w kolejności od pierwszego wiersza macierzy) występuje element równy -1. Utwórz macierz C usuwając z macierzy D'' (lub D') wiersze nadmiarowe, równe uprzednio przeglądanym.

2.2 Algorytm LCFS - 2

Poniższy opis algorytmu jest werbalną eksplikacją jego podstawowych kroków. Budowę i działanie algorytmu przedstawiono w [2], natomiast jego formalne podstawy przedstawiono w [3].

Niech $\sigma = t_{j_1} t_{j_2} \ldots t_{j_n}$ będzie zadaną sekwencją przejść zbioru T. Poszukiwanymi parametrami są: macierz incydencji C oraz znakowanie początkowe M_0 sieci Petriego modelującej zadaną sekwencję przejść σ.

krok 1. Krok ten jest powtórzeniem pierwszego kroku algorytmu LCFS - 1. Założenia i.,iii. nie są warunkami koniecznymi kroku. Warunek ii. rozszerzony jest na elementy zbioru WM, tzn.

$(\forall D[b], D[c] \in WM)(D[b] = D[c] \longleftrightarrow (\forall j \in N_1^a)(D[b,j] = D[c,j]))$.

Przejdź do kroku 3.

krok 2. Sprawdź czy w każdej z nowo wprowadzonych kolumn macierzy D^i element równy -1 pojawia się tylko w powtarzających się, równych sobie, wierszach. Jeżeli tak, wówczas przejdź do kroku 4, w przeciwnym wypadku przejdź do kroku 3.

krok 3. Dla każdej z nowo wprowadzonych kolumn macierzy D^i utwórz sekwencję σ_i, której elementy odpowiadają wierszom zawierającym, w aktualnie rozważanej kolumnie, element równy -1. Dla każdego elementu utworzonego w ten sposób zbioru sekwencji powtórz krok 1.

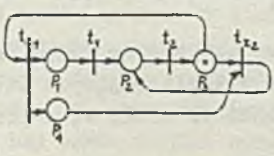
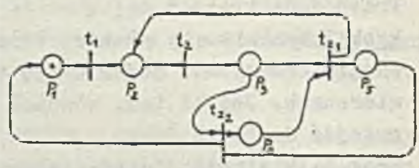
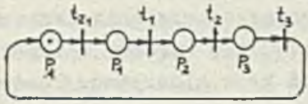
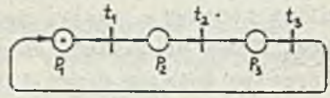
W rezultacie wykonania kroku 3, macierz D^i rozszerzana jest do macierzy D^{i+1}. Przejdź do kroku 2.

krok 4. Krok ten jest powtórzeniem czwartego kroku algorytmu LCFS - 1.

2.3 Charakterystyki algorytmów

Przedstawione algorytmy różnią się zarówno budową jak i założeniami warunkującymi ich realizację. Pierwszy z algorytmów pozwala uzyskiwać rezultat końcowy w wyniku jednokrotnego przejścia przez każdy z jego kroków. Drugi natomiast, osiąga rezultat końcowy w trybie iteracyjnym, powtarzając kolejno drugi i trzeci krok algorytmu. Spośród założeń warunkujących realizację algorytmów, warunki i.,ii.,iii. są warunkami podstawowymi dla algorytmu LCFS - 1. Warunkiem takim dla algorytmu LCFS - 2 jest odpowiednio zmodyfikowany warunek ii.

Jakkolwiek złożoność obliczeniowa obu algorytmów jest jednakowa $(O(n^2))$, tak uzyskiwane za ich pomocą wyniki, dla tych samych danych wejściowych, różnią się w sposób istotny. Dotyczy to zarówno różnic strukturalnych sieci, jak również generowanych przez nie sekwencji modelujących. Fakt ten został zilustrowany odpowiednimi przykładami na rys.1.

algorytm	LCFS - 1	LCFS - 2
sekwencja modelowana	$\sigma = t_1 t_2 t_2$	$\sigma = t_1 t_2 t_2$
macierz procesu	$\begin{array}{c\|cccc} & p_1 & p_2 & p_3 & p_4 \\ \hline t_{z_1} & 1 & & -1 & 1 \\ t_1 & -1 & 1 & & \\ t_2 & & -1 & 1 & \\ t_{z_2} & & 1 & -1 & -1 \\ t_2 & & -1 & 1 & \end{array}$	$\begin{array}{c\|ccccc} & p_1 & p_2 & p_3 & p_4 & p_5 \\ \hline t_1 & -1 & 1 & & & \\ t_2 & & -1 & 1 & & \\ t_{z_1} & & 1 & -1 & -1 & 1 \\ t_2 & & -1 & 1 & & \\ t_{z_2} & 1 & & -1 & 1 & -1 \end{array}$
sieć		
sekwencja modelująca	$\sigma' = t_{z_1} t_1 t_2 t_{z_2} t_2$	$\sigma' = t_1 t_2 t_{z_1} t_2 t_{z_2}$
sekwencja modelowana	$\sigma = t_1 t_2 t_3$	$\sigma = t_1 t_2 t_3$
macierz procesu	$\begin{array}{c\|cccc} & p_1 & p_2 & p_3 & p_4 \\ \hline t_{z_1} & 1 & & & -1 \\ t_1 & -1 & 1 & & \\ t_2 & & -1 & 1 & \\ t_3 & & & -1 & 1 \end{array}$	$\begin{array}{c\|ccc} & p_1 & p_2 & p_3 \\ \hline t_1 & -1 & 1 & \\ t_2 & & -1 & 1 \\ t_3 & 1 & & -1 \end{array}$
sieć		
sekwencja modelująca	$\sigma' = t_{z_1} t_1 t_2 t_3$	$\sigma' = t_1 t_2 t_3$

Rys.1 Porównanie algorytmów syntezy sieci Petriego.

W konkretnych implementacjach komputerowych, algorytmy te funkcjonują jako programy realizujące procedury sterowania w cyklicznych i sekwencyjnych procesach technologicznych.

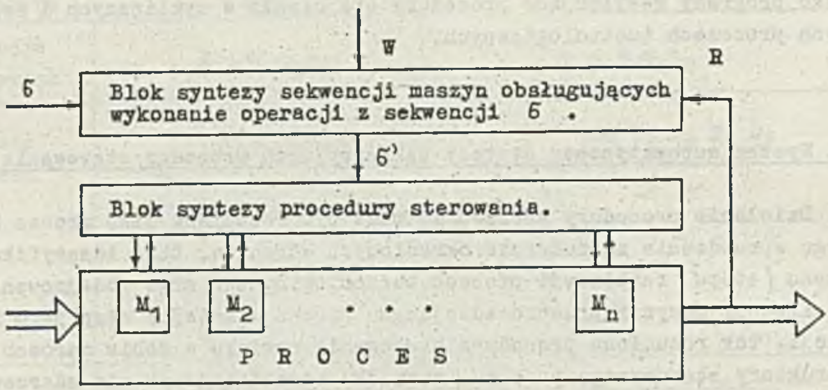
3. System automatycznej syntezy sekwencyjnych procedur sterowania

Działanie procedury sterowania może być rozumiane jako proces kolejnego sprawdzania zachodzenia określonych warunków, tzn. identyfikacji stanu (etapu) realizacji procesu technologicznego oraz podejmowania i realizacji decyzji przeprowadzających proces w kolejny etap jego realizacji. Tak rozumiana procedura sterowania zawiera w sobie zarówno opis struktury sterowanego procesu (porządku zachodzących w nim zdarzeń) jak również opis jego zachowania i własności, tzn. warunków koniecznych dla zajścia danego zdarzenia, warunków określonych w wyniku zajścia zdarzenia oraz charakterystyk określających, np. cykliczność procesu, jego sekwencyjność lub współbieżność, występowanie blokad, itp.

3.1 Koncepcja sterownika procesów sekwencyjnych

Dalsze rozważania ograniczone zostaną do przedstawienia koncepcji adaptacyjnych sterowników sekwencyjnych, umożliwiających realizację dowolnych struktur kombinacyjnego i sekwencyjnego sterowania logicznego. Koncepcję realizacji tego typu systemów, opartą na algorytmach syntezy modeli sieciowych procedur sterowania, przedstawia rys. 2.
Prezentowane podejście przyjęte zostało dla następująco sformułowanego zadania. Dana jest sekwencja σ zbioru operacji A będących składowymi realizowanego procesu P. Ponadto, zadany jest zbiór maszyn M, na których operacje ze zbioru A mogą być wykonywane (przyporządkowanie to określone jest przez relację $R \subset A \times M$) oraz zadane są charakterystyki W określające własności i ograniczenia procesu.
Zadaniem procedury sterowania jest załączanie i wyłączanie maszyn systemu produkcyjnego realizujących kolejne operacje procesu w sposób zapewniający jego zdeterminowany i bezblokadowy przebieg.

Proponowane podejście zakłada dwupoziomową strukturę systemu sterowania. Na wejście bloku poziomu pierwszego (najwyższego) podawana jest sekwencja operacji procesu, relacja R charakteryzująca stan systemu produkcyjnego oraz warunki określające charakterystyki realizowanego procesu. Na wyjście bloku podawana jest sekwencja maszyn σ realizujących operacje procesu P. Zadaniem tego bloku jest wyznaczanie, optymalnej w sensie przyjętego kryterium, sekwencji maszyn systemu oraz określenie typu programu



Rys.2 Schemat funkcjonalny samoprogramującego się sterownika.

wyznaczającego model sieciowy procedury sterowania.

Na wejście bloku poziomu drugiego podawana jest sekwencja maszyn б' realizujących operacje procesu. Na wyjściach bloku pojawiają się flagi, których numery przyporządkowane są maszynom systemu produkcyjnego, w kolejności określonej porządkiem operacji procesu. Pojawieniu się flagi towarzyszy zawieszenie wykonywania programu sterowania trwające do momentu pojawienia się sygnału zwrotnego z procesu - sygnału sygnalizującego zakończenie wykonywania operacji. Poziom ten jest reprezentowany przez, wyselekcjonowany na poziomie pierwszym, program będący dla zadanej sekwencji б' procedurą kierującą jej wykonaniem.
Proces realizuje obsługę zadań podawanych na jego wejście. Na wyjściu procesu, oprócz aktualnie obsłużonych zadań, pojawia się informacja o bieżącym stanie zasobów (maszyn) systemu produkcyjnego.

Możliwości budowy procedury sterowanie procesami przebiegającymi współbieżnie oraz strukturę adaptacyjnego systemu sterowania taką klasą procesów przedstawiono w [4].

Przedstawione wyżej podejście zakłada w ogólności możliwość budowy algorytmów wyznaczających modele sieciowe procedur sterowania dla określonych klas zachowania procesów dyskretnych. Założenie to zostało zilustrowane w kolejnym punkcie.

3.2 Sieciowe modele procedur sterowania

Algorytmy syntezy modeli sieciowych procedur sterowania mogą być sparametryzowane typem zachowania sterowanego procesu, np. wymaganiami cyklicznego lub przepływowego charakteru procesu, wymaganiami maksymali-

zacji wykorzystania czasu pracy maszyn w procesie przepływowym z nawrotami itp. Podejście takie wymaga określenia dla każdego typu zachowania procesu odpowiednich algorytmów syntezy modeli sieciowych.
Rozważmy przykład syntezy modelu sieciowego procedury sterowania procesem cyklicznym, sekwencyjnym z nawrotami, realizującym w każdym cyklu tylko jedno zadanie.

Przykład 1

Niech $\sigma = a_1a_2a_3a_4a_5a_6$ będzie daną sekwencją zbioru operacji A, procesu technologicznego realizowanego w systemie produkcyjnym składającym się ze zbioru maszyn M. Relacja R określająca zbiory maszyn obsługujących operacje ze zbioru A zadana jest w tablicy 1. Niech na zbiorze par relacji R określone będą funkcje czasu i kosztu wykonania operacji na obsługujących je maszynach. Ponadto, niech kryteriami oceny rozwiązań dopuszczalnych będą odpowiednio, minimalny czas cyklu oraz minimalny koszt realizacji zadania.

Tablica 1

operacje	maszyny
a_1	M_1, M_2
a_2	M_2, M_3
a_3	M_1, M_3
a_4	M_2, M_3
a_5	M_1, M_2, M_3
a_6	M_2

Należy wyznaczyć model sieciowy procedury sterowania procesem technologicznym należącym do klasy procesów sekwencyjnych, przepływowych z nawrotami, realizujących w każdym cyklu tylko jedno zadanie.
Niech $\sigma' = M_1M_2M_3M_2M_1M_2$ będzie wybraną sekwencją maszyn procesu minimalizującą czas cyklu oraz koszty realizacji zadania. Implementacja algorytmów LCFS-1, LCFS-2 dla sekwencji σ' pozwala wyznaczyć i porównać poszukiwane modele sieciowe - przedstawione odpowiednio na rys.3 i rys.4. Łatwo zauważyć, że dla zadanej sekwencji σ' algorytm LCFS-1 wyznacza sieć o mniejszej liczbie miejsc i przejść, co implikuje jej dalsze wykorzystanie jako modelu sieciowego procedury sterowania przebiegiem procesu technologicznego.

4. Uwagi końcowe

W niniejszej pracy przedstawiono niektóre zagadnienia budowy samoprogramujących się systemów sterowania dyskretnymi procesami produkcyjnymi. Zakres rozważań ograniczony został do przedstawienia koncepcji automatycznej budowy procedur sterowania procesami sekwencyjnymi, a w szczególa/

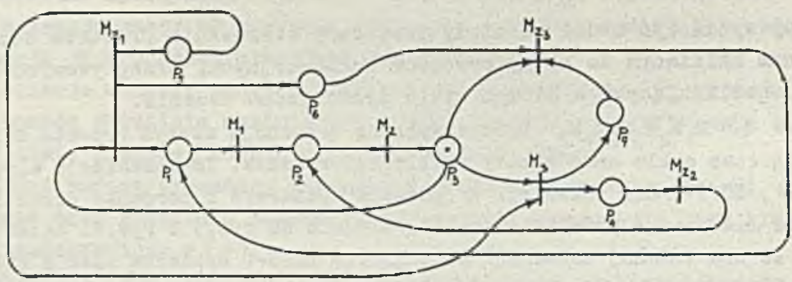
	P_1	P_2	P_3	P_4	P_5	P_6	P_7
M_{z_1}	1		-1		1	1	
M_1	-1	1					
M_2		-1	1				
M_3			-1	1	-1		1
M_{z_2}		1		-1			
M_2		-1	1				
M_{z_3}	1		-1			-1	-1
M_1	-1	1					
M_2		-1	1				

krok 1 krok 2

b/

$$\sigma' = M_{z_1} M_1 M_2 M_3 M_{z_2} M_2 M_{z_3} M_1 M_2$$

c/



Rys.3 Model sieciowy procedury sterowania wyznaczony według algorytmu LCFS - 1. a/ macierz procesu, na której wyróżniono podmacierze wyznaczane w kolejnych krokach algorytmu, b/ sekwencja modelująca sekwencję σ', c/ model sieciowy procedury sterowania.

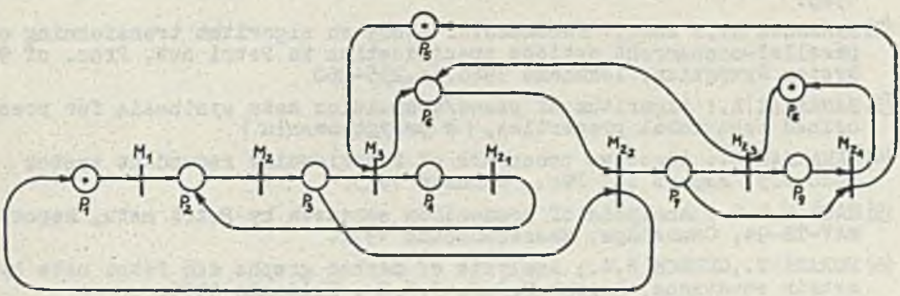
a/

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9
M_1	-1	1							
M_2		-1	1						
M_3			-1	1	-1	1			
M_{z_1}		1		-1					
M_2		-1	1						
M_{z_2}	1		-1			-1	1		
M_1	-1	1							
M_{z_3}						1	-1	-1	1
M_2		-1	1						
M_{z_2}	1		-1			-1	1		
M_{z_4}					1		-1	1	-1
	iteracja pierwsza				iteracja druga			iteracja trzecia	

b/

$$\sigma' = M_1 M_2 M_3 M_{z_1} M_2 M_{z_2} M_1 M_2 M_{z_3} M_{z_2} M_{z_4}$$

c/



Rys.4 Model sieciowy procedury sterowania wyznaczony według algorytmu LCFS - 2. a/ macierz procesu, na której wyróżniono podmacierze wyznaczane w kolejnych iteracjach pierwszego kroku algorytmu, b/ sekwencja modelująca sekwencję σ', c/ model sieciowy procedury sterowania.

ności procesami cyklicznymi realizującymi w każdym cyklu tylko jedno zadanie. Dla klasy takich systemów zaproponowane zostały dwa algorytmy syntezy modeli sieciowych procedur sterowania. Danymi wejściowymi algorytmów są porządki maszyn uczestniczących w realizacji procesu. Wynikami końcowymi są zupełne sieci Petriego, realizujące sekwencje przejść i modelujące sekwencje maszyn wykorzystywanych w procesie. Ponadto, przedstawiono koncepcję budowy samoprogramującego się sterownika opartą na wprowadzonych algorytmach. Koncepcja ta zakłada możliwość wyznaczania modelu sieciowego procedury sterowania procesem określonym przez ciąg technologiczny, zbiór maszyn systemu oraz założone wymagania użytkownika określające charakter procesu.
Przedstawione wyniki są punktem wyjścia dla budowy algorytmów wyznaczających procedury sterowania procesami przepływowymi z nawrotami, jak również procesami przebiegającymi równolegle. W szerszym sensie, stwarzają one również przesłanki dla budowy interakcyjnych jak i samoprogramujących się (adaptacyjnych) systemów sterowania dyskretnymi procesami produkcyjnymi. Systemy tego typu stwarzają szanse pełnej bądź częściowej automatyzacji etapu przygotowania oraz etapu programowania optymalnych procedur sterowania.

LITERATURA

[1] BANASZAK Z.: Algorithm of live and conflict-free Petri nets synthesis for prescribed system performance, Raport ICT PWr. nr 65/83, Wrocław 1983.

[2] BANASZAK Z., i inni,: Fundamental study on algorithm transforming of parallel-concurrent actions specification to Petri nets, Proc. of 9th System Symposium, Yokohama 1983. s.255-260

[3] BANASZAK Z.: Algorithm of place/transition nets synthesis for prescribed behavioral properties, (w przygotowaniu)

[4] BANASZAK Z.: Adaptive procedure of functionally redundant system recovery, Raport ICT PWr., Wrocław 1983.

[5] HACK M.H.T.: Analysis of production schemata by Petri nets, Report MAV-TR-94, Cambridge, Massachusetts 1972.

[6] MURATA T., CHURCH R.W.; Analysis of marked graphs and Petri nets by matrix equations, Report M.D.C. 1.1.8 , Illinois 1975.

[7] PETERSON J.L.: Petri net theory and the modelling of systems, Prentice-Hall, New York 1981.

[8] STARKE P.H.: Petri-netze , VEB Deutscher Verlag der Wissenschaften, Berlin 1980.

Recenzent: Prof. dr hab. inż. Antoni Niederliński

Wpłynęło do Redakcji do 30.03.1984r.

АЛГОРИТМЫ СИНТЕЗА СЕТЕВЫХ МОДЕЛЕЙ ПРОЦЕДУР УПРАВЛЕНИЯ ДИСКРЕТНЫМИ ПРОИЗВОДСТВЕННЫМИ ПРОЦЕССАМИ

Р е з ю м е

В статье представлена концепция синтеза последовательных процедур управления технологическими процессами в дискретных системах. Рассматриваются , являющиеся основой концепции , алгоритмы безблокадных и безконфликтных сетевых моделей процедур управления последовательными и циклическими процессами , в котором обслуживается в каждом цикле только одна задача.

DESIGN ALGORITHMS OF NET CONTROL PROCEDURES MODELS FOR SEQUENTIAL PRODUCTION PROCESSES

S u m m a r y

The paper presents a concept of automatic design of control procedure for technological process performance. The considerations are related to algorithms of design of conflict-free and live nets which are models of control procedures for cyclic and sequential processes, where only one job is performed in each cycle.