

Jerzy Cyklis

Politechnika Krakowska

## ALGORYTM SYMULACJI ESP

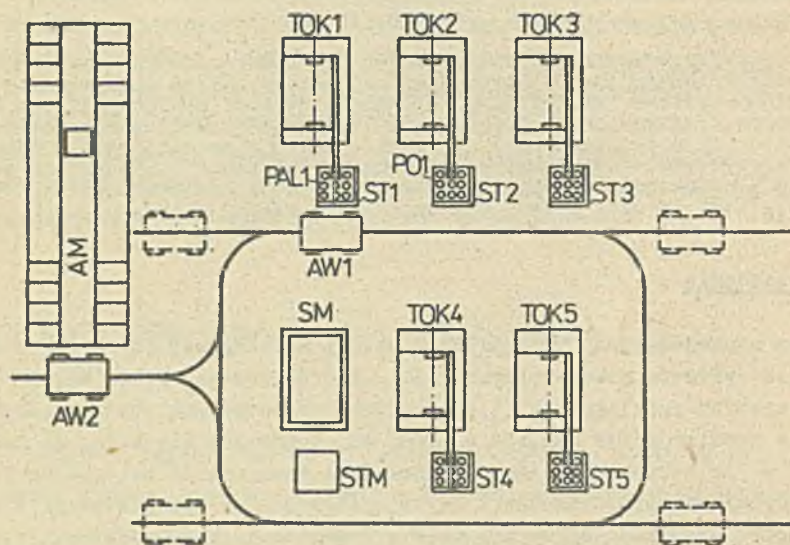
**Streszczenie.** Algorytm symulacji ESP opiera się na obliczaniu macierzy stanu  $[S_{jk}(i)]$  oraz czasu bieżącego  $T(i)$  na każdym etapie i zmiany systemu. Macierz  $[S_{jk}(i)]$  określa różnicę pomiędzy liczbą elementów  $k$ , gotowych do rozpoczęcia czynności  $j$ , a liczbą elementów  $k$  używanych w tej czynności. Macierz  $[S_{jk}(i)]$  i czas bieżący  $T(i)$  są wykorzystywane do wyznaczenia początku lub końca czynności na etapie  $i$ . Stosownie do ostatniego zdarzenia oblicza się macierz stanu  $[S_{jk}(i+1)]$  i czas  $T(i+1)$  dla następnego etapu  $(i+1)$ .

1. Wprowadzenie

Praca niniejsza jest rozwinięciem metody macierzowej modelowania elastycznych systemów produkcyjnych [ESP] zaproponowanej przez autora we wcześniejszych publikacjach [1,2]. Informacje dotyczące stanu wiedzy w zakresie symulacji ESP zawarto w pracy [2], opierając się m.in. na pozycjach [5 ÷ 7]. Stwierdzono, że zaproponowana nowa metoda pozwala na lepsze porozumienie projektantów i użytkowników ESP ze specjalistami z zakresu metod symulacji cyfrowej. Zaletą jej jest łatwość modelowania złożonych układów i możliwość dostosowania do różnych strategii sterowania ESP. Dalsze prace nad metodą prowadzone w Centralnym Problemie Badań Podstawowych 02.04 potwierdziły przydatność metody w zakresie macierzowego zapisu struktury ESP, wskazały także na celowość jej modyfikacji. Celowe okazało się wprowadzenie możliwości występowania w jednej czynności więcej elementów tego samego rodzaju. Zauważono także przydatność modelu do sterowania systemu rzeczywistego. Wykorzystano ogólne wskazania budowy takich systemów zawarte np. w [3,4] z zastosowaniem sieci komputerów pracujących w czasie rzeczywistym. W modelu opisanym w poprzednich pracach symulacja odbywała się wg algorytmu sekwencyjnego, w którym nie występowało pojęcie czasu bieżącego. W opisanym tu modelu wprowadzono czas bieżący, który w systemie rzeczywistym może być zastąpiony czasem rzeczywistym.

## 2. Macierze charakterystyczne

Dla ilustracji metody przyjęto ESP wg schematu na rys. 1. Dwa typy przedmiotów obrabianych /PO1, PO2/ są transportowane w paletach /PAL1,



rys.1. Schemat ESP do obróbki przedmiotów obrotowych /FMS for turning/

PAL2/ po 9 sztuk w każdej. Automatyczne wózki /AW1, AW2/ obsługują cały transport pomiędzy automatycznym magazynem /AM/, stołami obrabiarek /ST1, ST2, ST3, ST4, ST5/ oraz stołem stacji mycia /STM/. Oba przedmioty /PO1, PO2/ mogą być obrobione na dowolnej stacji tokarskiej /TOK1, TOK2, TOK3, TOK4, TOK5/, a następnie myte na stacji mycia STM. Czynności wykonywane przez system są zdyskretyzowane, tworząc zbiór tzw. czynności elementarnych  $\{j = 1 \div J\}$ . Każdej czynności są przyporządkowane dwa elementarne zdarzenia - jej początek i koniec. Cały system jest również zdyskretyzowany ze względu na występujące w nim części składowe. Części systemu, które działają zawsze jako jedna całość tworzą zbiór elementów  $\{k = 1 \div K\}$ . W każdej czynności  $j$  bierze udział zbiór elementów  $\{k(j)\} \subset \{k = 1 \div K\}$ , który musi być przygotowany do wykonania tej czynności przed jej rozpoczęciem. Natomiast po zakończeniu czynności  $j$  niektóre z elementów ze zbioru  $\{k(j)\}$  mogą być przygotowane do rozpoczęcia następnej czynności ze zbioru  $\{j = 1 \div J\}$ . Zbiór czynności, do których przygotowany jest element  $k$  po zakończeniu czynności  $j$ , jest oznaczony  $\text{Out}(j, k)$  i nazwany zbiorem wyjść. Tablica 1 pokazuje zbiory wyjść  $\text{Out}(j, k)$  dla wybranych czynności



Tablica 1		Zbiory wyjść Out (j,k)							
Element		AM	AW1	TOK1	SM	ST1	STM	PAL1	PO1
Czynność	j \ k	1	2	4	9	10	15	16	18
PAL1, AM → AW1	1	1,2,45-48,91,92	3-8,45					3-7	3-7
PAL1, AW1 → ST1	3		93			33,39		15,33,39	15
PAL1, AW1 → STM	8		93				20	20	
PO1, PAL1 → TOK1	15			21				27	21
PAL1, STM → SM	20				26		32	26	
PO 1, TOK1 "	21			27					27
PAL1, SM "	26				32			32	
PO1, TOK1 → PAL1	27			15				15,33,39	33
PAL1, SM → STM	32				20		38,44	38,44	
PAL1, ST1 → AW1	33		3-8,45			3,9		8	End
PAL1, STM → AW1	38		3-8,45				8,14 54,60	45	
PAL1, AW1 → AM	45	1,2,45-48,91,92	93					End	
AW1 "	93		1,33-38,47,79-84						

$j \in \{j = 1 \div 94\}$ ,  $k \in \{k = 1 \div 19\}$ . Wprowadzono tu uproszczone nazwy czynności /wygodne przy wprowadzaniu danych do komputera/. Pierwsza nazwa, przed przecinkiem odnosi się do elementu przekazywanego lub podlegającego przemianom. Następne nazwy wskazują skąd i dokąd element jest przekazywany /przypadek strzałki/ lub gdzie podlega przemianom /gwiazdka/. Np. /PAL1, AM — AW1/ oznacza, że PAL1 jest pobierana z AM przez AW1. PO1, TOK1\* oznacza, że PO1 jest obrabiany na TOK1. AW1\* oznacza, że AW1 przechodzi do pozycji wyczekiwania. Można podać proste przykłady interpretacji zbiorów wyjść;  $\text{Out}(15, 18) = \{21\}$  oznacza, że po wykonaniu czynności  $j = 15(\text{PO1, PAL1} \rightarrow \text{TOK1})$ , element  $k = 18(\text{PO1})$  jest przygotowany do rozpoczęcia czynności  $j = 21(\text{PO1, TOK1}^*)$ .  $\text{Out}(1, 6) = \{3, 4, 5, 6, 7\}$  oznacza, że po zakończeniu czynności  $j = 1(\text{PAL1, AM} \rightarrow \text{AW1})$  element  $k = 16(\text{PAL1})$  jest przygotowany do rozpoczęcia czynności  $j = 3(\text{PAL1, AW1} \rightarrow \text{ST1})$ ,  $j = 4(\text{PAL1, AW1} \rightarrow \text{ST2})$ ,  $j = 5(\text{PAL1, AW1} \rightarrow \text{ST3})$ ,  $j = 6(\text{PAL1, AW1} \rightarrow \text{ST4})$ ,  $j = 7(\text{PAL1, AW1} \rightarrow \text{ST5})$ . Słowo End oznacza, że dany element bierze udział w czynności, ale kończy swoje samodzielne istnienie w systemie. Np. po czynności  $j = 45(\text{PAL1, AW1} \rightarrow \text{AM})$ , element  $k = 16(\text{PAL1})$  pozostaje już na stałe w magazynie (AM). Zbiór wyjść dla tych przypadków jest zbiorem pustym  $\emptyset$ , np.  $\text{Out}(45, 16) = \emptyset$ . Również tam, gdzie nie zadeklarowano wyjść, zbiór wyjściowy jest zbiorem pustym  $\emptyset$ .

Informacja - podana w tablicy 1 - może być również zawarta w tzw. macierzach wyjść  $[\text{Out}_{jk}(1)]$ , dla każdej czynności  $1 \in \{j = 1 \div J\}$ . Elementy tej macierzy mogą mieć wartość 1 lub 0.  $\text{Out}_{jk}(1) = 1$  oznacza, że po zakończeniu czynności 1 element  $k$  jest przygotowany do rozpoczęcia czynności  $j$ . Np.  $\text{Out}_{3, 16}(1) = 1$  oznacza, że po zakończeniu czynności  $1 = 1(\text{PAL1, AM} \rightarrow \text{AW1})$  element  $k = 16(\text{PAL1})$  jest gotowy do rozpoczęcia czynności  $j = 3(\text{PAL1, AW1} \rightarrow \text{ST1})$ .  $\text{Out}_{jk}(1) = 0$  oznacza, że po zakończeniu czynności 1, element  $k$  nie jest przygotowany do rozpoczęcia czynności  $j$ . Definiuje się też macierze wejść  $[\text{Inp}_{jk}(1)]$ , dla każdej czynności  $1 \in \{j = 1 \div J\}$ . Dla  $\text{Inp}_{jk}(1) = 1$  po zakończeniu czynności  $j$ , element  $k$  jest przygotowany do rozpoczęcia czynności 1. Dla  $\text{Inp}_{jk}(1) = 0$ , po zakończeniu czynności  $j$ , element  $k$  nie jest przygotowany do rozpoczęcia czynności 1. Elementy macierzy wyjść i wejść są związane w następujący sposób:

$$\text{dla } \text{Out}_{jk}(1) = 1, \quad \text{Inp}_{1k}(j) = 1$$

/1/

$$\text{dla } \text{Out}_{jk}(1) = 0, \quad \text{Inp}_{1k}(j) = 0$$

Dla celów dalszej analizy wprowadza się także tzw. macierze eliminacji  $[\text{El}_{jk}(1)]$  dla każdej czynności  $1 \in \{j = 1 \div J\}$ . Macierz ta opisuje zablokowanie niektórych elementów  $k$  wymaganych do wykonania czynności  $j$  na skutek rozpoczęcia czynności 1. Element tej macierzy  $\text{El}_{jk}(1)$  może po-



siadać wartość 1 lub 0.  $El_{jk}(1) = 1$  oznacza, że przy rozpoczęciu czynności 1 użyto elementu k, który uprzednio mógł być użyty do rozpoczęcia czynności j.  $El_{jk}(1) = 0$  oznacza, że przy rozpoczęciu czynności 1 nie użyto elementu k, który bierze udział w czynności j. Dla wyznaczenia elementów  $El_{jk}(1)$  wykorzystuje się macierz wejść  $[Inp_{jk}(1)]$ . Dla czynności j, dla których  $Inp_{j=n} k(1) = 1$ , przeszukuje się macierz  $[Out_{jk}(1=n)]$ .

Dla  $Out_{j=n} k(1) = 1$  przyjmuje się  $El_{j=n} k(1) = 1$ ; dla pozostałych czynności j  $El_{jk}(1) = 0$ . Np.  $Inp_{1\ 16}(3) = 1$ , stąd przeszukuje się macierz  $Out_{jk}(1) = Out_{3\ 16}(1) = Out_{4\ 16}(1) = Out_{5\ 16}(1) = Out_{6\ 16}(1) = Out_{7\ 16}(1)$ . Stąd, stosownie do podanej wyżej reguły  $El_{3\ 16}(3) = El_{4\ 16}(3) = El_{5\ 16}(3) = El_{6\ 16}(3) = El_{7\ 16}(3) = 1$ . Wynik ten jest interpretowany w ten sposób, że po rozpoczęciu czynności 1 = 3 (PAL1, AW1 → ST1) element k = 16 jest zablokowany dla rozpoczęcia czynności j = 3 /powtórne rozpoczęcie czynności 1 = 3/ (PAL1, AW1 → ST1), j = 4 (PAL1, AW1 → ST2), j = 5 (PAL1, AW1 → ST3), j = 6 (PAL1, AW1 → ST4), j = 7 (PAL1, AW1 → ST5).

W wielu praktycznych przypadkach celowe jest wprowadzenie uogólnienia polegającego na tym, że w danej czynności może brać udział dowolna liczba całkowita elementów oznaczonych numerem k. Tablica 2 podaje fragment ma-

Tablica 2. Macierz liczebności elementów biorących udział w czynnościach  $[U_{jk}]$

Element		AM	AW1	TOK1	SM	ST1	STM	PAL1	PO1
Czynność	$j \backslash k$	1	2	4	9	10	15	16	18
PAL1, AM → AW1	1	1	1					1	9
PAL1, AW1 → ST1	3		1			1		1	9
PAL1, AW1 → STM	8		1				1	1	
PO1, PAL1 → TOK1	15			1				1	1
PAL1, STM → SM	20				1		1	1	
PO1, TOK1 *	21			1					1
PAL1, STM *	26				1			1	
PO1, TOK1 → PAL1	27			1				1	1
PAL1, SM → STM	32				1		1	1	
PAL1, ST1 → AW1	33		1			1		1	9
PAL1, STM → AW1	38		1				1	1	
PAL1, AW1 → AM	45	1	1					1	
AW1 *	93		1						

cierzy liczebności elementów  $[U_{jk}]$  dla czynności i elementów wg tab.1. Np.  $U_{3\ 18} = 9$  oznacza, że w czynności j = 3 (PAL1, AW1 → ST1) bierze udział 9 elementów k = 18 (PO1). Wynika to z przyjęcia 9 sztuk PO1 znajdujących się w PAL1.

Macierze  $[Out_{jk}(1)]$ ,  $[Inp_{jk}(1)]$ ,  $[El_{jk}(1)]$  są wyznaczane algorytmicznie przez program obliczeniowy na podstawie zbiorów wyjść  $\{Out(j,k)\}$ , zadeklarowanych przez użytkownika. Macierz liczebności  $[U_{jk}]$  jest wyznaczana również w przeważającej swojej części w sposób algorytmiczny. Dla tych par liczb  $(j,k)$  /patrz tab.1/, dla których występuje niepusty zbiór wyjściowy lub słowo End program przyjmuje  $U_{jk} = 1$ , dla pozostałych par  $(j,k)$  program przyjmuje  $U_{jk} = 0$ . Następnie dokonuje się zmian wartości  $U_{jk} = 1$  na inne dla tych par  $(j,k)$ /w praktyce nielicznych/, dla których użytkownik zadeklaruje inne wartości. Macierze  $[Out_{jk}(1)]$ ,  $[Inp_{jk}(1)]$ ,  $[El_{jk}(1)]$ ,  $[U_{jk}]$  nazwano macierzami charakterystycznymi ESP.

### 3. Macierze stanu, funkcja realizowalności

Symulacja systemu opiera się na analizie stanu systemu w kolejnych jego stanach  $i$ . Liczba elementów  $k$ , dostępnych do rozpoczęcia zdarzenia  $j$  na etapie  $i$  jest określona macierzą  $[N_{jk}(i)]$ . Macierz stanu  $[S_{jk}(i)]$  jest określona wzorem

$$[S_{jk}(i)] = [N_{jk}(i)] - [U_{jk}] \quad /2/$$

$i$  podaje nadwyżkę elementów  $k$  dostępnych do rozpoczęcia czynności  $j$  na etapie  $i$  nad minimalną ich liczbą używaną w tej czynności. Wprowadza się funkcję realizowalności  $F_j(i)$  danej czynności  $j$  na etapie  $i$ . Jeżeli  $F_j(i) = 1$ , czynności  $j$  można rozpocząć, jeżeli  $F_j(i) = 0$ , czynności rozpocząć nie można. Biorąc pod uwagę zależność /2/ warunki realizowalności można zapisać:

$$\begin{aligned} F_j(i) &= 1, \text{ jeżeli } S_{jk}(i) \geq 0 \text{ dla każdego } k = 1 \dots K \\ F_j(i) &= 0, \text{ jeżeli } S_{jk}(i) < 0 \text{ dla } k \in \{k = 1 - K\} \end{aligned} \quad /3/$$

Jeżeli na danym etapie  $i$  następuje rozpoczęcie czynności  $l$  spełniającej warunek  $F_l(i) = 1$ , to wówczas liczba elementów  $k$  dostępnych do rozpoczęcia czynności  $j$  zmniejszy się wg wzoru:

$$N_{jk}(i+1) = N_{jk}(i) - U_{lk} \cdot El_{jk}(1) \quad /4/$$

Biorąc pod uwagę wzór /2/ można też napisać:

$$S_{jk}(i+1) = S_{jk}(i) - U_{lk} \cdot El_{jk}(1) \quad /5/$$



Jeżeli na etapie  $i$  następuje zakończenie czynności  $l$ , to wówczas liczba elementów  $k$  dostępnych do rozpoczęcia czynności  $j$  zwiększy się wg wzoru:

$$N_{jk}(i+1) = N_{jk}(i) + U_{lk} \text{Out}_{jk}(l) \quad /6/$$

Biorąc pod uwagę wzór /2/ można też napisać:

$$S_{jk}(i+1) = S_{jk}(i) + U_{lk} \text{Out}_{jk}(l) \quad /7/$$

W trakcie obliczeń symulacyjnych nie ma potrzeby posługiwania się macierzą  $[N_{jk}(i+1)]$ , wystarczy korzystać z macierzy stanu  $[S_{jk}(i+1)]$ , na podstawie której oblicza się funkcję realizowalności  $F_j(i)$  wg wzoru /3/. W tym celu należy obliczyć wyłącznie macierz stanu początkowego  $[S_{jk}(0)]$ , następnie stosować wzory /5/ i /7/. Macierz stanu początkowego oblicza się ze wzoru /2/ podstawiając macierz liczebności  $[N_{jk}(0)]$ . Program pomaga zalgorytmizować wprowadzanie tych danych przyjmując na początku  $N_{jk}(0) = 0$  dla  $U_{jk} > 0$  i dopiero w następnym kroku użytkownik deklaruje wartości  $N_{jk}(0) \neq 0$ . Przykład danych dla tej macierzy podano w tablicy 3.

Tablica 3 Macierz liczebności dostępnych elementów  $[N_{jk}(0)]$  w stanie  $i=0$

Element		AM	AW1	TOK1	SM	ST1	STM	PAL1	PO1
Czynność	$k \backslash j$	1	2	4	9	10	15	16	18
PAL1, AM $\rightarrow$ AW1	1	1	1					30	270
PAL1, AW1 $\rightarrow$ ST1	3		0			1		0	0
PAL1, AW1 $\rightarrow$ STM	8		0				1	0	
PO1, PAL1 $\rightarrow$ TOK1	15			1				0	0
PAL1, STM $\rightarrow$ SM	20				1		0	0	
PO1, TOK1 *	21			0					0
PAL1, STM *	26				0			0	
PO1, TOK1 $\rightarrow$ PAL1	27			0				0	0
PAL1, SM $\rightarrow$ STM	32				0		0	0	
PAL1, ST1 $\rightarrow$ AW1	33		1			0		0	0
PAL1, STM $\rightarrow$ AW1	38		1				0	0	
PAL1, AW1 $\rightarrow$ AM	45	1	0					0	
AW1 *	93		0						

$N_{16}(0) = 30$  oznacza, że w magazynie znajduje się 30 elementów  $k = 16$  (PAL1). Ponieważ w każdej palecie znajduje się 9 przedmiotów, stąd też  $N_{18}(0) = 270$ .

#### 4. Symulacja systemu

Na każdym etapie  $i$  symulacji istnieje zbiór czynności  $S(i) = \{j: F_j(i) = 1\}$ , które można rozpocząć w czasie  $T_{\text{start } j}(i) < M$ , gdzie  $M$  jest dużą wielkością. Jeżeli dana czynność nie może się rozpocząć na etapie  $i$ , to umownie przyjęto  $T_{\text{start } j}(i) = M$ . Stosownie do przyjętej taktyki starowania ze zbioru  $S(i)$  wybiera się tę czynność, którą należy rozpocząć jako pierwszą i oznacza się  $l_{\text{start}}(i)$ . Czas możliwego rozpoczęcia tej czynności oznaczono  $T_{\text{start}}(i)$ . Na każdym etapie  $i$  istnieje również zbiór czynności  $j$ , które mogą się zakończyć /ponieważ wcześniej się rozpoczęły/ oznaczony  $E(i)$ ; czas zakończenia każdej z nich oznaczono  $T_{\text{end } j}(i)$ . W przypadku gdy dana czynność  $j$  nie może się zakończyć na danym etapie  $i$  /ponieważ wcześniej się nie rozpoczęła/, umownie przyjęto  $T_{\text{end } j}(i) = M$ . Czas najwcześniejszego zakończenia czynności wyznacza się ze wzoru:

$$T_{\text{end}}(i) = \min T_{\text{end } j}(i) \quad /8/$$

Dla celów porządkowych przyjęto, że jeżeli kilka czynności  $E(i)$  posiada ten sam czas zakończenia  $T_{\text{end}}(i)$ , to na etapie  $i$  następuje zakończenie czynności  $l_{\text{end}}(i)$ , która ma najniższy numer.

W trakcie symulacji sprawdza się, czy nastąpi początek, czy koniec czynności  $l$ . Jeżeli:

$$T_{\text{start}}(i) < T_{\text{end}}(i), \quad /9/$$

to na etapie  $i$  następuje realizacja początku czynności  $l = l_{\text{start}}(i)$ . Wówczas czas bieżący  $T$  określa wzór:

$$T = T_{\text{start}}(i) \quad /10/$$

Następnie korzysta się ze wzorów /5/ dla wyznaczenia macierzy stanu  $[S_{jk}(i+1)]$  oraz /3/ dla wyznaczenia funkcji  $F_j(i+1)$ . Następnie oblicza się czasy możliwego rozpoczęcia czynności:

$$T_{\text{start } j}(i+1) = T \quad \text{dla } F_j(i+1) = 1 \quad /11/$$

$$T_{\text{start } j}(i+1) = M \quad \text{dla } F_j(i+1) = 0$$

oraz zakończenia czynności  $l$

$$T_{\text{end } l}(i+1) = T + T_l; \quad /12/$$



gdzie  $T_1$  czas trwania czynności 1. Jeżeli symuluje się awarię, to czas zakończenia czynności 1 jest równy:

$$T_{\text{end } l_b}(i+1) = T_{\text{end } l}(i+1) + b(i+1), \quad /13/$$

gdzie  $b(i+1)$  jest czasem przerwy spowodowanym awarią.

Jeżeli:

$$T_{\text{start}}(i) \geq T_{\text{end}}(i), \quad /14/$$

to na etapie  $i$  następuje realizacja zakończenia czynności  $l = l_{\text{end}}(i)$ . Wówczas czas bieżący  $T$  określa wzór:

$$T = T_{\text{end}}(i) \quad /15/$$

Następnie stosuje się wzór /7/ dla wyznaczenia macierzy stanu  $[S_{jk}(i+1)]$  oraz /3/ dla wyznaczenia funkcji  $F_j(i+1)$ . Następnie oblicza się czasy możliwego rozpoczęcia czynności wg wzorów /11/ oraz przyjmuje czas zakończenia czynności  $l(i+1)$ :

$$T_{\text{end } l}(i+1) = M \quad /16/$$

Tablica 4 podaje wyniki pierwszych etapów obliczeń dla przyjętego ESP. Tablica 5 podaje wyniki obliczeń dla etapu  $i = 905$  i  $906$ . Fakt niezrealizowania zakończenia czynności  $j = 22$  w przewidzianym czasie wskazuje na wygenerowanie awarii wg wzoru /13/. W systemie rzeczywistym powyższe wartości byłyby wynikiem niepotwierdzenia zakończenia czynności  $j = 22$  w przewidzianym czasie. W tablicach 4 i 5 podano także liczbę zrealizowanych początków  $N_{\text{start } j}(i)$  i zakończeń  $N_{\text{end } j}(i)$  czynności na etapie  $i$ , wykorzystywanych dla taktyki systemu.

Tablica 4 Zdarzenia realizowalne na etapach $i = 0, 1, 2, 3, 4$							
i	T	Nazwa czynności	j	$T_{\text{start } j}(i)$	$T_{\text{end } j}(i)$	$N_{\text{start } j}(i)$	$N_{\text{end } j}(i)$
0	0	PAL1, AM → AW1 <sup>1</sup>	1	0	M	0	0
		PAL1, AM → AW2	2	0	M	0	0
		PAL2, AM → AW1	47	0	M	0	0
		PAL2, AM → AW2	48	0	M	0	0
1	0	PAL1, AM → AW1	1	M	3	1	0
		PAL1, AM → AW2	2	0	M	0	0
		PAL2, AM → AW2 <sup>1</sup>	48	0	M	0	0

1	2	3	4	5	6	7	8
2	0	PAL1, AM → AW1 <sup>1</sup>	1	M	3	1	0
		PAL2, AM → AW2	48	M	3	1	0
3	3	PAL1, AW1 → ST1	3	3	M	0	0
		PAL1, AW1 → ST2	4	3	M	0	0
		PAL1, AW1 → ST3	5	3	M	0	0
		PAL1, AW1 → ST4	6	3	M	0	0
		PAL1, AW1 → ST5	7	3	M	0	0
		PAL2, AM → AW2 <sup>1</sup>	48	M	3	1	0
4	3	PAL1, AW1 → ST1 <sup>1</sup>	3	3	M	0	0
		PAL1, AW1 → ST2	4	3	M	0	0
		PAL1, AW1 → ST3	5	3	M	0	0
		PAL1, AW1 → ST4	6	3	M	0	0
		PAL1, AW1 → ST5	7	3	M	0	0
		PAL2, AW2 → ST1	49	3	M	0	0
		PAL2, AW2 → ST2	50	3	M	0	0
		PAL2, AW2 → ST3	51	3	M	0	0
		PAL2, AW2 → ST4	52	3	M	0	0
		PAL2, AW2 → ST5	53	3	M	0	0

1 zdarzenie zrealizowane na etapie 1

Tablica 5 Zdarzenia realizowalne na etapach 1 = 905,906

i	T	Nazwa czynności	j	$T_{start\ j(i)}$	$T_{end\ j(i)}$	$T_{start\ j(i)}$	$T_{end\ j(i)}$
905	624,5	PO1, PAL1 → TOK3 <sup>1</sup>	17	624,5	M	4	4
		PO1, TOK2 *	22	M	179,5	5	4
		PO1, TOK5 *	25	M	636,5	6	5
		PAL2, STM → SM	66	624,5	M	6	6
		PO2, TOK4	70	M	628	8	7
		AW1 *	93	624,5	M	25	25
		AW2 *	94	624,5	M	25	25
906	624,5	PO1, PAL1 →	17	M	625	5	4
		PO1, TOK2 *	22	M	179,5	5	4
		PO1, TOK5 *	25	M	636,5	6	5
		PAL2, STM → SM <sup>1</sup>	66	624,5	M	6	6
		PO2, TOK4	70	M	628	8	7
		AW1 *	93	624,5	M	25	25
		AW2 *	94	624,5	M	25	25

1 - zdarzenie zrealizowane

179,5

czas obliczeniowy zakończenia  $T_{end\ j(i)}$  i większy od czasu bieżącego  $T = 624,5$  wskazuje na awarię.



## 5. Zakończenie

Jednym z istotnych celów wprowadzania nowego modelu macierzowego było ominięcie dodatkowych ograniczeń dla strategii i taktyki sterowania ESP. Cel ten został wypełniony, ponieważ na każdym etapie i dysponuje się wszystkimi możliwościami czynności, które mogą się rozpocząć, lub zakończyć /tablice 4 i 5/. Model pozwala na elastyczne wykorzystanie wszystkich możliwości działania systemu sterowania, a więc optymalnego doboru strategii i taktyki ESP.

W poprzednich pracach [2] omówiono metodę symulacji modularnej. W modelu prezentowanym obecnie może być ona zrealizowana w znacznie dogodniejszy sposób. Sygnał rozpoczęcia zdarzenia może być traktowany jako sygnał uruchamiający procedurę symulacji podsystemu albo sterowania podsystemu rzeczywistego. Proponowany model jest więc "otwarty" na już opracowane procedury obliczeniowe.

## LITERATURA

- [1] Cyklis J.: "Symulacja elastycznych systemów produkcyjnych z wykorzystaniem macierzy stanu", V Krajowa Konferencja Automatyzacji Dyskretnych Procesów Przemysłowych, Kozubnik, 1986.
- [2] Cyklis J.: "Towards Simple Simulation of FMS", Monografie Politechniki Krakowskiej, Nr 58, 1987.
- [3] Hartley J.: "FMS at Work", IFS /Publications/, UK, 1984.
- [4] Kowalowski H.: "Automatyzacja dyskretnych procesów przemysłowych", WNT, Warszawa, 1984.
- [5] Inter. Conf. Flexible Manufacturing Systems, London, 1983.
- [6] The 17th Annual Simulations Symposium, Florida, 1984.
- [7] 1st. Inter. Conf. Simulation in Manufacturing, Stratford upon-Avon, UK., 1985.

Recenzent: Prof.dr inż.H.Kowalowski

Wpłynęło do Redakcji do 1988-04-30.

## АЛГОРИТМ СИМУЛЯЦИИ ГИБКИХ АВТОМАТИЗИРОВАННЫХ УЧАСТКОВ

### Резюме

Алгоритм симуляции гибких автоматизированных участков основан на расчёте матрицы состояния  $[S_{jk}(i)]$  и текущего времени  $T(i)$  на каждом шагу  $i$  изменения системы. Матрица  $[S_{jk}(i)]$  определяет разность между числом элементов  $k$  готовых к началу действия  $j$  и числом элементов  $k$

употребляемых в этом действии. Матрица  $[S_{jk}(i)]$  и текущее время  $T(i)$  используются для определения начала или конца действия на  $i$ -ом шагу. Соответственно к последнему событию рассчитывается матрица состояния  $[S_{jk}(i+1)]$  и время  $T(i+1)$  для следующего шага  $(i+1)$ .

#### ALGORITHM OF FLEXIBLE MANUFACTURING SYSTEMS SIMULATION

##### S u m m a r y

The algorithm of FMS simulation consists in calculation of state matrix  $[S_{jk}(i)]$  and current time  $T(i)$  at each stage  $i$  of the change of the system. The matrix  $[S_{jk}(i)]$  describes the difference between the number of elements  $k$  ready to start the activity  $j$  and the number of elements  $k$  used in this activity. The matrix  $[S_{jk}(i)]$  and current time  $T(i)$  are used for determination of the start or the end of an activity at the stage  $i$ . According to the last event the state matrix  $[S_{jk}(i+1)]$  and the time  $T(i+1)$  is calculated for the next stage  $(i+1)$ .