ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1975
Seria: Automatyka z. 34 Nr kol. 458

Zbigniew Czech

Instytut Kompleksowych Systemów Sterowania

SYSTEMY OPERACYJNE MASZYN STERUJĄCYCH PROCESAMI O CHARAKTERZE SEKWENCYJNYM

Streszczenie. Praca dotyczy systemów sterowania procesami o charakterze sekwencyjnym.
Podano opis matematyczny procesu sekwencyjnego, w oparciu o model sieciowy. Sformułowano zadania sterowania. Omówiono program zarządzający sterowaniem oraz scharakteryzowano strukturę programów użytkownika. Dokonano analizy cech języka sterowania sekwencyjnego i budowę jego translatora.

1. WSTĘP

Jedną z grup procesów przemysłowych, podobnych ze względu na sterowanie, jakie w nich występuje, są procesy o sterowaniu sekwencyjnym. Przykładem mogą tu być procesy kampanijne, często spotykane w chemii. W procesach tych po załadowaniu do reaktora określonej porcji surowców (wsadu) następuje obróbka polegająca na przeprowadzeniu ściśle ustalonej sekwencji czynności (podgrzewanie, mieszanie, reakcja itp.), przy utrzymaniu określonych parametrów na zadanych poziomach.

Podobnym typem sterowania charakteryzują się procesy rozruchu i zatrzymywania, które ogólnie polegają na kolejnym załączaniu lub wyłączaniu poszczególnych urządzeń procesu przy spełnieniu uwarunkowań logicznych, wynikający z wymagań kolejności załączania, jak również uwarunkowań czasowych.

Opracowanie niniejsze dotyczy realizacji sterowania sekwencyjnego przez jednoprocesorowy system sterujący.

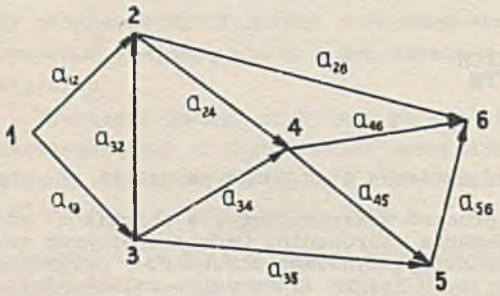
2. CHARAKTERYSTYKA I OPIS MATEMATYCZNY PROCESÓW SEKWENCYJNYCH

W rozdziale tym zostanie podana definicja i model procesu o sterowaniu sekwencyjnym. Na podstawie tego modelu sformułowane zostaną zadania sterowania. W części końcowej rozdziału zostanie omówiona struktura programów użytkowych. Programy te ujmują algorytmy sterowania otrzymane w wyniku rozwiązania zadań sterowania.

2.1. Określenie i model procesu sekwencyjnego

Pojęcia podstawowe [1] [7]

Niech U będzie skończonym zbiorem punktów ponumerowanych liczbami naturalnymi 1,2,...,n, zaś $\Gamma(u)$ zbiorem uporządkowanych par (i,j), gdzie $i \in U$ $j \in U$. Układ zawierający oba te zbiory $G = \{U, \Gamma(u)\}$ nazwiemy grafem. Graf składa się więc z punktów zbioru U (wierzchołków grafu) oraz par uporządkowanych zbioru $\Gamma(u)$ (łuków grafu). Łuk (i,j) ma początek w punkcie i oraz koniec w punkcie j. Graf, w którym istnieje jeden taki punkt, w którym żaden z łuków się nie kończy (punkt wejściowy) oraz jeden taki punkt, w którym żaden z łuków się nie zaczyna (punkt wyjściowy) i w którym każdemu łukowi (i,j) została przyporządkowana pewna nieujemna liczba rzeczywista a_{ij}, nazwiemy siecią (rys. 1.1).



Rys. 1.1. Schemat sieci

Powyższy schemat przedstawia sieć zorientowaną, ponieważ każdy łuk posiada określony kierunek albo orientację. W sieciach niezorientowanych zbiór $\Gamma(u)$ składa się z nieuporządkowanych par punktów zbioru U. Istnieją sieci mieszane, w których pewne łuki są zorientowane, a inne niezorientowane. W pracy niniejszej będą rozważane wyłącznie sieci zorientowane.

Proces o sterowaniu sekwencyjnym

Analiza sterowania procesem składającym się z pewnej liczby obiektów pozwala na wyodrębnienie czynności i zdarzeń. Pod czynnością będziemy rozumieli przebieg pewnych operacji technologicznych, związanych z pojedynczym obiektem np. nagrzewania mas, reakcje chemiczne, napełnianie zbiorników. Zdarzenie określimy jako koniec jednych czynności i początek następnych.

Niech własności dynamiczne obiektów opisane będą równaniami różniczkowymi

$$\dot{\underline{x}}_{ij} = \underline{f}(\underline{x}_{ij}, \underline{u}_{ij}), \quad \underline{u}_{ij} \in D, \tag{1}$$

gdzie

$\underline{x}_{ij}$ - wektor stanu (fazowy) obiektu,
$\underline{u}_{ij}$ - wektor sterowania,
D - obszar sterowań dopuszczalnych,
i,j - wskaźniki identyfikujące obiekt.

Czynnością w przypadku tak określonego pojedynczego obiektu jest przejście jego wektora fazowego od stanu początkowego $\underline{x}_{ij}^{p}$ do stanu końcowego $\underline{x}_{ij}^{k}$. Zdarzeniem jest osiągnięcie stanu $\underline{x}_{ij}^{k}$.

Dla każdej czynności określony jest wskaźnik jakości będący funkcją drogi przejścia od $\underline{x}_{ij}^{p}$ do $\underline{x}_{ij}^{k}$:

$$a_{ij} = \underline{g}\,(\underline{x}_{ij}, \underline{u}_{ij}) \qquad (2)$$

D e f i n i c j a 1

Przez proces o sterowaniu sekwencyjnym będziemy rozumieć proces, w którym:

a) istnieją logiczne powiązania pomiędzy czynnościami realizowanymi na obiektach opisanych równaniami (1); zależności te wynikają z narzuconej kolejności wykonywania operacji technologicznych procesu,
b) sygnały sterujące procesem mają charakter binarny (zero-jedynkowy).

W dalszym ciągu dla prostoty proces o sterowaniu sekwencyjnym będziemy nazywać procesem sekwencyjnym.

Model procesu sekwencyjnego

Jeżeli zdarzeniom procesu sekwencyjnego będą odpowiadać wierzchołki, czynnościom - łuki, a wskaźniki jakości (2) zostaną przyporządkowane określonym łukom, to powstanie sieć, która może być modelem procesu sekwencyjnego.

Model taki dobrze odzwierciedla relację poprzedzania i następowania między czynnościami, jak również uwzględnia jakość realizacji tych czynności.

W sterowaniu sekwencyjnym wskaźnikami jakości są najczęściej czasy realizacji poszczególnych czynności. Orientacja łuków sieci modelującej proces jest zgodna w tym przypadku z kierunkiem upływu czasu. Sieci takie można określić jako zorientowane czasowo.

Wprowadźmy dodatkowo kilka pojęć.

D e f i n i c j a 2

Terminem zaistnienia zdarzenia j będziemy nazywać moment zakończenia wszystkich czynności kończących się tym zdarzeniem i oznaczać symbolem T_j.

D e f i n i c j a 3

Ciąg łuków, w którym początek łuku następnego jest jednocześnie końcem poprzedniego $(i_1, i_2), (i_2, i_3) \ldots (i_k, i_{k+1})$ nazwiemy drogą od i_1 do i_{k+1}.

Jeżeli zachodzi $i_1 = i_{k+1}$, droga taka jest cyklem. Sieci zorientowane czasowo nie mogą zawierać cykli.

D e f i n i c j a 4

Przez czas przejścia $t_{i_1 i_{k+1}}$ drogi (i_1, i_{k+1}) utworzonej z ciągu czynności $(i_1,i_2)...(i_k, i_{k+1})$ będziemy rozumieli sumę czasów realizacji wszystkich czynności ciągu definiującego daną drogę. Czas trwania czynności (i,j) będziemy oznaczać przez t_{ij}.

D e f i n i c j a 5

Zdarzenie i nazywamy zdarzeniem poprzedzającym zdarzenie j, jeśli istnieje droga (i,j). Jeśli istnieje czynność (i,j), to zdarzenie i nazywamy bezpośrednio poprzedzającym zdarzenie j, zaś zdarzenie j bezpośrednio następującym po zdarzeniu i.

2.2. Postawienie zadań sterowania i niektóre rozwiązania

Dla procesu o obiektach opisanych równaniami (1) i globalnym wskaźniku jakości

$$I = F\left[s_{ij}(\underline{x}_{ij}, \underline{u}_{ij})\right] \qquad (3)$$

$$i \in U, \quad j \in U$$

można sformułować kilka podstawowych zadań sterowania.

Zadanie 1

Dany jest proces o modelu sieciowym określonym przez n - elementowy zbiór zdarzeń U oraz m - elementowy zbiór czynności $\Gamma(u)$. Obiekty opisane są równaniami (1).

Określić sterowania $\underline{u}^o_{ij} \in D$, $(i,j) \in \Gamma(u)$ realizujące wszystkie czynności procesu i minimalizujące globalny wskaźnik jakości I.

$$\min I = F\left[s_{ij}(\underline{x}^o_{ij}, \underline{u}^o_{ij})\right] \qquad (4)$$

$$\underline{u}_{ij} \in D \qquad i \in U$$

$$j \in U$$

Tak postawione zadanie jest ogólnym zadaniem sterowania dla procesów o modelach sieciowych. Należy dodać, że w pewnych przypadkach żąda się spełnienia warunku (4) przy dodatkowych ograniczeniach. Ograniczenia te mogą dotyczyć czasów realizacji czynności, ich wskaźników jakości, wektorów fazowych obiektów czy też innych wielkości związanych bezpośrednio ze sterowaniem. W dalszym ciągu rozpatrzmy zadanie sterowania dla procesu sekwencyjnego.

Załóżmy, że sygnały sterujące obiektów mają charakter binarny, a więc przyjmują wartości ze zbioru $\{0,1\}$. Wskaźnikami jakości czynności niech będą ich stałe czasy realizacji.

Zadanie 2

Dany jest proces o modelu sieciowym określonym przez n - elementowy zbiór zdarzeń U oraz m - elementowy zbiór czynności $\Gamma(u)$, Każdej czynności przyporządkowany jest stały czas jej realizacji t_{ij} = const.

Określić momenty wystąpienia wszystkich zdarzeń początkowych czynności tak, aby zapewnić

$$\min_{T_j \; j \in U,} (T_n - T_1) = t^{o}_{1n} \tag{5}$$

gdzie:

T_n, T_1 - terminy wystąpienia odpowiednio zdarzenia wyjściowego i wejściowego.

Globalnym wskaźnikiem jakości jest tu czas realizacji całego procesu. Wyżej sformułowane zadanie jest typowym zadaniem sterowania sekwencyjnego. Istnieje tu analogia do problemów występujących przy realizacji przedsięwzięć opisanych modelami sieciowymi. W problemach tych, rozwiązywanych metodami PERT [2] zakłada się, że czasy realizacji czynności są wielkościami przypadkowymi.

Rozwiązanie zadania 2 można otrzymać, stosując metodę drogi krytycznej. Metoda ta polega na określeniu najwcześniejszych możliwych $T_k^{(w)}$, $k \in U$ i najpóźniejszych dopuszczalnych $T_k^{(p)}$ momentów zaistnienia zdarzeń [7].Oznaczmy przez A(j) zbiór wszystkich zdarzeń,które bezpośrednio poprzedzają zdarzenie j. Najwcześniejszy możliwy moment wystąpienia zdarzenia j oblicza się według wzoru:

$$T_j^{(w)} = \max_{i \in A(j),} (T_i^{(w)} + t_{ij}) \tag{6}$$

idąc od zdarzenia wejściowego w kierunku końca sieci. Najpóźniejszy dopuszczalny moment wystąpienia zdarzenia i znajduje się, prowadząc obliczenia od zdarzenia wyjściowego do początku sieci, według wzoru:

$$T_j^{(p)} = \min_{j \in B(i),} (T_j^{(p)} - t_{ij}) \tag{7}$$

gdzie:

B(i) - zbiór wszystkich zdarzeń, które bezpośrednio następują po zdarzeniu i.

Przy obliczaniu momentów $T_j^{(w)}$ należy założyć $T_1^{(w)} = 0$, zaś przy obliczaniu momentów $T_i^{(p)}$ przyjąć $T_n^{(w)} = T_n^{(p)}$.

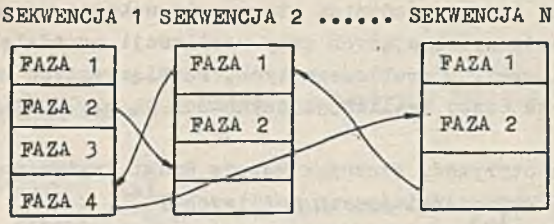
Ze względu na założenie $t_{ij} = \text{const}$, minimalny czas realizacji całego procesu jest stały i określony przez najwcześniejszy możliwy moment wystąpienia n-tego zdarzenia $\min(T_n - T_1) = T_n^{(w)}$. Czas ten jest także czasem ścieżki krytycznej tzn. tej z dróg sieci, na której czas przejścia jest najdłuższy. Ponieważ dla czynności leżących na ścieżce krytycznej zachodzi

$$T_j^{(p)} - T_i^{(w)} - t_{ij} = 0, \tag{8}$$

to początek ich realizacji powinien przypadać w momentach $T_i^{(w)}$. Pozostałe czynności dysponują zapasem czasu określonym lewą stroną równania (8).

2.3. Algorytmy sterowania

Algorytmy otrzymane po rozwiązaniu odpowiednich zadań sterowania procesem w stanach normalnych ujęte są przez programy użytkowe. Ze względu na swą specyfikę dzielą się one na sekwencje, które z kolei złożone są z faz. Strukturę programów przedstawia rys. 1.2. Poszczególne sekwencje najczęściej przyporządkowane są określonym obiektom. W przypadku, gdy obiekty pracują jednocześnie ich programy sterowania powinny być wykonywane przez jednostkę centralną równolegle. Sekwencje w trakcie wykonywania są często zawieszane w oczekiwaniu na spełnienie określonych warunków, np. osiągnięcie zadanych parametrów przez obiekt, zakończenie innej sekwencji, albo jej określonej fazy (rys. 1.2). W przypadku wystąpienia awarii w procesie, powinny zostać uruchomione programy awaryjne, które po rozpoznaniu sytuacji podejmą kroki zaradcze. Struktura programów awaryjnych jest taka sama, jak opisana powyżej struktura programów sterowania w stanach normalnych.



Rys. 1.2. Struktura programów użytkowych

Wymienione przyczyny powodują, że zachodzi konieczność koordynacji wykonywania programów użytkowych.

3. OPROGRAMOWANIE CENTRALNEJ JEDNOSTKI STERUJĄCEJ

3.1. Ogólna charakterystyka systemu operacyjnego

Wydaje się, że określenie systemu operacyjnego jako zbioru środków programowych przeznaczonych do najbardziej ekonomicznego rozdziału takich zasobów komputera jak: czas jednostki centralnej, pamięć, urządzenia wejścia-wyjścia i przerwania, podane w [8], jest trafne. Budowa systemu operacyjnego zależy od rodzaju spełnianych zadań określonych przez dziedzinę zastosowań. W niniejszej pracy omówiono pewne aspekty podziału czasu jednostki centralnej w systemach operacyjnych przeznaczonych dla sterowania sekwencyjnego. Szczególną uwagę zwrócono na program zarządzający, który jest odpowiedzialny za poprawne wykonanie programów użytkownika.

Ze względu na to, że podział czasu jednostki centralnej na poszczególne programy sterowania zależy od sytuacji w procesie, w dalszym ciągu zamiast określenia podziału czasu będzie używane określenie koordynacji sekwencji (programów).

3.2. Program zarządzający sterowaniem sekwencyjnym

3.2.1. Zadania i ogólna struktura programu zarządzającego

Program ten (zależny od czasu) uruchamiany jest w każdym cyklu systemu operacyjnego. Jego podstawowym zadaniem jest zarządzanie wykonywaniem programów sterowania użytkownika. Zarządzanie obejmuje następujące funkcje:

- koordynacja sekwencji,
- inicjowanie komutacji danych, związanych z sekwencją,
- sprawdzanie poprawności wykonania określonych akcji sterowania.

Zanim omówimy poszczególne funkcje, scharakteryzujemy strukturę programu zarządzającego i jego współdziałanie z systemem operacyjnym.

Zgodnie z [10] każdy program sterowania użytkownika może być uważany za jednowymiarowe źródło zgłoszeń. Zgłoszenia te są typu programowego. Zbiór programów sterowania zawiera N programów (sekwencji), opisujących stany normalne procesu $s_1, \ldots\ldots, s_N$.
Ponadto występuje K programów awaryjnych $s_{N+1}, \ldots\ldots\ldots, s_{N+K}$. W szczególności ilość sekwencji awaryjnych może być równa ilości sekwencji dotyczących sterowania procesu w stanach normalnych.

Przy rozwiązywaniu zadań sterowania o charakterze sekwencyjnym, powstaje problem organizacji przesyłu danych dwustanowych z procesu do jednostki centralnej (JC). Możliwe są tu dwa rozwiązania.

Pierwsze z nich zakłada, że przekazywanie danych odbywa się z inicjatywy procesu. Oznacza to, że sygnały określające stan procesu, albo inaczej - stan realizacji czynności sieci będącej modelem procesu, podane są na układ przerwań priorytetowych JC. Zmiana dowolnego z sygnałów powoduje przerwanie piorytetowe pracy JC informujące o zmianie stanu procesu.

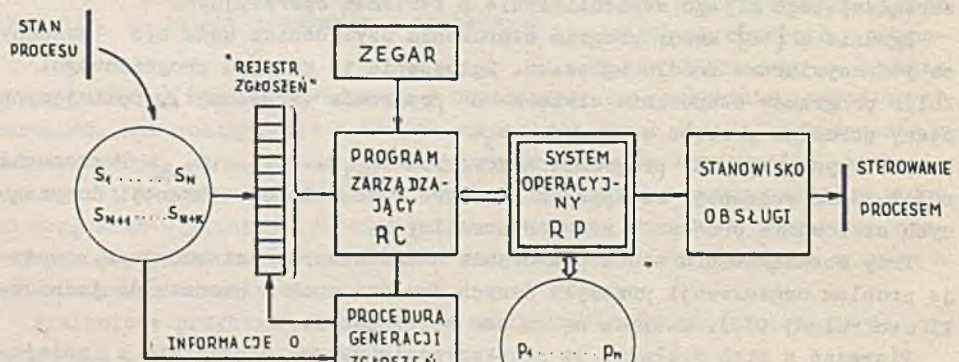
W drugim rozwiązaniu zakłada się, że dane pobierane są z inicjatywy JC. Z określoną częstotliwością są one wprowadzane do JC a następnie badane w celu stwierdzenia, czy stan procesu, na który składają się stany poszczególnych czynności, uległ zmianie.

Oba rozwiązania posiadają swoje wady i zalety. Pierwsze rozwiązanie zapewnia maksymalne wykorzystanie czasu pracy JC, jest natomiast wrażliwe na zakłócenia, co dla procesów przemysłowych jest niezwykle istotne. Drugie rozwiązanie, przy większej odporności na zakłócenia, pogarsza wykorzystanie czasu pracy JC wskutek konieczności okresowego badania stanu procesu. Należy dodać, że w przypadku dużych procesów, których stan jest określony przez zbiór kilkuset sygnałów, pierwsze rozwiązanie jest praktycznie trudne do realizacji.

W pracy niniejszej przyjęto, że przesył danych o charakterze dwustanowym z procesu do JC następuje według rozwiązania drugiego. Założono, że poszczególne sekwencje użytkownika są aktywizowane z określoną częstotliwością. Każda z sekwencji kontroluje stan przyporządkowanej sekcji procesu oraz realizuje zadania sterowania.

Podziału czasu procesora na poszczególne sekwencje użytkownika dokonuje program zarządzający zgodnie z przyjętym regulaminem lokalnym. Program zarządzający wraz ze zbiorem programów podstawowych i specjalnych [10] systemu $p_1,\ldots,p_n$ jest aktywizowany przez system operacyjny. Momenty aktywizacji tych programów wynikają z przyjętego, centralnego regulaminu priorytetowego.

Przy takich założeniach podział czasu procesora na programy sterowania użytkownika jest hierarchiczny i odbywa się poprzez program zarządzający (rys. 2.1).



Rys. 2.1. Współpraca systemu operacyjnego z programem zarządzającym RC - regulamin cykliczny, RP - regulamin priorytetowy $\{s_1,\ldots,s_N\}$ - zbiór programów sterowania procesem w stanach normalnych, $\{s_{N+1},\ldots,s_{N+K}\}$ - zbiór programów awaryjnych, $\{p_1,\ldots,p_n\}$ - zbiór programów podstawowych i specjalnych systemu

3.2.2. Stany sekwencji

Z każdą sekwencją związany jest zbiór informacji zawarty w bazie danych. Informacje te są wykorzystywane i modyfikowane przez program zarządzający, programy użytkowe, jak również mogą być zmieniane przez operatora. Dane te określają dynamiczny stan sekwencji w danej chwili sterowania (rys. 2.2). Omówimy znaczenie poszczególnych danych. Numer sekwencji służy do jej identyfikacji przez program zarządzający. Słowo stanu określa programowy stan sekwencji. Zawiera ono następujące informacje.

	NR SEKWENCJI
X	SŁOWO STANU SEKWENCJI
	ZEGAR
Y	ADRES FAZY BIEŻĄCEJ
V	ADRES FAZY AWARYJNEJ
	ADRES BLOKU KONTAKTÓW

Rys. 2.2. Dane określające stan sekwencji

1. Stan normalny

Jeśli sekwencja znajduje się w stanie normalnym, jest aktywizowana od adresu fazy bieżącej. Pod fazą bieżącą rozumie się fazę, która ostatnio była (w danej sekwencji) wykonywana, ale nie została zakończona.

2. Stan "stop"

Sekwencje znajdujące się w tym stanie nie są aktywizowane. Pozwala to na wyłączenie określonych obiektów spod sterowania.

3. Stan "awaryjny"

Stany awaryjne, podobnie jak stany normalne pracy, wymagają pewnego postępowania, które użytkownik określa programem awaryjnym. Programy te, zawierające rozpoznanie sytuacji i podjęcie określonych decyzji, są uruchamiane przez program zarządzający w przypadku, gdy sekwencja jest w stanie awaryjnym. Aktywizacja następuje od adresu fazy awaryjnej.

4. Stan "gotowy"

W stan ten wprowadzane są sekwencje, które mają być w danym momencie aktywizowane.

5. Stan "aktywny"

Informacja ta jest przeznaczona wyłącznie dla programu zarządzającego i określa, która sekwencja aktualnie jest wykonywana.

6. Stan "zawieszony"

W tym stanie znajdują się sekwencje oczekujące na realizację operacji we/wy.

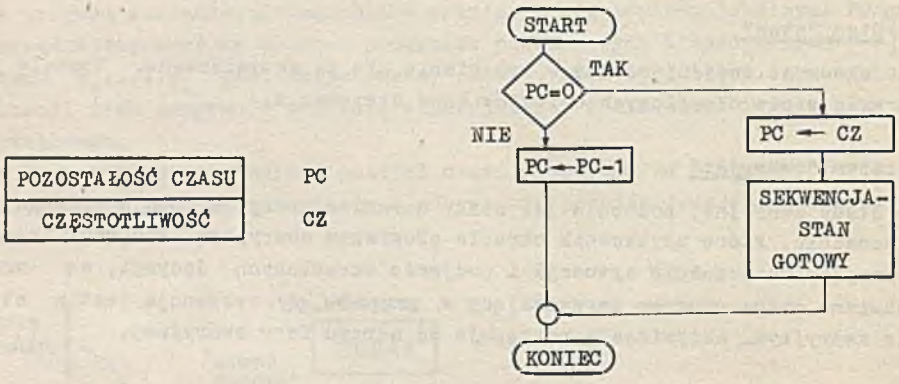
3.2.3. Koordynacja sekwencji

Na koordynację sekwencji sterowania składa się wybór regulaminu obsługi sekwencji oraz odpowiednia konstrukcja procedury generacji zgłoszeń. Zgłoszenie rozumiane jest jako przejście programu w stan "gotowy".

Z każdą sekwencją związany jest zegar programowy. Zegary te obsługiwane są przez procedurę generacji zgłoszeń. Ich budowę oraz algorytm obsługi przedstawia rys. 2.3.

Uwaga: zegary sekwencji będących w stanie "stop" są pomijane.

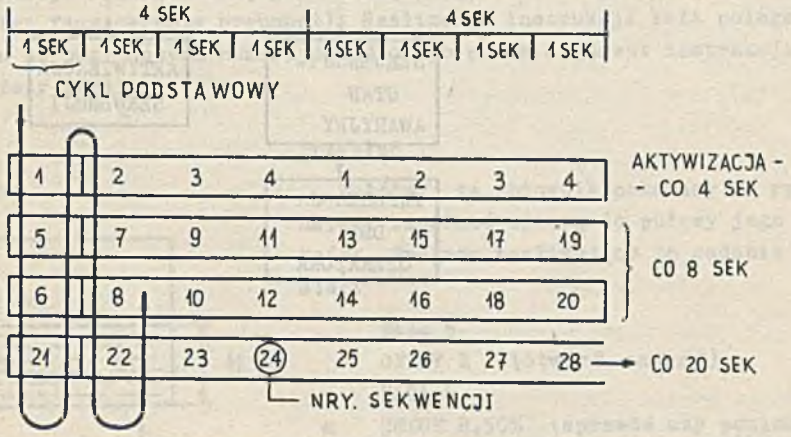
Generacja zgłoszenia następuje co określony odcinek czasu, zależny od częstotliwości zegara. Jednakowy priorytet wszystkich sekwencji sterowania pozwala na przyjęcie najprostszego regulaminu cyklicznego, zapewniającego obsługę wszystkich programów będących w stanie "gotowy". Procedura generacji zgłoszeń musi dysponować informacjami o wszelkich stanach awaryjnych procesu dla wyboru odpowiedniego programu z grup $s_1,\ldots,s_N$ i s_{N+1}, ..., s_{N+K}. Informacje te przekazywane są z programów użytkownika w trakcie realizacji sterowania. Generacja zgłoszeń wymaga również przyjęcia określonych częstotliwości zegarów. Jakie są przesłanki tego wyboru?



Rys. 2.3. Zegar programowy i algorytm jego obsługi

W przypadku dużej szybkości procesora i przy stosunkowo małej liczbie sekwencji, każdą z nich można by aktywizować co cykl systemu operacyjnego. Byłoby to rozwiązanie najprostsze a jednocześnie zapewniające minimalne opóźnienia wykonania poszczególnych programów. W praktyce tylko niewielka część sekwencji wymaga dużej częstotliwości aktywizacji. Są to sekwencje związane z czynnościami krytycznymi i podkrytycznymi procesu. Każde opóźnienie na tych drogach natychmiast odbija się na czasie realizacji całości procesu. Sekwencje alarmowe wymagają najwyższej częstotliwości aktywizacji. Załóżmy więc, że istnieje kilka podstawowych częstotliwości, z którymi programy mogą być aktywizowane. Powstaje problem, w jaki sposób ustalić kolejkę programów tak, aby każdy z nich był powtarzany z określoną czę-

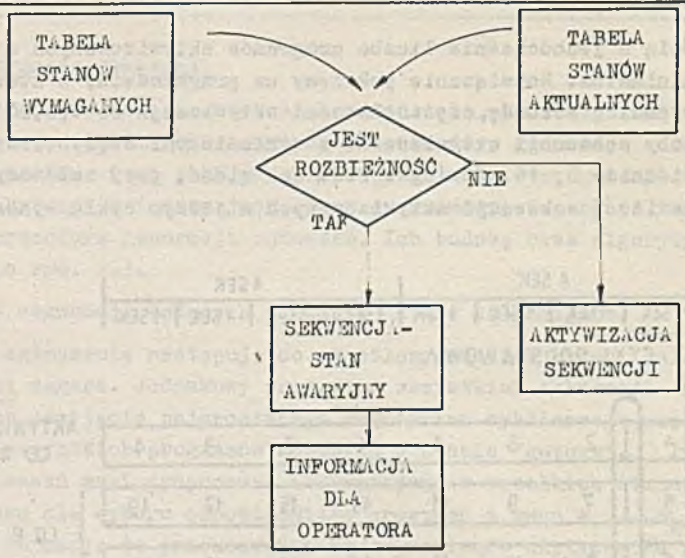
stotliwością a jednocześnie liczba programów aktywizowanych w jednym cyklu była minimalna. Rozwiązanie pokażemy na przykładzie, w którym cykl podstawowy wynosi 1 sekundę, częstotliwości aktywizacji co 4,8,20 sek a maksymalne liczby sekwencji aktywizowane z wymienionymi częstotliwościami wynoszą odpowiednio: 8, 16, 20 (rys. 2.4.) Jak widać, przy założonych danych maksymalna ilość sekwencji aktywizowanych w jednym cyklu wynosi 4.



Rys. 2.4. Koordynacja sekwencji

3.2.4. Nadzór nad poprawnością realizacji akcji sterowania

Dla zapewnienia poprawności sterowania procesem, program zarządzający w każdym swym cyklu realizuje programy diagnostyczne. Jednym z nich jest program sprawdzania poprawności działania kontaktów. Zakłada się, że każde urządzenie (zawór, silnik) sterowane z systemu, w przypadku otrzymania rozkazu zmiany stanu, potwierdza jego wykonanie sygnałem zwrotnym. Przed wykonaniem programu określonej sekwencji program zarządzający w oparciu o adres bloku kontaktów zawarty w bazie danych (rys. 2.2) inicjuje ich komutację. Sygnały te, pobierane przez jednostkę sterującą, grupowane są w tabelę stanów aktualnych. Oprócz niej istnieje tabela stanów wymaganych, modyfikowana po każdym rozkazie zmiany stanu określonego urządzenia procesu. Przed realizacją programów sekwencji tabele są porównywane. W przypadku wykrycia rozbieżności określona sekwencja przechodzi w stan awaryjny a informacja o tym przekazywana jest na zewnątrz systemu (rys. 2.5).



Rys. 2.5. Sprawdzanie poprawności działania urządzeń procesu (zaworów, silników)

3.3. Charakterystyka języka sterowania sekwencyjnego i jego translatora

Ujęcie algorytmów sterowania sekwencyjnego poprzez programy użytkowe wymaga zastosowania określonego języka. Analiza typowych problemów sterowania sekwencyjnego pozwala na określenie podstawowych grup instrukcji, jakie język ten powinien posiadać. Do podstawowych grup instrukcji należą:

- instrukcje sterujące; instrukcje te umożliwiają sterowanie kolejnością wykonywania faz programu, jak również mają wpływ na stan sekwencji;

instrukcje dotyczące:

- urządzeń binarnych tzn. urządzeń o dwóch stanach: otwarty, zamknięty np. zawór;
- zmiennych analogowych; instrukcje te pozwalają między innymi badać wartość poszczególnych zmiennych;
- pętli DDC; grupa tych instrukcji służy do otwierania/zamykania poszczególnych pętli DDC, zmian wartości zadanych pętli itp.;
- zegarów programowych; zegary pozwalają na kontrolę upływającego czasu; instrukcje służą do ustawiania i testowania wartości zegarów;
- przełączników programowych pozwalających na różne wykonywanie danego programu sterowania w zależności od stanu przełącznika.

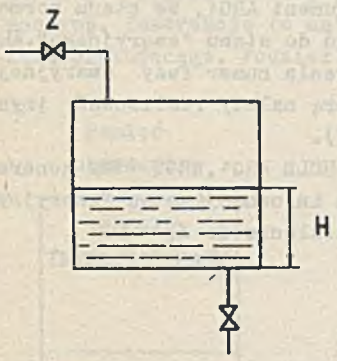
Omówimy obecnie podstawowe instrukcje sterujące, jakie powinien posiadać język sterowania sekwencyjnego.

SEKWENCJA ARG (ARG - argument)

Instrukcja ta jest przeznaczona wyłącznie dla translatora języka. Określa ona początek sekwencji o numerze określonym przez argument instrukcji.

FAZA ARG

Instrukcja kończąca fazę poprzednią programu i rozpoczynająca następną (numer tej fazy określa argument). Realizacja instrukcji FAZA polega na wpisaniu do bazy danych numeru określonego przez argument instrukcji jako numeru fazy bieżącej.



Rys. 2.6. Zbiornik z przykładu 1

Przykład 1

Załóżmy, że zbiornik pokazany na rys.2.6 należy zapełnić cieczą do połowy jego wysokości. Program realizujący to zadanie ma postać:

```
        FAZA 5
        OPENV Z    (otwór Z zawor Z)
        FAZA 6
a       CHECK H,50%  (sprawdź czy poziom jest
                      równy 0.5 H_MAX)
        WAIT   (czekaj; skok do systemu)
        FAZA 7
```

Wykonanie instrukcji FAZA polega na realizacji dwóch elementarnych operacji:

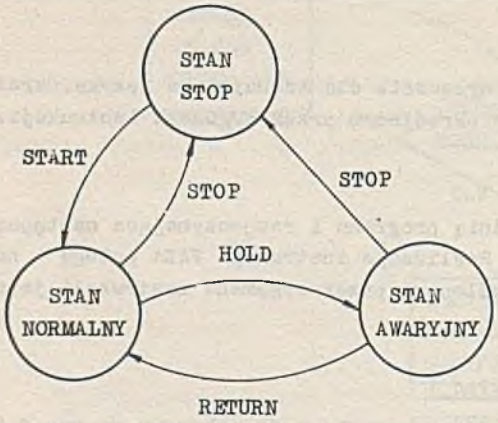
FAZA 6 ≡ generacja adresu a;
przesłanie do Y (baza danych).

Ponieważ sekwencja (w stanie normalnym) aktywizowana jest od adresu fazy bieżącej (komórka Y), to instrukcje CHECK i WAIT będą realizowane w kolejnych cyklach, aż do spełnienia warunku $H = 0.5\ H_{MAX}$, po czym nastąpi wykonanie instrukcji FAZA 7.

START ARG1,ARG2

Instrukcja realizuje przejście sekwencji określonej przez argument ARG1 ze stanu "stop" do stanu normalnego. ARG2 określa numer fazy, od której należy sekwencję aktywizować (rys. 2.7).

START ARG1,ARG2 - generacja informacji - stan normalny; przesłanie do X (słowo stanu); generacja adresu fazy określonej przez ARG2;
przesłanie do Y (faza bieżąca).



Rys. 2.7. Graf zmiany stanów sekwencji

STOP ARG

Instrukcja realizuje przejścia sekwencji określonej przez argument ze stanów normalnego i "awaryjnego" do stanu "stop" (rys. 2.7).

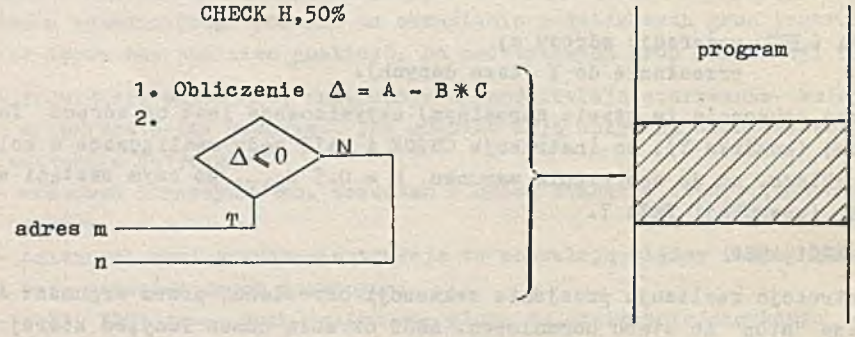
HOLD ARG1,ARG2

Instrukcja realizuje przejście sekwencji określonej przez argument ARG1, ze stanu normalnego do stanu "awaryjnego".ARG2 określa numer fazy awaryjnej, którą należy realizować (rys. 2.7).

HOLD ARG1,ARG2 ≡ generacja informacji-stan "awaryjny"; przesłanie do X; generacja adresu fazy określonej przez ARG2; przesłanie do V (faza awaryjna).

RETURN ARG

Instrukcja realizuje przejście sekwencji ze stanu "awaryjnego" do stanu normalnego. ARG określa fazę, od której należy sekwencję aktywizować (rys. 2.7).



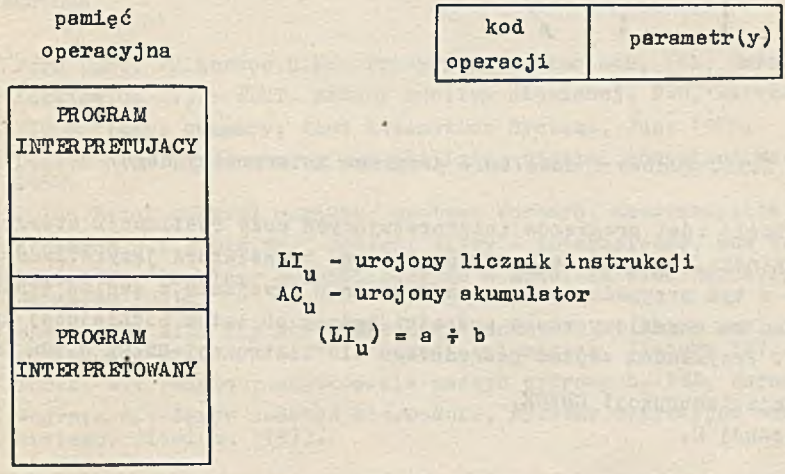
A, B, C - parametry formalne

H_{AKT}, 0.5, H_{MAX} - parametry aktualne

Rys. 2.8. Wzorzec instrukcji CHECK H,50%

Analiza instrukcji sterujących języka, jak również przykładowej instrukcji CHECK H,50% (rys. 2.8), pod kątem ich realizacji maszynowej, prowadzi do wniosku, że mamy do czynienia z makroinstrukcjami [9]. Ich wykonanie bowiem wymaga realizacji określonego ciągu rozkazów maszynowych. Translator języka będzie więc makroassemblerem. Jego budowa może być oparta na zasadzie podstawiania wzorców makroinstrukcji, z zamianą parametrów formalnych na aktualne.

Ze względu na fakt, że procesy sekwencyjne są stosunkowo "wolne", przy realizacji sterowania sekwencyjnego można wykorzystać ideę programów interpretujących [4] [6]. Programy te często stosowane są do symulacji jednych maszyn przez drugie. Podstawową cechą interpretacji jest to, że instrukcje programu interpretowanego nie są bezpośrednio wykonywane przez maszynę. Instrukcje te są interpretowane a to wymaga istnienia programu interpretującego. Podział pamięci w trakcie interpretacji podaje rys. 2.9.

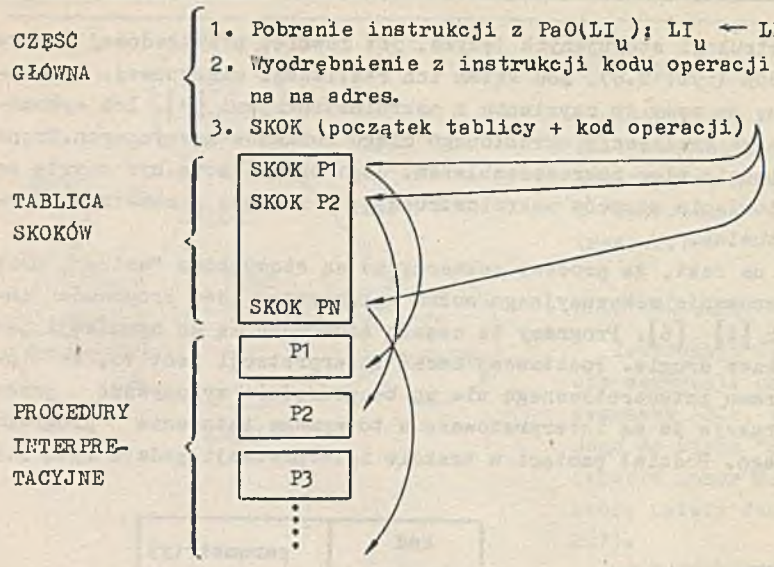


Rys. 2.9. Podział pamięci operacyjnej w trakcie interpretacji oraz postać instrukcji interpretowanej

LI$_u$ - urojony licznik instrukcji (rozkazów) określa adres instrukcji interpretowanej, w związku z czym może zawierać wyłącznie adresy programu interpretowanego (a ÷ b). Przy symulacji maszyn pojawia się dodatkowo konieczność wprowadzenia urojonych rejestrów maszyny symulowanej (np. AC$_u$).

Mimo zróżnicowania konkretnych programów interpretujących ich ideę budowy można prześledzić na rys. 2.10.

Program interpretujący, przedstawiony na rysunku posiada trzy podstawowe części. Część główną, zadaniem której jest rozpoznanie kodu operacji instrukcji, tablicę skoków oraz procedury interpretacyjne. Procedury te realizują elementarne operacje składające się na wykonanie konkretnej instrukcji.



Rys. 2.10. Budowa i działanie programu interpretującego

Wykorzystanie idei programów interpretujących przy realizacji sterowania sekwencyjnego, generalnie upraszcza budowę translatora języka. Zadanie translatora w tym przypadku polega na dokonaniu przejścia z zapisu źródłowego programu na określony zapis pośredni będący obiektem późniejszej interpretacji. Przykładem zapisu pośredniego dla instrukcji CHECK H,50% jest:

- kod operacji instrukcji CHECK,
- adres zmiennej H,
- 0.5,
- adres zmiennej H_{MAX}

4. ZAKOŃCZENIE

Grupa procesów przemysłowych o charakterze sekwencyjnym jest stosunkowo szeroka. Zaliczyć bowiem do niej można te procesy, które mają charakter typowo sekwencyjny np. wszelkie procesy rozruchu i zatrzymywania jak również i te, które tych własności z natury nie posiadają np. pewne procesy chemiczne. Wynika to z faktu ich aktualnie słabego rozpoznania. Sterowanie takimi procesami wskutek braku pełnego modelu matematycznego prowadzi się w oparciu o dane doświadczalne, na podstawie których definiuje się ciągi operacji (sekwencje czynności), jakie należy zrealizować w trakcie sterowania. Tego typu sterowanie można więc zaliczyć do sterowania sekwencyjnego.

Przykłady wskazują, że procesy sekwencyjne są często procesami o dużym znaczeniu przemysłowym.

W pierwszej części niniejszego opracowania scharakteryzowano procesy sekwencyjne oraz podjęto próbę ich ścisłej definicji. Rozważono również sieciowy model procesu sekwencyjnego.

Druga część opracowania została poświęcona oprogramowaniu centralnej jednostki sterującej procesem sekwencyjnym. Zdaniem autora - z powodów wymienionych na początku tego punktu - pożyteczne byłoby skonstruowanie specjalizowanego systemu oprogramowania maszyn cyfrowych dla sterowania grupą procesów sekwencyjnych.

Program zarządzający sterowaniem sekwencyjnym oraz język sterowania sekwencyjnego, omówione w niniejszej pracy, to niewątpliwie jedne z podstawowych elementów takiego systemu.

LITERATURA

[1] Ford L.R., Fulkerson D.R.: Przepływy w sieciach, PWN, Warszawa 1969.
[2] Idźkiewicz A.Z.: PERT. Metody analizy sieciowej. PWN, Warszawa 1967.
[3] K70 software summary; Kent Automation Systems, June 1970.
[4] Ledley R.S.: Programming and utilizing digital computers.Mc Graw-Hill 1962.
[5] PCP88 Batch control computer system; Foxboro, Massachusetts 1971.
[6] Richards R.K.: Digital design; Wiley - interscience, New York 1971.
[7] Solich R.: Problemy optymalizacyjne w modelach sieciowych,Prace COPAN Warszawa 1971.
[8] Spang A.H. III: The structure and comparison of three real-time operating systms for process control. Automatica, January 1972.
[9] Turski W.: Podstawy użytkowania maszyn cyfrowych. PWN, Warszawa 1968.
[10] Węgrzyn S.: Zarys podstaw kierowania, systemy operacyjne czasu rzeczywistego. Gliwice, 1973.

СИСТЕМЫ УПРАВЛЕНИЯ СЕКВЕНЦИОННЫХ ПРОЦЕССОВ

Р е з ю м е

В работе рассматриваются системы управления процессами секвенционного рактера.

На базе сетевои модели, представлено математическое описание секвенционного процесса.Сформулировано задачи управления. Рассмотрено заведующую программу управления и схарактеризовано структуру программ потребителя. Произведено анализ черт языка секвенционного управления и строения его транслятора.

CONTROL SYSTEMS OF SEQUENCE PROCESSES

S u m m a r y

The paper describes the control systems of sequence processes. The mathematical description of sequence process is given on the network model base and the control problems are formulated. The governing of control programs and the user programs structure are discussed. The feature analysis of the sequence control language and its translator structure is accomplished.