

2 1970



P.2229/70

prace

Instytutu Maszyn Matematycznych

rok XII

2 1970



P.2229/70

I n s t y t u t u
M a s z y n
M a t e m a t y c z n y c h

Rok XII

Warszawa 1970

Copyright © 1970 - by Instytut Maszyn Matematycznych
Poland

Wszelkie prawa zastrzeżone

Komitet Redakcyjny

Bartłomiej GŁOWACKI, Jerzy GRADOWSKI /redaktor naczelny/,
Andrzej KOJEMSKI, Włodzimierz MARDAL /z-ca redaktora naczelnego/,
Jan RELUGA

Sekretarz Redakcji: Romana NITKOWSKA

Adres Redakcji: Instytut Maszyn Matematycznych
Działowy Ośrodek INTE

Warszawa, ul. Krzywickiego 34, tel. 28-37-29

W. D. N. - Zam. 913/0/1970. Nakład 350 egz.

SPIS TREŚCI ZESZYTU 2
СОДЕРЖАНИЕ
CONTENTS

I. OPROGRAMOWANIE MASZYN CYFROWYCH

Madej W.

CEMMA - język do symulacji układów dynamicznych ... 7

CEMMA - язык для моделирования динамических схем
(Резюме) 31

CEMMA - dynamic system simulation language
/Summary/ 33

Bielkowicz P., Perkowski P.

ASTEK I - język do statystycznej analizy danych
eksperymentalnych 35

ASTEK - язык для статистического анализа экспери-
ментальных данных (Резюме) 54

ASTEK language for statistic analysis of expe-
rimental data /Summary/ 55

Witaszek J.

System magazynowania i aktualizacji dokumentów i je-
go zastosowanie w procesie uruchamiania programów 57

Система хранения и актуализации документов (Резюме) 72

The system of document storing and updating /Sum-
mary/ 73

Mysior H.

Oprogramowanie uniwersalne maszyny ZAM-41 do EPD .. 75

Универсальные средства программирования машины
ZAM-41 для обработки данных (Резюме) 87

ZAM 41 computer universal software for data pro-
cessing /Summary/ 88

II. KONSTRUKCJA I TECHNOLOGIA

Kuźnicka H.

Próba zastosowania metody Bittera do oceny jakości zapisu magnetycznego na warstwach rejestrujących pamięci bębnowych 91

Попытка применения метода Биттера к оценке качества магнитной записи на записывающих слоях памяти на магнитном барабанае (Резюме)106

A trial to apply the Bitter's method to estimate the quality of magnetic record on drum storage recording layers /Summary/107

III. AUTORZY ARTYKUŁÓW ZAMIESZCZONYCH W ZESZYCIE 2 "PRAC IMM"109

OPROGRAMOWANIE MASZYN CYFROWYCH

CEMMA - JĘZYK DO SYMULACJI
UKŁADÓW DYNAMICZNYCH

Waldemar MADEJ

Pracę złożono 28.04.1969

Praca zawiera opis języka symulacyjnego CEMMA.

Język CEMMA umożliwia modelowanie i badanie na maszynie cyfrowej ZAM 41 procesów opisywanych liniowymi i nieliniowymi równaniami różniczkowymi zwyczajnymi.

Język dzięki swej prostocie i elastyczności pozwala w łatwy sposób opisywać złożone układy dynamiczne.

Przeznaczony do specjalnego użytku przez inżynierów i naukowców, wymaga jedynie minimum wiedzy na temat programowania maszyn cyfrowych.

1. WSTĘP

Postęp w technologii maszyn cyfrowych w ostatnich latach spowodował, że symulacja cyfrowa stała się bardzo efektywną i ekonomiczną techniką badania zjawisk zachodzących w układach dynamicznych. Maszyny cyfrowe, wraz ze wzrostem szybkości, pojemności pamięci /pamięci dyskowe i taśmowe/ oraz powiększaniem się zestawu urządzeń wejścia-wyjścia /monitory ekranowe, pisaki X-Y, szybkie drukarki/, stają się bardzo praktycznym narzędziem dla inżyniera. Charakteryzują się idealną powtarzalnością wyników i łatwością ich rejestracji. Pozwalają na uzyskiwanie wysokiej dokładności obliczeń. Rozwój odpowiedniego oprogramowania /tzw. programy i języki symulacyjne/ pozwala na łatwą komunikację z maszyną, a systemy

wielodostępności i podziału czasu umożliwiają inżynierowi współpracę z maszyną bezpośrednio z jego stanowiska pracy.

W ciągu ostatnich dziesięciu lat powstało na świecie /głównie w Stanach Zjednoczonych/ ponad 30 różnych języków symulacyjnych. Można je w zasadzie podzielić na dwie grupy w zależności od postaci modelu, na podstawie którego są przygotowywane programy. W pierwszej grupie języków modele systemów ciągłych są reprezentowane funkcjonalnie przez schematy blokowe, które składają się z połączonych wzajemnie elementów funkcjonalnych. Schematy te są analogiczne do odpowiednich schematów blokowych, stosowanych przy modelowaniu na maszynach analogowych. Tę grupę języków określa się też w literaturze mianem "cyfrowych symulatorów maszyn analogowych". Można tu jako reprezentantów tej grupy wymienić języki: DEPI, DYSAC, DYNASAR, DAS, MIDAS, PACTOLUS, DES-1, 1130 CSMP.

W drugiej grupie języków odpowiednie modele są reprezentowane matematycznie, to znaczy przez zbiór równań różniczkowych i algebraicznych oraz zdań logicznych. Ta druga grupa języków stanowi w zasadzie wyższy szczebel rozwoju języków symulacyjnych, gdyż posiada możliwości modelowania blokowego powiększone o możliwości jakie daje stosowanie zdań logicznych i wyrażeń algebraicznych. Języki tej grupy zaczęły powstawać począwszy od 1965 roku. Wymienić tu można przykładowo takie języki jak: DSL/90, MIMIC, EASL, BHSL, S/360 CSMP.

Biorąc pod uwagę popularność analogowych schematów blokowych jako przejrzystą i dogodną formę opisu systemów dynamicznych zdecydowano, że pierwsza wersja języka symulacyjnego CEMMA wzorowana będzie na językach należących do pierwszej grupy. Dodatkowym założeniem było, aby język ten był możliwie prosty, a jednocześnie umożliwiający modelowanie dostatecznie szerokiej klasy problemów.

2. OPIS OGÓLNY JĘZYKA

Język CEMMA umożliwia modelowanie procesów opisywanych liniowymi i nieliniowymi równaniami różniczkowymi zwyczajnymi o współczynnikach zależnych lub niezależnych od czasu, przy czym modele takich procesów mogą być poddawane dowolnym wymuszeniom zdeterminowanym oraz zakłóceniom stochastycznym o normalnym rozkładzie prawdopodobieństwa.

Język CEMMA dzięki swej prostocie i elastyczności pozwala w łatwy sposób opisywać złożone układy dynamiczne. Przeznaczony do specjalnego użytku przez inżynierów i naukowców, wymaga jedynie minimum wiedzy na temat programowania maszyn cyfrowych. Program w języku CEMMA jest programem zorientowanym blokowo; programy dla maszyny cyfrowej są przygotowywane na podstawie odpowiednich schematów blokowych reprezentujących symulowany model. Użytkownik na podstawie schematu opisuje za pomocą zdań strukturalnych języka symulowany obiekt fizyczny i tak ułożony program wprowadza do maszyny cyfrowej.

Język ten można zaliczyć do rodziny istniejących języków, tak zwanych "cyfrowych symulatorów maszyn analogowych". Program w języku CEMMA symuluje bowiem elementy i organizację maszyny analogowej. Zestawowi elementów funkcjonalnych maszyny analogowej odpowiada w języku zestaw podprogramów numerycznych realizujących funkcje takie jak: całkowanie, sumowanie, mnożenie, jak również funkcje bardziej złożone, na przykład: ograniczenie, pierwiastek kwadratowy, funkcja sinus. Połączenia między elementami, które w maszynie analogowej uzyskuje się za pomocą przewodów elektrycznych, w maszynie cyfrowej muszą być zdefiniowane programowo. Ponadto, o ile całkowanie w maszynach analogowych jest realizowane za pomocą układu fizycznego, to w maszynie cyfrowej mamy oczywiście do czynienia z algorytmami całkowania numerycznego.

Program w języku CEMMA nie jest programem proceduralnym. Kolejność w jakiej są dokonywane obliczenia przez poszczególne podprogramy numeryczne nie zależy od porządku w jakim wys-

tępują zdania języka w programie. Kolejność obliczeń jest ustalana przez translator języka na podstawie odpowiedniego algorytmu sortowania zapewniającego poprawną sekwencję obliczeniową.

Programista piszący program w języku CEMMA ma do dyspozycji zestaw elementów blokowych /25 rodzajów operacji/ realizujących funkcje matematyczne, generatory funkcji i funkcje nieliniowe. Język CEMMA pozwala symulować stosunkowo złożone i rozbudowane układy. Na schemacie blokowym w jednym programie może występować do 200 elementów.

Programy w języku CEMMA są wprowadzane do maszyny cyfrowej na taśmie perforowanej, natomiast wyniki obliczeń są rejestrowane na arkuszu wydawniczym drukarki wierszowej w formie wykresów lub w formie tabel liczbowych.

3. ELEMENTY JĘZYKA I BUDOWA PROGRAMU W JĘZYKU CEMMA

Program w języku CEMMA składa się z trzech zasadniczych części:

- komentarza,
- programu właściwego,
- nagłówka.

Komentarz - poprzedza program właściwy. Składa się z dowolnych zdań zawierających charakterystykę rozwiązywanego problemu. Komentarz nie ma wpływu na wykonywanie programu i jest przepisywany bez zmian na arkuszu drukarki wierszowej, na którym są rejestrowane następnie wyniki obliczeń.

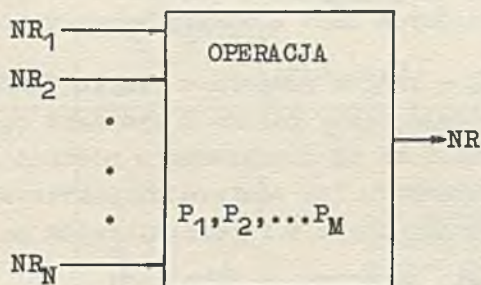
Program właściwy - składa się z zespołu zdań strukturalnych opisujących wszystkie elementy schematu. Zdania strukturalne opisują zależności funkcjonalne między zmiennymi modelu. Każdemu elementowi odpowiada jedno zdanie, które zawiera informacje o wszystkich cechach jednego elementu ze schematu blokowego.

Poszczególne zdania mają następującą ogólną postać:

NR OPERACJA NR₁ NR₂... NR_N P₁ P₂ ... P_M;

gdzie: NR - numer elementu na schemacie blokowym,
OPERACJA - rodzaj realizowanej przez element funkcji,
NR_i /i=1,2,...,N/ - numer elementu będącego argumen-
tem dla danego elementu,
P_j /j=1,2,...,M/ - parametr związany z realizowaną
funkcją.

Zdanie takie opisuje jeden element ze schematu blokowego
o ogólnej poniżej przedstawionej postaci:



Na przykład jedno zdanie w języku CEMMA może mieć postać:

2 MST 1 = 0.1;

Zdanie to opisuje element MST /mnożenie przez stałą/ o nu-
merze 2, który realizuje mnożenie wartości wielkości wyjścio-
wej elementu o numerze 1, przez liczbę 0.1.

Podstawowymi elementami, z których tworzy się zdania struk-
turalne są:

- numery bloków czyli nazwy zmiennych,
- operacje czyli nazwy elementów blokowych,
- stałe czyli wartości parametrów.

Numery bloków, czyli nazwy zmiennych, są to liczby całko-
wite z zakresu od 1 do 200. Numer jest utożsamiany z wielkoś-
cią wyjściową elementu o danym numerze.

Operacja określa rodzaj realizowanej przez dany element funkcji i stanowi trójliterową nazwę tego elementu. Wykaz elementów zrealizowanych w języku CEMMA oraz ich znaczenie, to znaczy rodzaj wypełnianych przez nie funkcji, podaje tabela 1.

Stałe, czyli wartości parametrów związanych z realizowaną funkcją, są to wielkości nie zmieniające się w trakcie obliczeń. Ogólnie, stałe reprezentują liczby rzeczywiste w zapisie zmiennoprzecinkowym o module z zakresu od 0 do 10^{76} .

Program właściwy zaczyna się od zdania początkowego, opisującego element GZN, a kończy zdaniem końcowym - opisującym element KON. Kolejność występowania w programie pozostałych zdań, opisujących inne elementy schematu, jest dowolna.

Nagłówek - kończy program w języku CEMMA. W nagłówku programista podaje nazwy kolumn z wynikami obliczeń /w przypadku, gdy wyniki te są uzyskiwane w postaci tabeli liczbowej na arkuszu drukarki/, lub zawiera charakterystyki i parametry krzywych uzyskiwanych w formie wykresów na drukarce wierszowej.

Oznaczenia stosowane w tabeli 1:

- Δt_B - wartość przyrostu zmiennej niezależnej /krok całkowania/,
- t_B - wartość wielkości wyjściowej elementu GZN - wartość zmiennej niezależnej,
- N - liczba całkowita - wskazuje numer kroku obliczeniowego / $N=0,1,2,\dots$ /,
- y - wielkość wyjściowa elementu,
- X lub x_i / $i=1,2,\dots$ / - wielkość wyjściowa elementu będącego argumentem dla danego elementu.

Tabela 1

Lp.	Nazwa elementu	Symbol graficzny	Realizowana funkcja
1	GZN - generator zmiennej niezależnej /element początkowy/	$\boxed{\text{GZN}} \text{---} t$	$t_B(N+1) = t_B(N) + \Delta t_B$

Tabela 1 /o.d./

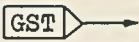


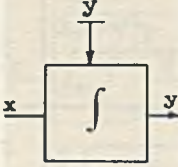
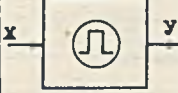

Lp.	Nazwa elementu	Symbol graficzny	Realizowana funkcja
2	GST - generator zmiennej stochastycznej		Wytwarza ciąg liczb przypadkowych o normalnym rozkładzie prawdopodobieństwa /o wartości oczekiwanej M i odchyleniu standardowym σ /
3	STA - element "źródło stałej"		$y = A; A = \text{const.}$
4	OPZ - element opóźnienia		$y(t_B) = x(t_B - \tau)$ dla $t_B \geq \tau$; $y(t_B) = 0$ dla $t_B < \tau$;
5	INT - element całkujący		$y(t_B) = \int_0^t x(t_B) - dt + y_0$ przy czym całkowanie odbywa się za pomocą metody prostokątów: $y(N+1) = y(N) + x(N) \cdot \Delta t_B$ $y(0) = y_0$
6	IMP - element impulsujący		$y = x(nT)$ dla $nT \leq t_B < nT + \tau$; gdzie T - okres impulsowania τ - długość impulsu n - liczba naturalna $y = 0$ dla $nT + \tau < t_B < (n+1)T$;
7	MST - mnożenie przez stałą		$y = k \cdot x;$

Tabela 1 /c.d./

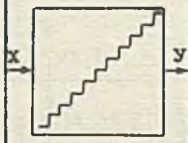
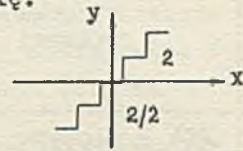
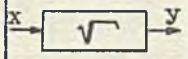
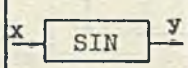
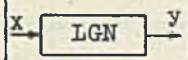
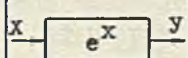

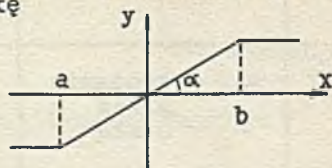
Lp.	Nazwa elementu	Symbol graficzny	Realizowana funkcja
8	KNT - element kwantujujący		Realizuje charakterystykę:  $y = q \cdot \text{ENT} \left\{ \frac{x + \frac{1}{2} q}{q} \right\};$ dla $q > 0$
9	PWK - pierwiastek kwadratowy		$y = \sqrt{x};$ dla $x \geq 0$
10	SIN - funkcja sinus		$y = \sin x$
11	LGN logarytm naturalny		$y = \ln x;$ dla $x > 0$
12	EXP - element wykładniczy		$y = e^x;$ dla $x < 176$
13	OGR - element "ograniczenie"		Realizuje charakterystykę  $y = k \cdot x \text{ dla } a \leq x \leq b$ $y = k \cdot a \text{ dla } x < a$ $y = k \cdot b \text{ dla } x > b$ gdzie: $k = \text{tg} \alpha; a < b;$ $0^\circ < \alpha < 90^\circ;$

Tabela 1 /c.d./


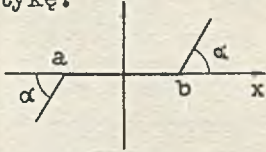

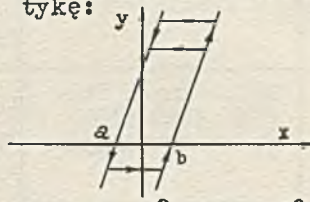
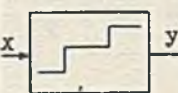
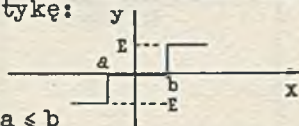
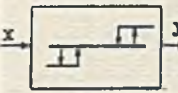
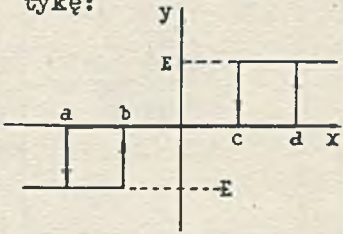
Lp.	Nazwa elementu	Symbol graficzny	Realizowana funkcja
14	STN - element "strefa nieczułości"		<p>Realizuje charakterystykę:</p>  <p> $y=0$ dla $a \leq x \leq b$ $y=k(x-a)$ dla $x < a$ $y=k(x-b)$ dla $x > b$ gdzie: $k = \operatorname{tg} \alpha$; $a \leq b$; $0^\circ < \alpha < 90^\circ$ </p>
15	LUZ - element "luzu maszynowego"		<p>Realizuje charakterystykę:</p>  <p> $\operatorname{tg} \alpha = k$; $0^\circ < \alpha < 90^\circ$ $a < b$ </p>
16	PRZ - element o charakterystyce przekąźnikowej		<p>Realizuje charakterystykę:</p>  <p> $a \leq b$ </p>
17	PHZ - element o charakterystyce przekąźnikowej z histerezą		<p>Realizuje charakterystykę:</p>  <p> $a \leq b$; $c \leq d$ $a \leq c$; $b \leq d$ </p>

Tabela 1 /c.d./

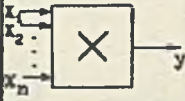
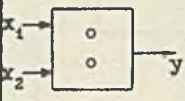
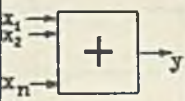
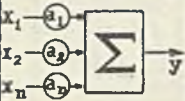


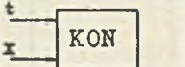
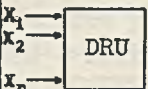
Lp.	Nazwa elementu	Symbol graficzny	Realizowana funkcja
18	MNZ - mnożenie zmiennych		$y = \prod_{i=1}^n x_i;$ $n = 1, 2, 3, \dots$
19	DZL - dzielenie zmiennych		$y = \frac{x_1}{x_2}; \quad x_2 \neq 0$
20	SUM - sumowanie zmiennych		$y = \sum_{i=1}^n x_i;$ $n = 1, 2, 3, \dots$
21	DOD - suma "ważona" zmiennych		$y = \sum_{i=1}^n a_i x_i;$ $n = 1, 2, 3, \dots$
22	FUN - funkcja jednej zmiennej zadana tablicą		Wielkość wyjściowa y i wielkość wejściowa x związane są zależnością zadaną "tablicą funkcji jednej zmiennej". Wartość y wyznaczana jest za pomocą metody interpolacji liniowej
23	FUD - funkcja dwóch zmiennych zadana tablicą		Wielkość wyjściowa y i wielkości wejściowe x1 i x2 związane są zależnością zadaną "tablicą funkcji dwóch zmiennych". Wartość y wyznaczana jest metodą interpolacji liniowej
24	KON - element końcowy		Określa warunki zakończenia obliczeń: $t \geq T_K; \quad x \geq X_K$

Tabela 1 /c.d./

Lp.	Nazwa elementu	Symbol graficzny	Realizowana funkcja
25	DRU - element drukowania wyników obliczeń		<p>Określa wielkości, których wartości są rejestrowane na arkuszu drukarki wierszowej oraz określa częstotliwość drukowania ($1 \leq n \leq 8$)</p> <p>DRU1 - wyniki w postaci numerycznej,</p> <p>DRU2 - wyniki w postaci graficznej</p>

4. PROGRAMOWANIE I METODYKA ROZWIĄZYWANIA ZADAŃ W JĘZYKU CEMMA

W celu poprawnego rozwiązania zadania na maszynie cyfrowej za pomocą języka symulacyjnego CEMMA należy metodycznie opracować, w odpowiedniej kolejności, szereg zagadnień. Obejmują one kolejno:

- sformułowanie zadania,
- sporządzenie schematu blokowego modelu,
- ułożenie programu,
- wprowadzenie programu do maszyny,
- sprawdzenie otrzymanych wyników i interpretację całości rozwiązania.

Sformułowanie zadania. Każde zadanie przeznaczone do rozwiązania za pomocą języka symulacyjnego CEMMA może być przedstawione w postaci równania lub układu równań albo w postaci schematu strukturalnego; zarówno w pierwszym jak i drugim sformułowaniu powinny być określone wszystkie parametry zada-

nia, warunki początkowe i brzegowe, funkcje wymuszające oraz przedział zmiennej niezależnej. W zagadnieniach fizycznych podaje się również miana wszystkich wielkości.

Schemat blokowy. Zaletą języka CEMMA jest bezpośrednio programowanie maszyny cyfrowej wprost ze schematu blokowego, który jest graficznym modelem równań lub schematu strukturalnego, podanych w sformułowaniu zadania. Schemat taki składa się z połączonych wzajemnie podstawowych elementów funkcjonalnych /zrealizowanych w języku/. Przesyłanie sygnałów między elementami schematu zaznacza się odpowiednimi liniami ciągłymi, a kierunek przesyłania informacji - strzałkami. Następnie wszystkim elementom operacyjnym nadaje się nazwy przyjęte w języku oraz dowolne numery z zakresu od 1 do N, gdzie N - liczba wszystkich elementów na schemacie blokowym /przy czym $N \leq 200$ /. W ten sposób każdemu elementowi schematu przyporządkowany jest odpowiedni numer porządkowy. Ponadto, przy każdym elemencie podaje się parametry opisujące właściwości elementu, na przykład współczynniki i warunki początkowe.

W języku CEMMA praktycznie nie występują ograniczenia zmiennych, a zatem są zbędne równania zamiany skal między rzeczywistym układem współrzędnych zadania a współrzędnymi modelu. Można więc wprost na schemacie blokowym uwzględnić wszystkie dane, jakie są zawarte w zadaniu. Trzeba jednak pamiętać, że wszystkie wielkości zmiennych zależnych są odniesione do miana zmiennej niezależnej. Jeżeli na przykład w zadaniu zmienna niezależna jest czasem podanym w sekundach a prędkość w m/s, to droga będzie w m, a przyspieszenie w m/s^2 .

Programowanie zadania w języku CEMMA. Tak przygotowany problem może być zaprogramowany w języku symulacyjnym, przy czym programowanie polega tu na odpowiednim opisanu schematu blokowego za pomocą zdań strukturalnych /zgodnie z regułami podanymi w punkcie 3/.

Wprowadzanie programu do maszyny. Po ułożeniu programu, koduje się go odpowiednio na dziurkowanej taśmie papierowej

i wprowadza do maszyny cyfrowej. Translator języka CEMMA, po przetłumaczeniu programu i utworzeniu programu wynikowego w pamięci operacyjnej maszyny, ustala kolejność w jakiej będą odbywać się obliczenia oraz inicjuje proces liczenia. Proces liczenia, czyli rozwiązywanie danego problemu przez maszynę cyfrową, odbywa się iteracyjnie w kolejnych krokach obliczeniowych, przy czym poszczególne podprogramy numeryczne dokonują obliczeń sekwencyjnie - według kolejności ustalonej przez translator. Jako ostatni w tej sekwencji inicjowany jest podprogram realizujący funkcję reprezentowaną przez element końcowy KON. Zakończenie obliczeń przez ten podprogram stanowi zakończenie danego kroku obliczeniowego. Następnie licznik kroków jest zwiększany o 1, wartość zmiennej niezależnej jest zwiększana o wartość kroku całkowania i sekwencja obliczeń jest powtarzana. Obliczenia są prowadzone do momentu, gdy spełnione zostają warunki zdefiniowane przez programistę w zdaniu końcowym, opisującym element KON.

Krok całkowania /elementarny przyrost zmiennej niezależnej/ określany jest w zdaniu początkowym programu opisującym element GZN. W języku CEMMA krok całkowania nie jest wybierany automatycznie, a jego wartość wybiera programista doświadczalnie dla każdego problemu oddzielnie. Od wartości kroku całkowania zależy bowiem dokładność wyników oraz czas liczenia przez maszynę. Oczywiście im mniejsza jest ta wartość, tym większa jest dokładność i lepsza stabilność rozwiązania, ale i dłuższy czas liczenia. Ogólnie, wartość kroku całkowania powinna być przyjmowana w wyniku kompromisu między wymaganiami dotyczącymi dużej dokładności a krótkim czasem liczenia. W pierwszej wersji języka CEMMA całkowanie odbywa się za pomocą metody prostokątów. Zaletami tej metody są prostota algorytmu obliczania całki oraz łatwość startu rozwiązania. Wadą natomiast, stosunkowo długi czas rozwiązywania problemu przy wymaganej dużej dokładności obliczeń.

Czas symulacji danego problemu, przy użyciu języka CEMMA, na maszynie ZAM 41 można określić w przybliżeniu z następującej zależności:

$$t \cong (0.005 - 0.01) \frac{T_k}{\Delta t} \cdot N \text{ [sek]}$$

gdzie: N - liczba elementów na schemacie blokowym,
 T_k - wartość końcowa zmiennej niezależnej /czasu/,
 Δt - wartość kroku całkowania.

Na przykład w przypadku symulacji kotła energetycznego OP-650K, jako obiektu regulacji, czas rozwiązywania wynosił około 20 minut.

Dynamika kotła była opisana za pomocą około 60 równań różniczkowych. Liczba elementów na schemacie blokowym wynosiła 126. Odpowiedzi były wyznaczone w przedziale zmiennej niezależnej od 0 do 3600 przy kroku całkowania 0.5 i rejestracji wyników co 20.0.

Wyniki obliczeń. W trakcie rozwiązywania problemu przez maszynę cyfrową następuje jednoczesna rejestracja wyników na arkuszu wydawniczym drukarki wierszowej, w zadanych przez programistę odstępach zmiennej niezależnej. Wyniki drukowane są w formie tabeli liczbowej lub w formie wykresów. Liczbę rejestrowanych przebiegów oraz częstotliwość drukowania wyników określa programista w zdaniu DRU. Forma uzyskiwanych wyników zależy od tego, czy zdanie opisujące element DRU zawiera operację DRU1 czy DRU2. Zdanie DRU1 zapewni wyniki w postaci tabeli liczbowej. Liczba jednocześnie drukowanych kolumn zawiera się między 1 i 8. Podobnie, w przypadku występowania w programie zdania DRU2 na arkuszu wydawniczym może być wykreślone jednocześnie do 8 krzywych. W programie w języku CEMMA może występować tylko jedno zdanie DRU, nie można więc jednocześnie uzyskiwać wyników w formie tabeli liczbowej i w postaci wykresu.

Po zakończeniu obliczeń należy sprawdzić otrzymane wyniki, przynajmniej dla kilku współrzędnych rozwiązania. Zazwyczaj, w prawie każdym zadaniu udaje się znaleźć szczególną wartość rozwiązania, na przykład w stanie ustalonym, w stosunku do której można określić rozwiązanie uzyskane z maszyny cyfrowej. W przypadku niezadawalającej dokładności, obliczenia należy powtórzyć z mniejszą wartością kroku całkowania.

5. PRZYKŁAD PROGRAMOWANIA W JĘZYKU CEMMA

Badanie układu inercyjnego pierwszego rzędu.

A/ Sformułowanie zadania

Dany jest układ inercyjny pierwszego rzędu opisywany równaniem różniczkowym:

$$y(t) + T \frac{d}{dt} y(t) = k_0 \cdot x(t);$$

$$\text{przy czym } y(0) = y_0;$$

gdzie: $x(t)$ - wymuszenie, zmienna wejściowa,
 $y(t)$ - odpowiedź, zmienna wyjściowa,

należy zbadać zachowanie układu przy warunku początkowym $y_0 = -2.5$, dla wymuszenia skokowego:

$$x(t) = 10.1(t)$$

przy następujących parametrach układu:

- wartość stałej czasowej $T = 10$ sek,
- wartość wzmożenia $K_0 = 1$.

Wyniki mają być wyprowadzane z maszyny cyfrowej w postaci numerycznej i graficznej w odstępach zmiennej niezależnej $\Delta t = 0.5$ sek. Krok całkowania $\Delta t_B = 0.1$ sek, a wartość końcowa zmiennej niezależnej $T_K = 50$ sek.

Należy zarejestrować w funkcji zmiennej niezależnej przebiegi: odpowiedź $y(t)$, oraz jej pierwszą pochodną $\frac{d}{dt} y(t)$.

B/ Schemat blokowy

W celu narysowania schematu blokowego, ponieważ jest dane równanie różniczkowe układu, można skorzystać z metody modelowania równań. W tym celu przekształcamy dane równanie do postaci następującej:

$$\frac{d}{dt} y(t) = \frac{1}{T} [k_0 \cdot x(t) - y(t)]$$

Rozpatrywany układ możemy zmodelować w języku CEMMA za pomocą na przykład trzech elementów:

- DOD - element sumowania zmiennych,
- MST - element mnożenia przez wartość stałą,
- INT - element całkujący.

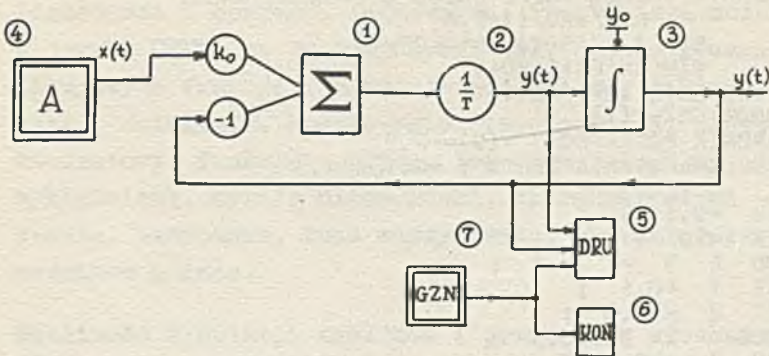
W elemencie całkującym INT zadajemy także warunek początkowy $y(0) = y_0$, co zaznaczamy odpowiednio na schemacie przedstawionym na rys. 1. Następnie, chcąc zbadać odpowiedź czasową takiego układu na wymuszenie skokowe, czyli $x(t) = A \cdot 1(t)$, należy uzupełnić schemat blokowy elementem STA /element "źródło stałej"/ o wartości parametru równej A, a także elementami pomocniczymi: GZN, DRU i KON, które muszą występować w każdym programie w języku CEMMA.

Przy czym:

- GZN - stanowi generator zmiennej niezależnej /realizuje funkcję rosnącą liniowo/,
- DRU - element drukowania; określa elementy, których wartości wielkości wyjściowych będą rejestrowane,
- KON - element końcowy; definiuje warunki zakończenia obliczeń.

Sformułowanemu w ten sposób problemowi, w języku symulacyjnym CEMMA, odpowiada schemat blokowy jak na rys. 1 /przy zastosowaniu symboli graficznych proponowanych w tabeli 1/.

Po sporządzeniu schematu, każdemu elementowi przypisujemy numer porządkowy. Na rys. 1 numery nadane elementom modelu zostały wpisane w kółeczkach. Będą one oznaczać w programie odpowiednie zdania zapisane w języku CEMMA.



Rys. 1

C/ Programowanie zadania

Napisanie programu na podstawie sformułowania zadania i schematu blokowego nie przedstawia trudności. Przepisując następnie program, na przykład na dalekopisie, otrzymuje się taśmę perforowaną i tabulogram programu /czyli arkusz z zapisanym programem/. Tabulogramy programu odpowiadające sformułowanemu powyżej zadaniu zostały przedstawione niżej.

```

: PRZYKŁAD PROGRAMU W JEZYKU CEMMA
: BADANIE UKŁADU INERCYJNEGO O TRANSMITANCJI
:      G/S/=K0/(1+T.S)
:      STALA CZASOWA T=10[SEK.]
:      WZMOCNIENIE K0=1
: WYMUSZENIE X(T)=10.1(T)
: ODPOWIEDZ-Y(T)
: WARUNEK POCZĄTKOWY Y(0)=-2.5
: KROK CAŁKOWANIA DELTA T =0.1[SEK.]

```

```

7 GZN =0.1 ;
4 STA =10 ;
1 DOD 4 3 =1 =-1 ;
2 MST 1 =0.1 ;
3 INT 2 =-2.5 ;
5 DRU1 7 2 3 =0.5 ;
6 KON 7 =50 =50 ;

```

```

/*
/ CZAS T /
/ Y(T) PRIM /
/ Y(T) /
*/

```

Tabulogram 1. Tabulogram programu - wyniki uzyskiwane w postaci tabeli liczbowej

```
: PRZYKŁAD PROGRAMU W JEZYKU CEMMA
: BADANIE UKŁADU INERCYJNEGO O TRANSMITANCJI
:           G/S/=K0/(1+T.S)
:           STALA CZASOWA T=10[SEK.]
:           WZMOCNIENIE K0=1
: WYMUSZENIE X(T)=10.1(T)
: ODPOWIEDZ-Y(T)
: WARUNEK POCZĄTKOWY Y(0)=-2.5
: KROK CAŁKOWANIA DELTA T =0.1[SEK.]
```

```
7 GZN =0.1 ;
4 STA =10 ;
1 DOD 4 3 =1 =-1 ;
2 MST 1 =0.1 ;
3 INT 2 =-2.5 ;
5 DRU2 2 3 =0.5 ;
6 KON 7 =50 =50 ;
```

```
*/Y(T) PRIM / + =0 =4 /
*/Y(T) / Y =-5 =15 /
*
```

Tabulogram 2. Tabulogram programu - wyniki w postaci wykresów

D/ Wyniki obliczeń

Wyniki obliczeń dotyczące powyżej sformułowanego zadania, w postaci oryginalnego arkusza wydawniczego drukarki wierszowej, zostały zamieszczone w DODATKU.

6. MODYFIKACJE JEZYKA

6.1. Zalety i wady modelowania w języku CEMMA

Zalety i wady modelowania w języku symulacyjnym CEMMA na maszynie ZAM 41, w porównaniu z modelowaniem na maszynie analogowej, można ująć w kilku następujących punktach.

Do zalet należą:

1. Możliwość symulacji dużych układów dynamicznych. Maksymalna liczba elementów na schemacie blokowym może być równa 200, co odpowiada maszynie analogowej o około 150 miejscach operacyjnych.

2. Łatwość realizacji przebiegów impulsowych, złożonych, nieliniowości i operacji logicznych. Programista modelujący w języku CEMMA ma do dyspozycji cały zestaw elementów realizujących funkcje i operacje nieliniowe, takie jak: mnożenie, dzielenie, kwantowanie, impulsowanie, pierwiastek kwadratowy, funkcje - sinus, logarytm naturalny, funkcję wykładniczą, strefę nieczułości, charakterystyki - ograniczenia, nasycenia, luzu maszynowego, charakterystyki przekładników i inne.
3. Możliwość symulacji zakłóceń i przebiegów stochastycznych o normalnym rozkładzie prawdopodobieństwa.
4. Możliwość dokładnej realizacji funkcji opóźnienia w czasie /dokładniej - o zadaną wartość zmiennej niezależnej/.
5. Idealna powtarzalność wyników. Przy symulacji na maszynie cyfrowej nie istnieje problem "dryftu" i nie odgrywają roli zakłócenia i szumy.
6. Przy modelowaniu w języku CEMMA nie trzeba dokonywać skalowania czasu i amplitudy. Wszystkie stałe i wielkości zmienne mają w maszynie cyfrowej reprezentację zmiennoprzecinkową.
7. Dokładność wyników nie zależy od dokładności elementów maszyny cyfrowej, a jedynie od przyjętego kroku całkowania. Zmniejszając krok całkowania można uzyskać w zasadzie dowolnie dużą dokładność wyników /nieosiągalną przy modelowaniu na maszynie analogowej/.
8. Możliwość łatwego dokonywania zmian w programie /zmiana wartości parametru lub wymiana zdania w programie/.
9. Język CEMMA zapewnia dogodny i łatwy sposób rejestracji wyników na arkuszu drukarki wierszowej /w formie tabeli liczbowej lub w formie wykresów/. Jednocześnie może być drukowane do ośmiu kolumn liczbowych lub ośmiu wykresów.

Natomiast do wad symulacji cyfrowej /przy użyciu języka CEMMA/ zaliczamy:

1. Brak możliwości zmiany nastaw parametrów w trakcie obliczeń, czego w maszynie analogowej dokonuje się poprzez regulację ręczną.
2. W przeciwieństwie do symulacji na maszynie analogowej, czas obliczeń zależy od liczby elementów na schemacie blokowym, gdyż poszczególne podprogramy numeryczne dokonują obliczeń szeregowo według kolejności ustalonej przez translator języka, a więc odmiennie niż w maszynie analogowej, gdzie poszczególne elementy prowadzą obliczenia równoległe.
3. Całkowanie odbywa się w obecnej wersji języka za pomocą metody prostokątów. Tak więc, gdy wymagana jest duża dokładność wyników, znacznie wydłuża się czas liczenia.
4. Wybór wartości kroku całkowania stanowi kompromis między dokładnością a czasem obliczeń. W języku CEMMA krok całkowania nie jest wybierany automatycznie, a jego wartość wybiera programista doświadczalnie.
5. Nie ma możliwości wprowadzania danych z zewnątrz w czasie rzeczywistym, co zapewniają niektóre maszyny analogowe.

6.2. Język CEMMA 2 ^{*)}

Obecnie prowadzone są dalsze prace nad językiem CEMMA. Następna, zmodyfikowana i udoskonalona wersja języka pod nazwą CEMMA 2 została rozszerzona o następujące elementy i funkcje:

1. Zwiększenie maksymalnej liczby elementów na schemacie blokowym do 500.
2. Możliwość powtarzania obliczeń ze zmianą wartości określonych parametrów.

^{*)} Ukazanie się opisu języka CEMMA 2 przewidziane jest w IV kw. 70 r.

3. Możliwość automatycznego doboru parametrów przez maszynę na podstawie określonego kryterium, czyli optymalizację modelu ze względu na zadane kryterium.
4. Możliwość definiowania i stosowania w programie makrobloków.
5. Możliwość występowania na schemacie zdefiniowanych pętli algebraicznych, czyli możliwość rozwiązywania funkcji uwikłanych.
6. Zrealizowany został bogatszy zestaw elementów blokowych, na przykład elementy realizujące funkcje logiczne i elementy pamiętające wartości ekstremalne.
7. Zrealizowanych zostało siedem różnych algorytmów całkowania /sześć ze stałym i - jeden ze zmiennym krokiem całkowania/.
8. Możliwość jednoczesnego rejestrowania do 50 przebiegów w postaci numerycznej i do 50 wykresów graficznych.
9. Możliwość wprowadzania danych do maszyny cyfrowej z kart perforowanych i z taśmy dziurkowanej.

Realizacja translatora języka CEMMA 2 przyniesie niewątpliwie istotne ułatwienia użytkownikom w rozwiązywaniu zarówno problemów analizy jak i syntezy układów dynamicznych.

Literatura

- [1] SELFRIDGE R.G.: Coding a General-Purpose Digital Computer to Operate as a Differential Analyzer, Proc. 1955 Western Joint Computer Conference /IRE/.
- [2] STEIN M.L., ROSE J.: Changing from Analog to Digital Programming by Digital Techniques, ACM Journal, January 1960: 7, 1, 10-23.
- [3] HURLEY J.R.: DYSAC, A Digitally Simulated Analog Computer, AIEE Summer General Meeting, Denver, Colorado, June 17-22, 1962.
- [4] GASKILL R.A., HARRIS J.W., Mc KNIGHT A.L.: DAS - A Digital Analog Simulator, Proc. 1963 Spring Joint Computer Conference.

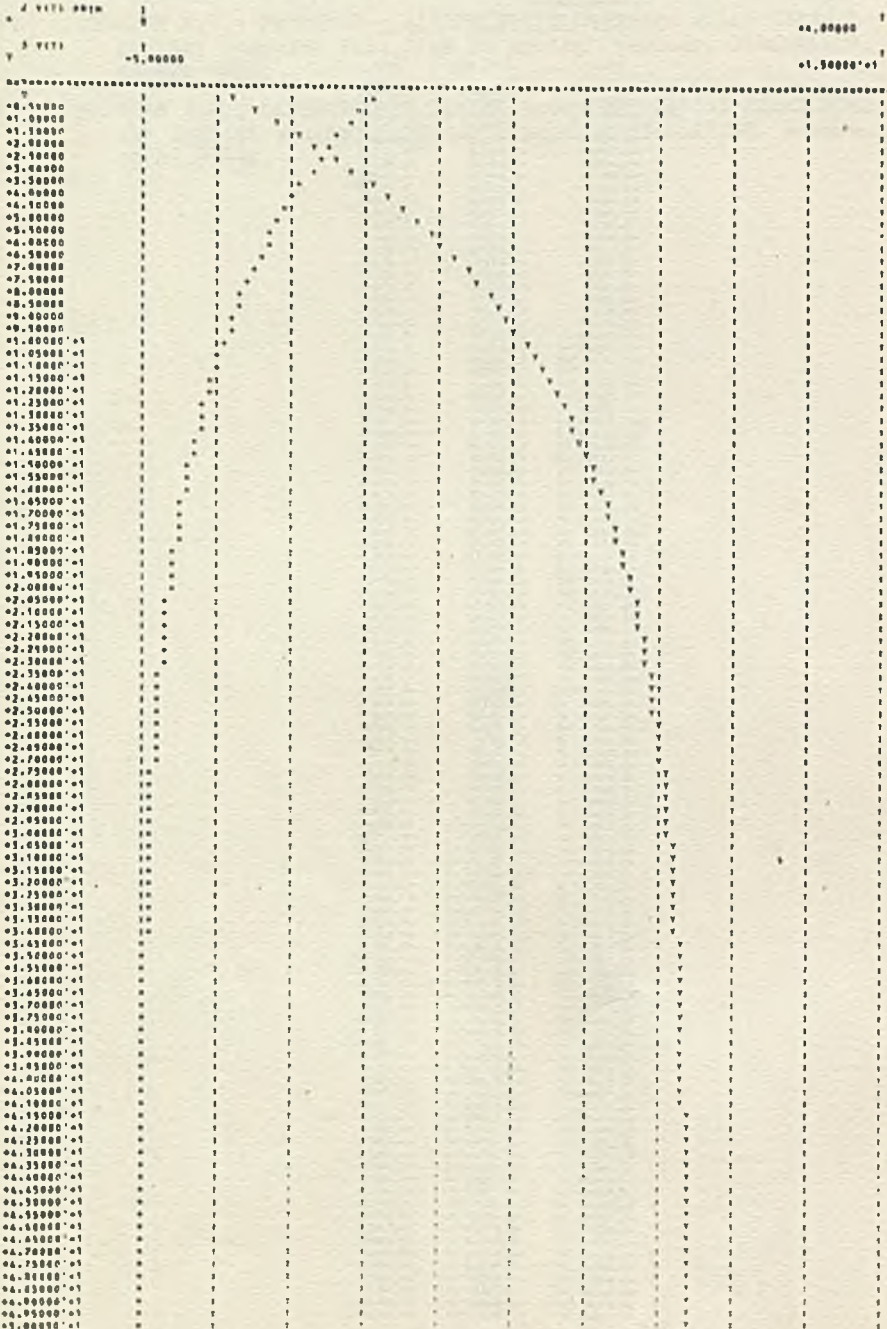
- [5] PALEVSKY M., HOWELL J.V.: DES-1, Fall 1963 Joint Computer Conference, Vol. 24, Spartan Books, Inc., Washington, D.C., 1963.
- [6] BRENNAN R.D.: PACTOLUS - A Simulation Language which Makes a Digital Computer Feet Like an Analog Computer, SIMULATION, August 1964.
- [7] BRENNAN R.D., LINEBARGER R.N.: A Survey of Digital Simulation and Digital Analog Simulator Programs, SIMULATION, December 1964: 22-36.

DODATEK

! PRZEWID PROGRAMU W 302400 CENNA
! BADANIE WŁASNOŚCI IBERCYSKIEGO O TRANSMITANCJI
! 4/17/80B(1/10, 1)
! STALA CZAJDWA T=0.1000E+1
! WZROSTNIEC W=1
! WYKUSZENIE B(T)=10.1(1)
! UNDOU1002=1(1)
! WZROSTNIEC DOZWIADU W(0)=2.1
! KROK CALKOWANIA DELTA 1 00.1(1000.)

F CZAS T	Z W(1) BAIN	I Y(1)
00.50000	+1.25000	-2.50000
01.00000	+1.18076	-1.80238
01.50000	+1.15848	-1.58478
02.00000	+1.07507	-0.75975
02.50000	+1.02238	-0.22344
03.00000	+0.97228	+0.22723
03.50000	+0.92443	+0.75375
04.00000	+0.87815	+1.26600
04.50000	+0.83421	+1.83985
05.00000	+0.79323	+2.06748
05.50000	+0.75426	+2.63762
06.00000	+0.71810	+3.20007
06.50000	+0.68383	+3.14054
07.00000	+0.65063	+3.48376
07.50000	+0.61855	+3.81652
08.00000	+0.58823	+4.11747
08.50000	+0.55960	+4.48594
09.00000	+0.53199	+4.68012
09.50000	+0.50591	+4.90005
10.00000	+0.48112	+5.10000
10.50000	+0.45756	+5.28000
11.00000	+0.43512	+5.46000
11.50000	+0.41379	+5.80200
12.00000	+0.39351	+6.06000
12.50000	+0.37421	+6.29701
13.00000	+0.35568	+6.44115
13.50000	+0.33844	+6.41557
14.00000	+0.32168	+6.78146
14.50000	+0.30608	+6.83918
15.00000	+0.29108	+7.02910
15.50000	+0.27681	+7.23125
16.00000	+0.26325	+7.56752
16.50000	+0.25035	+7.49054
17.00000	+0.23800	+7.65423
17.50000	+0.22641	+7.73501
18.00000	+0.21531	+7.80000
18.50000	+0.20476	+7.95000
19.00000	+0.19472	+8.05275
19.50000	+0.18518	+8.14810
20.00000	+0.17611	+8.23894
20.50000	+0.16747	+8.32575
21.00000	+0.15927	+8.40733
21.50000	+0.15144	+8.48359
22.00000	+0.14404	+8.55502
22.50000	+0.13698	+8.63071
23.00000	+0.13027	+8.69715
23.50000	+0.12388	+8.76110
24.00000	+0.11781	+8.82180
24.50000	+0.11204	+8.87944
25.00000	+0.10654	+8.93455
25.50000	+0.10132	+8.98677
26.00000	+0.09636	+9.03643
26.50000	+0.09163	+9.08165
27.00000	+0.08714	+9.12506
27.50000	+0.08287	+9.17127
28.00000	+0.07881	+9.21180
28.50000	+0.07495	+9.25051
29.00000	+0.07128	+9.28724
29.50000	+0.06778	+9.32210
30.00000	+0.06444	+9.35510
30.50000	+0.61301	+9.38680
31.00000	+0.58207	+9.41765
31.50000	+0.55440	+9.44580
32.00000	+0.52723	+9.47227
32.50000	+0.50130	+9.49801
33.00000	+0.47681	+9.52110
33.50000	+0.45344	+9.54085
34.00000	+0.43122	+9.56078
34.50000	+0.41000	+9.58001
35.00000	+0.38899	+9.60001
35.50000	+0.37000	+9.62012
36.00000	+0.35270	+9.64110
36.50000	+0.33541	+9.66459
37.00000	+0.31897	+9.68103
37.50000	+0.30334	+9.69900
38.00000	+0.28847	+9.71553
38.50000	+0.27436	+9.72946
39.00000	+0.26089	+9.73911
39.50000	+0.24811	+9.75180
40.00000	+0.23595	+9.76405
40.50000	+0.22438	+9.77502
41.00000	+0.21338	+9.78462
41.50000	+0.20291	+9.79207
42.00000	+0.19290	+9.80702
42.50000	+0.18332	+9.81648
43.00000	+0.17433	+9.82547
43.50000	+0.16589	+9.83402
44.00000	+0.15784	+9.84210
44.50000	+0.15011	+9.84980
45.00000	+0.14273	+9.85725
45.50000	+0.13575	+9.86475
46.00000	+0.12910	+9.87000
46.50000	+0.12277	+9.87723
47.00000	+0.11670	+9.88126
47.50000	+0.11093	+9.88497
48.00000	+0.10539	+9.88841
48.50000	+0.10002	+9.89059
49.00000	+0.09481	+9.89161
49.50000	+0.08976	+9.89164
50.00000	+0.08487	+9.89167

```
I PRZYKŁAD PROGRAMU W JĘZYKU C#  
I BAZA DANYCH (BANKOWY) O TITANIE  
I  
I 0/1/10/1/1/1/1  
I STALA CZASOWA T=10[sec.],  
I WZROSTOWA H=1  
I WZROSTOWA U=10, V=1  
I ODPOWIEDZ-V(T)  
I WZROSTOWA POCZATKOWA Y(0)=0,5  
I RODZAJ ROZWIĄZANIA 0/1/1/1/1/1/1/1
```



СЕММА - ЯЗЫК ДЛЯ МОДЕЛИРОВАНИЯ ДИНАМИЧЕСКИХ СХЕМ

Резюме

Работа включает описание моделирующего языка СЕММА. Язык СЕММА (польское сокращение Цифровое Моделирование Аналоговой Машины) является специализированным языком с применением блок-схем и предназначенным для облегчения цифрового моделирования непрерывных процессов на цмв ЗАМ-41. Это универсальный язык и применяя его можно решать проблемы связанные с моделированием любых динамических схем. Этот язык позволяет моделировать процессы описываемые линейными и нелинейными обыкновенными дифференциальными уравнениями, при чём модели таких процессов могут подвергаться любым стохастическим вынуждениям с нормальным распределением вероятности.

Моделирующий язык СЕММА предназначен для специального употребления инженерами и учеными и требует лишь минимум знания по программировании цифровых машин. Программа на языке СЕММА является программой базированной на блок-схеме; программы для цифровой машины готовятся на основе соответствующих блок-схем, представляющих моделированную модель. Потребитель, на основе схемы, описывает при помощи структурных предложений языка, моделируемую физическую систему и так созданную программу вводит в цифровую машину. Структурные предложения являются основными элементами языка СЕММА. Они описывают функциональные зависимости между переменными модели. Одно предложение заключает информацию о всех характерных чертах одного элемента блок-схемы.

Этот язык можно зачислить в семью существующих языков, так называемых "цифровых моделирующих устройств аналоговых машин". Программа на языке СЕММА моделирует элементы и организацию аналоговой машины. Набор операционных элементов аналоговой машины отвечает на языке набор числовых подпрограмм выполняющих такие функции как: интегрирование, суммирование, умножение, а также более сложные функции, например ограничение, квадратный корень, функция синус. Соединения между элементами, которые в цифровой машине получаются при помощи электрических проводов, в цифровой машине должны быть определены в программе. Кроме того, поскольку интегрирование в аналоговых машинах выполняется при помощи физического устройства, то в цифровой машине мы очевидно имеем дело с алгоритмами числового интегрирования.

Программа на языке СЕММА не является процедуральной программой. Очередность выполнения вычислений отдельными числовыми подпрограммами, отвечающими соответственным элементам на блок-схеме, не зависит от порядка, в котором предложения языка выступают в программе. Очередность вычислений определяет транслятор языка на основе соответствующего алгоритма сортировки, обеспечивающего правильную последовательность вычисления.

Программист, пишущий программу на языке СЕММА, имеет в распоряжении набор блоковых элементов (25 родов операции) выполняющих математические функции, генераторы функции и нелинейные функции. Язык СЕММА позволяет моделировать относительно сложные и развёрнутые динамические схемы. В одной программе может выступать до 200 элементов.

Программы на языке СЕММА вводятся в цифровую машину на перфоленте, но результаты вычислений записываются на издательском листу построчного печатного устройства в виде графиков или числовых таблиц.

CEMMA - DYNAMIC SYSTEM SIMULATION LANGUAGE

Summary

The paper presents a description of CEMMA simulating language. CEMMA /Polish abbreviation - Digital Simulation of Analog Computer/ is a specialized block-oriented language destined to facilitate digital simulation of continuous processes on ZAM-41 computers. It is a universal language and its application helps to solve problems connected with modeling and simulation of any dynamic system. This language enables processes described by linear and nonlinear ordinary differential equations, and these process models may be submitted to any stochastic input signal with normal distribution of probability.

The simulating language CEMMA is destined for a special use of engineers and scientists, it requires a minimal knowledge of digital computer programming. The CEMMA program is block-oriented; programs for digital computers are prepared on the basis of appropriate block-diagrams, representing a simulated model. On the basis of the scheme the user describes, by means of structural statements of the simulated language, the physical system and introduces the so formed program into a digital computer. Structural statements are the basic units of CEMMA language. They describe functional dependences of the model variables. One statement includes information about all features of one element of the block-diagram.

This language can be accepted to the family of existing languages, so-called "analog computer simulators". The CEMMA program simulates elements and the organization of an analog computer. In CEMMA language the set of numerical subroutines, that realizes such functions as: integrating, summing, multiplying, and more complex ones, e.g. limited, square root, sinus, corresponds to the set of operational elements of an analog computer. Connections between elements obtained in an analog computer, by means of electric wires, in a digital computer must be program-defined. Moreover, integration in analog computers is realized by means of a physical system, but in a digital computer it must be approximated by summation technics.

The CEMMA program is not procedural. The order of execution of computations by separate numerical subroutines, corresponding to proper block-diagram elements, does not depend on the order of language statement presentation in the program. The sequence of computations is fixed by the language translator on the basis of an adequate sorting algorithm that ensures the right computation.

The programmer who writes the program in CEMMA language has to his disposal a set of block-elements /25 kinds of operations /realizing mathematical generator and nonlinear functions. The CEMMA language permits to simulate relatively complex and developed systems. 200 elements can appear in a block-diagram in one program.

CEMMA programs are introduced into a digital computer on a punch-tape, but computation results are recorded on an editorial sheet of a line-printer as diagrams or numerical tables.

ASTEK I - JĘZYK DO STATYSTYCZNEJ
ANALIZY DANYCH EKSPERYMENTALNYCH

Piotr BIELKOWICZ
Piotr PERKOWSKI

Pracę złożono 5.07.1969

ASTEK jest zorganizowanym pakietem programów do wstępnej obróbki danych eksperymentalnych, sporządzenia i weryfikacji modelu właściwości statycznych i dynamicznych procesu przemysłowego w oparciu o metody analizy regresyjnej, korelacyjnej i widmowej.

1. WSTĘP

Język ASTEK /Analiza Statystyczna danych Eksperymentalnych/ stanowi próbę usystematyzowania obliczeń mających na celu analizę statystyczną danych eksperymentalnych, uzyskanych z rzeczywistego procesu przemysłowego /np. chemicznego/, identyfikację właściwości statycznych i dynamicznych obiektu w oparciu o metody regresji, korelacji i analizę widmową oraz badanie i weryfikację sporządzonego modelu. Analiza danych eksperymentalnych, sporządzenie modelu procesu stanowią zwykle pierwszy etap prac zmierzających do optymalizacji badanego procesu.

Struktura języka systematyzuje opis problemów obliczeniowych, formalizuje sposób korzystania ze zbioru procedur, ułatwia programowanie oraz umożliwia szybkie badanie wielu różnych wariantów analizy. Operatory logiczne pozwalają podejmować w trakcie obliczeń proste decyzje dotyczące selekcji danych pomiarowych, wyboru określonych parametrów itp.

Głównym celem projektu języka ASTEK jest maksymalne uproszczenie i ujednoczenie sposobu posługiwania się dużymi zbiorami danych, dostarczenie możliwie pełnej informacji o przebiegu i wynikach obróbki danych, giętkość w planowaniu i programowaniu problemów analizy statystycznej procesu.

Zestaw programów wchodzących w skład ASTEKA nie jest zamknięty. Autorzy przewidzieli możliwość szybkiego dołączania nowych operatorów, dla szczególnego procesu lub szczególnej struktury danych eksperymentalnych. Ograniczona pojemność pamięci operacyjnej maszyny cyfrowej ZAM 41 uniemożliwia zbytne rozszerzanie zestawu operatorów. Przewiduje się, że w miarę gromadzenia wyników doświadczeń eksploatacyjnych i potrzeb użytkowników koncepcja języka będzie modyfikowana, dostosowywana do wymagań i specyfiki większej liczby różnych procesów.

2. STRUKTURA JĘZYKA

Podstawowymi elementami ASTEKA są instrukcje, parametry, zbiory i liczby. Program składa się z deklaracji /parametry, formaty/ oraz z ciągu kolejno numerowanych instrukcji.

Na wstępie omówimy, nie wnikając w szczegóły, podstawowe pojęcia występujące w języku ASTEK.

2.1. Instrukcje

Ogólna postać instrukcji jest następująca^{*)}:

<instrukcja> ::= <nazwa instrukcji> sp <argumenty instrukcji>; nazwa instrukcji jest na ogół nazwą czynności, która ma być wykonana przez daną instrukcję.

W skład członu <argumenty instrukcji> wchodzi, oddzielone przeciskami, nazwy parametrów, zbiorów i formatów, liczby, a także symbole specjalne np. DODAJ +, ABC1, X1, X2;

^{*)}Do opisu struktury języka zastosowano notację Backus'a. Skrót sp oznacza spację.

2.2. Parametry

W wielu przypadkach zamiast operować w programie konkretnymi wartościami pewnych wielkości wygodniej jest posługiwać się parametrami identyfikowanymi w sposób symboliczny.

Każdemu z tych parametrów można nadawać w dowolnym punkcie programu dowolną wartość liczbową. Taki pośredni sposób dostępu do określonej wielkości pozwala wpływać dynamicznie na sposób wykonywania obliczeń /patrz np. pośrednie indeksowanie zbiorów/. Ponadto, parametry spełniają w ASTEKU rolę pamięci, w której można przechować wskaźniki, stałe, dane, wyniki uzyskane w efekcie działania pewnych instrukcji /np. poziomy ufności, wartości średnie, wariancje itp./. Parametr określa się w następujący sposób:

<parametr> := P <indeks parametru>

/Maksymalna wartość indeksu - 200/. Umiejętne wykorzystanie parametrów pozwala znacznie skrócić program.

Nadawanie wartości liczbowej parametrom może także odbywać się za pomocą instrukcji PRZYPI SZ.

Parametry można indeksować bezpośrednio np. P10, P150, lub pośrednio, za pomocą innego parametru, np. P(P10), P(P100). W przypadku pośredniego indeksowania wartość indeksu jest równa aktualnej wartości parametru występującego w nawiasach okrągłych. Jeśli np. P12 = 37, to zapis P(P12) jest równoważny P37. Zatem

<indeks parametru> ::= <liczba naturalna> |(P <liczba naturalna>)

Pośrednie indeksowanie znajduje zastosowanie w przypadku realizacji pewnych operacji w pętli, co ilustruje przedstawiony niżej przykład:

- 1/ PRZYPI SZ P1, 5;
- 2/ PĘTLA P1, 10, 3, 5;

3/ LAMBDATEST X (P1), P(P1);

4/ SKOCZ 2;

.

Instrukcja PRZYPIŚZ w pierwszym wierszu programu powoduje podstawienie liczby 5 w miejsce parametru P1. Instrukcja PĘ-TLA porównuje aktualną wartość parametru P1 z liczbą 10; w przypadku, gdy wartość P1 jest mniejsza od 10 następuje zwiększenie o 1 tego parametru oraz przejście do instrukcji 3. Jeśli natomiast parametr P1 jest równy lub większy od 10, jedyną czynnością jest przejście do wykonywania instrukcji 5. Jak łatwo zauważyć można w taki prosty sposób przetestować /Test Lambda - Kołmogorowa/ zbiory X5 do X10 i wyniki testu dla kolejnych zbiorów podstawiać w miejsce parametrów P5 do P10. Po przetestowaniu zbioru X10 następuje wyjście z pętli i przejście do wykonania instrukcji 5.

Liczba parametrów wykorzystywanych w programie deklaruje się na początku programu.

2.3. Zbiory

W analizie statystycznej rozpatrujemy zagadnienia, w których na podstawie znajomości cech odpowiednio wybranej próbki pewnej zbiorowości wysnuwa się wnioski dotyczące rozważanych cech dla całej zbiorowości.

W języku ASTEK przyjęliśmy pojęcie próby /zbioru/ za pojęcie podstawowe. Zbiór traktujemy jako ciąg liczb stanowiących wyniki pomiarów, wyniki pośrednie lub końcowe obliczeń względnie inne dane. Liczby te są uporządkowane w określony sposób według wartości argumentu, którym może być np. czas /zbiór wartości funkcji zależnej od czasu/. Zbiory, na których wykonywane są operacje mogą być wprowadzane do maszyny cyfrowej z taśmy papierowej lub z taśmy magnetycznej. Dla większości operacji maksymalna liczność zbioru wynosi ok. 5000 liczb /operacje, których argumentem są dwa zbiory/.

W przypadku operacji wykonywanych na jednym zbiorze /np. transformata Fouriera/ liczność zbioru może być równa 10000 liczb.

Wszystkie zbiory są identyfikowane na podstawie nazwy. Każdy zbiór stanowiący argument dowolnej operacji musi być nazwany według następującej reguły:

$\langle \text{identyfikator} \rangle ::= \langle \text{nazwa zbioru} \rangle \langle \text{indeks zbioru} \rangle$
 $\langle \text{nazwa zbioru} \rangle ::= \langle \text{litera} \rangle [\langle \text{litera} \rangle \langle \text{litera} \rangle \langle \text{litera} \rangle]$
 $\langle \text{indeks zbioru} \rangle ::= \langle \text{liczba naturalna} \rangle | (P \langle \text{indeks parametru} \rangle)$

Pośrednie indeksowanie zbioru ilustruje poniższy przykład: jeżeli aktualna wartość parametru P10 = 5, a P5 = 15, to zapis ABC (P10) jest równoważny ABC5 oraz ABC (P (P10)) jest równoważny ABC 15.

Nazwa jest nadawana zbiorom przy wprowadzaniu z taśmy papierowej lub przed odesłaniem z pamięci operacyjnej na taśmę magnetyczną.

Do wprowadzania zbiorów z taśmy papierowej, nazywania ich i odsyłania na taśmę magnetyczną służy instrukcja WCZYTAJ. Instrukcja WCZYTAJ powoduje wczytanie zbioru i nadanie mu określonej nazwy i indeksu. Do odesłania zbioru z pamięci operacyjnej na taśmę magnetyczną służy instrukcja PAMIĘTAJ.

ASTEK rejestruje nazwy zbiorów wykorzystywanych w trakcie obliczeń na liście nazw. Aktualizacja listy nazw odbywa się w chwili tworzenia nowego zbioru lub za pośrednictwem instrukcji AKTUALIZUJ, jeżeli zbiory zostały uprzednio zapisane na taśmie magnetycznej np. jako wyniki wcześniejszego cyklu obliczeń.

Liczby wchodzące w skład zbioru są przedstawione w skróconym zapisie zmiennoprzecinkowym wg następującej zasady:

mantysa	14 bitów	+	bit znaku
wykładnik	8 bitów	+	bit znaku

Wszystkie obliczenia są wykonywane na liczbach przedstawionych w pełnym zapisie zmiennoprzecinkowym. Konwersja na zapis skrócony odbywa się przed zapamiętaniem zbioru wynikowego oraz w czasie wprowadzania zbioru danych do maszyny cyfrowej.

3. ANALIZA WŁAŚCIWOŚCI STATYCZNYCH

Przed przystąpieniem do właściwej analizy statystycznej wyników eksperymentu należy poddać dane wstępnej obróbce w celu określenia charakteru ich rozkładu gęstości prawdopodobieństwa. Mogą nas interesować np. charakterystyki położenia środka grupowania, symetria, spłaszczenie krzywej rozkładu.

Do znajdowania średniej i wariancji próbki służą instrukcje ŚREDNIA i WARIANCJA. Pewnych informacji o rozkładzie może dostarczyć także położenie wartości ekstremalnej krzywej rozkładu.

Znajdowanie ekstremum krzywej odbywa się za pośrednictwem instrukcji EKSTREMUM.

Jak wiadomo w wielu metodach analizy statystycznej przyjmuje się założenie, że rozkład danych pomiarowych jest zbliżony do normalnego. Duże odchylenia od rozkładu normalnego mogą w takich przypadkach podważyć celowość stosowania tych metod. W każdym przypadku należy określić charakter rozkładu /służy do tego instrukcja DYSTRYBUANTA/ i oszacować wielkość odchylenia rozkładu danych pomiarowych od rozkładu normalnego. W tym celu wykorzystuje się testy istotności, wśród których najczęściej stosowanymi są test χ^2 /instrukcja CHITEST/ oraz test λ Kołmogorowa /instrukcja LMBDATEST/.

Wyniki testów mogą wpływać na dalszy przebieg analizy /np. jeśli przyjęty poziom istotności jest mniejszy od poziomu oszacowanego w wyniku testu, nie ma podstaw do odrzucenia hipotezy o normalności rozkładu prawdopodobieństwa/. Podejmowanie takich prostych decyzji umożliwia instrukcja BADAJ /porównywanie dwóch wartości liczbowych/.

Głównym zagadnieniem, które należy rozważyć w ramach analizy statycznej jest analiza regresyjna. W bardzo ogólnym zarysie zadania analizy regresyjnej przedstawiają się następująco. Niech będzie dana wielowymiarowa przestrzeń danych eksperymentalnych. Chcemy przeprowadzić twór algebraiczny przez punkty otrzymane z eksperymentu. Może nas interesować "rozprostowanie" tego tworu na hiperpłaszczyznę. Rozwiązanie powyższych zadań osiąga się najczęściej drogą minimalizacji sumy kwadratów odchyień punktów eksperymentalnych od powierzchni /względnie od hiperpłaszczyzny/.

W języku ASTEK analizę regresyjną dla modeli liniowych można przeprowadzić za pomocą instrukcji REGRESJA.

Niżej podamy krótki opis działania instrukcji.

Niech $Y_1, \dots, Y_M, X_1, \dots, X_K$ oznaczają zbiory obserwowanych cech. Chcemy znaleźć metodą najmniejszych kwadratów wektory

$$\bar{\delta}_j = [\delta_{j0}, \delta_{j1}, \dots, \delta_{jK}]$$

parametrów modelu liniowego

$$Y_j = \delta_{j0} + \delta_{j1} X_1 + \dots + \delta_{jK} X_K \quad j = 1, 2, \dots, M \quad /1/$$

Kolejność w jakiej są uszeregowane cechy X_1, \dots, X_K w równaniu /1/ zależy od ich istotności w szukanym modelu. Cechy mające duży wpływ na strukturę modelu muszą występować na początku równania.

Parametry δ_{ij} znajdujemy z warunków minimalizacji funkcjonalów.

$$\sum_{i=1}^N (y_{ij} - \delta_{j0} + \delta_{j1} X_{i1} + \dots + \delta_{jK} X_{iK})^2 ; j = 1, \dots, M$$

Znalezienie $\delta_{j0}, \dots, \delta_{jK}$ minimalizujących powyższe funkcjonalny jest równoważne rozwiązaniu układu równań liniowych.

$$X^T X G = X^T Y \quad /2/$$

/2/

gdzie

$$X = \begin{bmatrix} 1 & X_{11} & \dots & X_{1K} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 1 & X_{N1} & \dots & X_{NK} \end{bmatrix}$$

jest macierzą obserwacji cech występujących po prawej stronie równania /1/

$$G = \begin{bmatrix} \delta_{10} & \dots & M_0 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \delta_{1K} & \dots & M_K \end{bmatrix}$$

jest macierzą szukanych parametrów

$$Y = \begin{bmatrix} Y_{11} & \dots & Y_{1M} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ Y_{N1} & \dots & Y_{NM} \end{bmatrix}$$

jest macierzą obserwacji cech występujących po lewej stronie równania modelu

Układ /2/ rozwiązywany jest metodą eliminacji bez wyboru elementu głównego. Po pierwszym kroku eliminacyjnym otrzymamy macierz układu w postaci:

$$\begin{bmatrix} 1 & \bar{x}_1 & \dots & \bar{x}_K \\ 0 & N\delta_1^2 & N\delta_1\delta_2r_{12} & \dots & N\delta_1\delta_Kr_{1K} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & N\delta_K\delta_1r_{K1} & \dots & \dots & N\delta_K^2 \end{bmatrix}$$

gdzie \bar{x}_j /j=1,...,K/ są wartościami średnimi cech,

δ_1^2 - wariancje cech X_1 ,

r_{iK} są współczynnikami korelacji cząstkowej rzędu zerowego

Na tym etapie sprawdza się czy wariancje kolejnych cech są mniejsze od zadanego z góry dla każdej cechy poziomu szumów - jeśli tak, to usuwa się tę cechę oraz wszystkie następn...

ne z modelu. Następuje zatem redukcja macierzy układu. Dla nowej macierzy kontynuuje się proces eliminacji. Po n-tym kroku eliminacyjnym wyrazy leżące na głównej przekątnej mają następującą postać:

$$c_{ii} = N\delta_K^2 (1 - R_{i:12,\dots,n}); \quad i = n+1, n+2, \dots, K$$

gdzie $R_{i:12,\dots,n}$ jest współczynnikiem korelacji całkowitej cechy X_i względem cech X_1, X_2, \dots, X_n . Jeżeli współczynnik $R_{i:12,\dots,n}$ jest mniejszy od zadanego z góry poziomu, to i-tą cechę odrzuca się z modelu razem ze wszystkimi następnymi. Eliminację kontynuujemy dalej dla nowej, zredukowanej macierzy.

Przez sprowadzenie macierzy do postaci trójkątnej i postępowanie odwrotne metody eliminacji znaleźliśmy szukane wektory $\bar{\gamma}_j / j=1 \dots K/$.

Za pomocą znalezionych parametrów możemy obliczyć z modelu wartości

$$y_{jl}^* = \bar{\gamma}_{j0} + \bar{\gamma}_{j1} x_{j1} + \bar{\gamma}_{j2} x_{j2} + \dots + \bar{\gamma}_{jK} x_{jK}; \quad l=1, \dots, N$$

oraz wariancje resztowe.

$$S_i^2 = \sum_{t=1}^N \frac{(y_{jt}^* - y_{jt})^2}{N} \quad j = 1, \dots, m$$

Duże znaczenie dla dalszej analizy ma obliczenie macierzy $(X^T X)^{-1}$ ocen błędów parametrów $\bar{\gamma}_{ij}$. Dzięki temu, że macierz $(X^T X)^{-1}$ jest wyliczana równoległe z procesem eliminacji, czas działania całej instrukcji REGRESJA uległ prawie dwukrotnemu skróceniu. Ze względu na ograniczoną pamięć operacyjną i bębnową maszyny cyfrowej, liczba zbiorów Y nie może przekraczać 10, a liczba zbiorów X - wartości 100.

W praktyce najczęściej rozpatruje się modele liniowe o bardziej złożonej strukturze, np. w postaci formy kwadratowej

$$Y = \gamma_0 + \sum_{i=1}^n \gamma_i X_i + \sum_{i < j} \gamma_{ij} X_i X_j + \sum_{i=1}^n \gamma_{ii} X_i^2 \quad /3/$$

Aby model ten sprowadzić do postaci /1/ trzeba wykonać działania na zbiorach:

$$X_i X_j = X_{ij} \quad X_i^2 = X_{ii}$$

Wstawiając X_{ij} w miejsce $X_i X_j$ oraz X_{ii} zamiast X_i^2 otrzymamy model w postaci /1/.

Potrzebne są więc instrukcje pozwalające na wykonywanie najprostszych działań arytmetycznych na zbiorach. Do dodania i odejmowania zbiorów służy instrukcja DODAJ. Za pomocą instrukcji MNÓŻ możemy wykonywać mnożenie lub dzielenie zbiorów.

Aby zweryfikować model otrzymany w wyniku analizy regresyjnej, należy przetestować go za pomocą testu Fishera. W tym celu znajduje się stosunek wariancji resztowej do wariancji charakteryzującej błąd doświadczenia

$$F = \frac{S_R^2}{S_{\text{eksp}}^2}$$

Jeżeli dana jest ilość stopni swobody odpowiadająca wariancji błędu /ilość stopni swobody odpowiadająca wariancji resztowej jest policzona w REGRESJI/, to można znaleźć prawdopodobieństwo P , że zmienna losowa Z Fishera jest większa od F .

Jeżeli obliczony model ma strukturę formy kwadratowej /3/, to można przyjąć, że równanie modelu określa w przestrzeni n -wymiarowej twór rzeczywisty stopnia drugiego. Znajomość wartości własnych macierzy formy kwadratowej pozwala na skłasyfikowanie tworu oraz umożliwia sprowadzenie równania do postaci kanonicznej. Do obliczenia wartości i wektorów własnych służy instrukcja WŁASNEWARTIWEKT. Algorytm obliczeniowy

oparty jest na przybliżonej metodzie Rutishausera. Badania tworu mogą iść też w kierunku znalezienia jego ekstremów. Aby znaleźć ekstremum wystarczy rozwiązać układ równań

$$\frac{d}{dx_1} \left(\delta_0 + \sum_{i=1}^n \delta_i x_i + \sum_{i < 1} \delta_{i1} x_i x_1 + \sum_{i=1}^n \delta_{ii} x_i^2 \right) = 0$$

$$i = 1, \dots, n$$

Na tej właśnie zasadzie oparty jest algorytm instrukcji MAX-MIN.

4. ANALIZA WŁAŚCIWOŚCI DYNAMICZNYCH

Wśród programów ASTEKA umożliwiających analizę właściwości dynamicznych procesów znajdują się: program obliczający współczynniki korelacji /KORELACJA/, programy obliczania transformaty Fouriera /FFT/, gęstości widmowych /WIDMO/ i odpowiedzi impulsowych /IMPULS/.

Obliczanie współczynników korelacji wzajemnej i własnej odbywa się wg zależności definicyjnej.

$$R_{xy}(r) = \frac{\sum_{k=0}^{N-1-r} (x_k - m_x)(y_{k+r} - m_y)}{\sqrt{\delta_x^2 \delta_y^2}} \quad r = 0, 1 \dots N-1$$

gdzie: m_x i m_y - oznaczają odpowiednio średnią zbioru X i zbioru Y

δ_x^2 i δ_y^2 - oznaczają odpowiednio wariancje zbioru X i zbioru Y

N - liczność próbki.

Jak łatwo zauważyć, aby obliczyć $R_{xy}(r)$ dla próbek złożonych z N liczb należy wykonać w przybliżeniu $\frac{N}{4} \left(\frac{N}{2} + 3 \right)$ operacji mnożenia. W praktyce często mamy do czynienia z danymi pomiarowymi złożonymi z wielu tysięcy liczb. Zatem obliczanie funkcji korelacji staje się, nawet w przypadku stosowania bardzo szybkiej maszyny cyfrowej, operacją bardzo czasochłonną. Tę samą funkcję korelacji można obliczyć znacznie szybciej metodą pośrednią, wykorzystując znane właściwości mnożenia spłotowego dwóch funkcji w dziedzinie częstotliwości. Znaczny zysk czasowy osiąga się w tym przypadku dzięki wykorzystaniu algorytmu szybkiego obliczania transformaty Fouriera, opracowanego przez Cooley'a i Tukey'a. Idee tego algorytmu przedstawiamy zwięźle poniżej.

Rozpatrujemy próbkę X_k złożoną z N liczb /dla uproszczenia rozważań założymy, że $N = 2^n$ /. Zadanie polega na znalezieniu dyskretnej transformaty Fouriera opisanej zależnością

$$A_r = \sum_{k=0}^{N-1} X_k \exp \left(-2 \pi jrk/N \right) \quad r = 0, \dots, N-1 \quad /4/$$

X_k dzielimy na dwie równe próbki / $\frac{N}{2}$ liczb/ Y_k i Z_k według następującej zasady:

$$Y_k = X_{2k}$$

$$Z_k = X_{2k+1} \quad k = 0, 1, 2, \dots, \frac{N}{2} - 1$$

Dyskretne transformaty Fouriera dla Y_k i Z_k mają postać:

$$B_r = \sum_{k=0}^{\frac{N}{2} - 1} Y_k \exp \left(-4 \pi jrk/N \right) \quad /5/$$

$$C_r = \sum_{k=0}^{\frac{N}{2} - 1} Z_k \exp(-4 \pi j r k / N) \quad /6/$$

$$r = 0, 1, \dots, \frac{N}{2} - 1$$

Zatem:

$$A_r = \sum_{k=0}^{\frac{N}{2} - 1} \left\{ Y_k \exp(-4 \pi j r k / N) + Z_k \exp\left[-\frac{2 \pi j r}{N} (2k + 1)\right] \right\} \quad /7/$$

lub

$$A_r = \sum_{k=0}^{\frac{N}{2} - 1} Y_k \exp(-4 \pi j r k / N) + \exp(-2 \pi j r / N) \sum_{k=0}^{\frac{N}{2} - 1} Z_k \exp(-4 \pi j r k / N) \quad /8/$$

to znaczy

$$A_r = B_r + \exp(-2 \pi j r / N) \cdot C_r; \quad 0 \leq r < \frac{N}{2} \quad /9/$$

Ze względu na okresowość transformat B_r i C_r /okres $N/2$ / można w wyrażeniu /9/ w miejsce r podstawić $r + N/2$. Stąd

$$\begin{aligned} A_{r+N/2} &= B_r + \exp(-2 \pi j \left[r + \frac{N}{2}\right] / N) C_r = \\ &= B_r - \exp(-2 \pi j r / N) \cdot C_r, \end{aligned}$$

$$0 \leq r \leq N/2$$

/10/

Analizując wyrażenie /9/ i /10/ stwierdzamy, że transformata Fouriera funkcji dyskretnej X_k może być obliczona na podstawie transformat funkcji dyskretnych Y_k i Z_k . Czas obliczania transformaty wg zależności definicyjnej /4/ jest proporcjonalny do N^2 . Wykorzystując zależności /9/ i /10/ stwierdzamy, że ilość podstawowych operacji arytmetycznych /a zatem czas obliczeń/ uległa zmniejszeniu i jest teraz proporcjonalna do $2 \left(\frac{N}{2}\right)^2$.

Przedstawiony wyżej sposób postępowania można kontynuować, dokonując n kolejnych podziałów próbki i dochodząc w rezultacie do podpróbek złożonych z dwóch elementów. W efekcie, całkowity czas obliczeń będzie proporcjonalny do $2N \log_2 N$.

Zatem w przypadku gdy $N = 1024$, czas obliczania transformaty Fouriera z wykorzystaniem omówionego algorytmu będzie ok. 50 razy krótszy niż w przypadku stosowania algorytmu konwencjonalnego.

Omówiony algorytm transformaty Fouriera wykorzystuje się także do wyznaczania funkcji gęstości widmowej na podstawie funkcji korelacji. Pełną identyfikację właściwości dynamicznych procesów w zakresie liniowości uzyskuje się obliczając odpowiedź impulsową /instrukcja IMPULS/.

5. WYNIKI DZIAŁANIA PROGRAMU

Wyniki obliczeń mogą być wyprowadzane na drukarkę w postaci wykresu lub zbiorów liczbowych. W pierwszym przypadku należy korzystać z instrukcji WYKRES, w drugim - z instrukcji WYPISZ.

Język ASTEK daje możliwość drukowania zbiorów liczbowych w dowolnej postaci. Postać wydrukowanego zbioru ustala się za pomocą deklaracji formatu.

Deklaracja formatu wygląda następująco:

```
<deklaracja formatu> ::= <nazwa formatu>  
                           <parametry formatu>;
```

gdzie:

< nazwa formatu > ::= F < numer formatu >

< numer formatu > jest dowolną liczbą naturalną mniejszą od 200

oraz

< parametry formatu > ::= < ilość cyfr części całkowitej > ,
< ilość cyfr części ułamkowej > ,
< ilość cyfr znaczących > ,
< ilość kolumn > .

np. F3/ 2, 4, 6, 5;

Format jest argumentem instrukcji WYPISZ. Instrukcją, która kończy wykonywanie programu jest STOP.

6. ZESTAW INSTRUKCJI JĘZYKA ASTEK

NAZWA INSTRUKCJI	FUNKCJA
WCZYTAJ	Wczytanie danych /zbiorów/ z taśmy papierowej i zapisanie ich na wybranej jednostce taśmy magnetycznej.
WYPISZ	Wypisanie zbioru lub parametru na drukarce.
KREŚL	Kreślenie wykresu /jednocześnie można wykreślać do pięciu krzywych/.
PAMIĘTAJ	Zapamiętanie zbioru /wynik operacji/ na wybranej jednostce taśmy magnetycznej.
PRZY PISZ	Nadawanie parametrowi wartości liczbowej.
PĘTLA	Blok umożliwiający realizację pętli programowej.

NAZWA INSTRUKCJI	FUNKCJA
DODAJ { + - }	Dodawanie lub odejmowanie zbiorów, parametrów lub liczb.
MNÓŻ { * / }	Mnożenie lub dzielenie zbiorów, parametrów lub liczb.
EXPONENT	Funkcja wykładnicza.
SQ	Pierwiastek kwadratowy.
LOGARYTM	Logarytm naturalny.
SKOCZ	Skok do określonego miejsca w programie.
AKTUALIZUJ	Aktualizacje listy zbiorów znajdujących się na określonej jednostce taśmy magnetycznej.
KORELACJA	Obliczenie współczynników korelacji.
BADAJ	Porównywanie dwóch wielkości liczbowych lub parametrów.
EKSTREMUM	Znajdowanie wartości maksymalnej lub minimalnej w zbiorze.
REGRESJA	Obliczenie współczynników regresji.
WEKTORY	Znajdowanie wektorów i wartości własnych.
ŚREDNIA	Obliczenie wartości średniej zbioru.
WARIANCJA	Obliczenie wariancji zbioru.
MINMAX	Znajdowanie ekstremum tworu geometrycznego /forma kwadratowa/.
FTEST	Test Fishera.

NAZWA INSTRUKCJI	FUNKCJA
LAMBDA TEST	Test Kołmogorowa.
CHITEST	Test χ^2 .
DYSTRYBUANTA	Znajdowanie dystrybuanty empirycznej.
FFT	Obliczenie transformaty Fouriera.
WIDMO	Obliczenie funkcji gęstości widmowej.
IMPULS	Znajdowanie odpowiedzi impulsowej układu dynamicznego.
PRZERWIJ	Przerwanie obliczeń w dowolnym miejscu programu.
KONTYNUUJ	Kontynuacja przerwanych /instrukcja PRZERWIJ/ obliczeń.
STOP	Koniec obliczeń.

Uwaga: Translator identyfikuje instrukcje na podstawie dwóch pierwszych liter nazwy instrukcji. Dokładny opis każdej instrukcji, tj. nazwy instrukcji i argumentów, znajduje się w "Podręczniku programowania języka ASTEK".

7. WYMAGANY ZESTAW MASZYNY ZAM 41

Pamięć operacyjna - 20 K /K = 1024 słowa/
Pamięć taśmowa - przynajmniej 2 jednostki
Drukarka wierszowa
Monitor dalekopisowy
Czytnik taśmy papierowej

8. PRZYKŁAD PROGRAMU W JĘZYKU ASTEK

Należy obliczyć metodą najmniejszych kwadratów wartości parametrów G_0, G_1, \dots, G_9 modelu liniowego

$$Y_1 = G_0 + G_1 X_0 + \dots + G_7 X_6 + G_8 R_1 = G_9 R_2$$

oraz wariancję resztową S_1^2 .

Zbiory X_0, \dots, X_6 mają być wozytane z taśmy papierowej, a zbiory Y_1, R_1, R_2 obliczone wg wzorów:

$$R_1 = (1 - X_4 - X_5 - X_6 - X_1) \cdot 0.06952$$

$$R_2 = X_4 \cdot 5.5$$

$$Y_1 = X_1/R_1$$

Zadanie to realizuje poniższy program.

```

* ASTEK                               /POCZĄTEK PROGRAMU/
  P10;                                /DEKLARACJA PARAMETRÓW/
F1/ 3,3,6,5;                          /DEKLARACJA FORMATU/
  T11;                                /DEKLARACJA TAŚMY ROBOCZEJ/
*
1/ PR 0,P2;
2/ WC T12, X /P2/;
3/ FE P2,5,2,4;                       /WCZYTANIE W PĘTLI 7 ZBIORÓW X0, X1.....X6/
4/ DO -,1,X4;
5/ DO -,*,X5;
6/ DO -,*,X6;
7/ DO -,*,X1;
8/ MN .,*,0.06952;
9/ PAM T12,R1;                         /ZAPAMIĘTANIE ZBIORU R1.
                                         R1 = (1-X4-X5-X6-X1) * 0.06952/
10/ MN ., X4, 5.5;                    /R2 = X4 * 5.5/
11/ PAM T12, R2;
12/ MN /, X1, R1;                     /Y1 = X1 DZIELONE PRZEZ R1/
13/ PAM T12, Y1;
14/ REG Y1: X /0-6/, R1, R2 : 3, P1, T21; /OBLICZONY ZOSTAJE
                                         ZBIÓR GAMM1, KTÓREGO ELEMENTAMI SĄ WSPÓŁ-
                                         CZYNNIKI G0, G1,... G9 MODELU LINIOWEGO
                                         Y1 = G0 + G1X0 + G2X1 + ... + G7X6 + G8R1
                                         + G9R2,
```

ORAZ WARIANCJA RESZTOWA P1/

15/ WY P1, F1; /WYDRUK WARIANCJI RESZTOWEJ/
16/ STOP; /KONIEC PROGRAMU/

KONIEC

9. ZAKOŃCZENIE

Opisany język ASTEK stanowi próbę sformułowania i realizacji języka specjalizowanego do programowania analizy statystycznej danych eksperymentalnych w celu identyfikacji właściwości statycznych i dynamicznych procesów ciągłych. Język ten znacznie ułatwia programowanie zadań tego typu i przyspiesza proces poszukiwania modelu matematycznego procesu.

Doświadczenia zebrane w trakcie użytkowania posłużyły do opracowania rozszerzonej i udoskonalonej wersji języka ASTEK II /rozszerzona lista instrukcji, ulepszony translator/ umożliwiającej programowanie szerszej grupy zagadnień z zakresu analizy statystycznej. Szczegółowy opis języka ASTEK II zawarty jest w opracowaniu Zakładu Sterowania IMM pt: "Język ASTEK II - instrukcja programowania".

АСТЕК - ЯЗЫК ДЛЯ СТАТИСТИЧЕСКОГО АНАЛИЗА ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ

Резюме

Язык АСТЕК это попытка систематизировать вычисления, целью которых является статистический анализ экспериментальных данных, полученных из реального промышленного процесса (например химического), идентификация статичных и динамических свойств объекта, базируемых на методах регрессии, корреляции и спектральном анализе, а также исследование и проверка приготовленной модели.

Структура языка систематизирует описание вычислительных проблем, формализует способ использования набора процедур, облегчает программирование и позволяет быстро исследовать много различных вариантов анализа. Логические операторы позволяют в ходе вычислений принимать решение относительно селекции измерительных данных, выбирать определённые параметры и т.д.

Главной целью проекта языка АСТЕК является максимальное упрощение и унификация способа применения больших множеств данных, поставка возможно полной информации о ходе и результатах обработки данных, гибкость планирования и программирования проблем статистического анализа процесса.

Набор программ заключенных в составе АСТЕКА не закрыт. Авторы предвидели возможность быстрого присоединения новых операторов для особенного процесса или особенной структуры экспериментальных данных.

ASTEK LANGUAGE FOR STATISTIC ANALYSIS OF EXPERIMENTAL DATA

Summary

The language ASTEK is a trial to systematize computations aiming at statistic analysis of experimental data, obtained from an industrial process /e.g. chemical/, identification of static and dynamic properties of an object based on regression and correlation methods and spectrum analysis, as also investigation and verification of a model construction.

The structure of the language systematizes the description of computation problems, formalizes the way of using procedures, enables the programming and permits a fast investigation of many different variants of an analysis.

Logic operators allow simple decision making on the selection of measurement data during computation, and choosing the determined parameters and so on.

The main features of the language ASTEK are: a maximal simplification and uniformity of the way in which big sets of data are used, a supply of possibly full information about the course and results of data processing, a flexibility of planning and programming problems of statistical analysis process.

The set of ASTEK programs is not yet closed. The authors have foreseen the possibility of a quick adjoining of new operators for a special process or a special structure of experimental data.

SYSTEM MAGAZYNOWANIA I AKTUALIZACJI
DOKUMENTÓW I JEGO ZASTOSOWANIE
W PROCESIE URUCHAMIANIA PROGRAMÓW

Jacek WITASZEK

Pracę złożono 9.07.1969

W pracy opisano zaimplementowany na maszynie ZAM 41 system magazynowania i aktualizacji dokumentów /SMAD/. Do magazynowania tekstów, w szczególności programów, wykorzystano pamięci taśmowe. System ten umożliwia między innymi aktualizowanie oraz wykonywanie translacji programów zapisanych na taśmie magnetycznej. Przedstawiono zasady korzystania z omawianego systemu oraz zakres jego zastosowań.

1. WSTĘP

Jak wiadomo jednym z poważniejszych zagadnień, przed którymi staje programista w trakcie opracowywania programu, jest proces uruchamiania programu.

W procesie tym mamy do czynienia z wielokrotnym powtarzaniem pewnego cyklu czynności, tzw. cyklu uruchamiania.

W cyklu uruchamiania możemy wyróżnić trzy kolejne fazy:

- faza 1: wprowadzenie kolejnej wersji programu do maszyny i jej wykonanie /na ogół wielokrotne/,
- faza 2: analiza wyników wykonania programu i lokalizacja ewentualnych błędów,
- faza 3: poprawienie zlokalizowanych błędów.

Wynikiem każdego cyklu jest nowa wersja programu, która z kolei staje się punktem wyjścia dla następnego cyklu uruchamiania. Proces uruchamiania zostaje zakończony w chwili, gdy czynności drugiej fazy wykażą pełną poprawność programu.

Uruchamianie programów bez zastosowania specjalnych środków pomocniczych jest procesem żmudnym i długotrwałym. Pochłania ono w zależności od stopnia złożoności programu oraz doświadczenia programisty od 30 do 70 procent całości czasu przeznaczanego na opracowanie tego programu. Uwzględniając ponadto fakt, że w procesie uruchamiania programu uczestniczy nie tylko człowiek, lecz także maszyna wraz z odpowiednim zestawem urządzeń pomocniczych dochodzimy do wniosku, iż udział kosztów uruchamiania w kosztach programu jest odpowiednio większy /40-90%/. Stąd widać jak duży wpływ na efekty ekonomiczne pracy ośrodka obliczeniowego ma zagadnienie wyposażenia EMC w specjalne środki do uruchamiania programów, które umożliwiłyby zarówno zmniejszenie liczby cykli uruchamiania jak i skrócenie poszczególnych faz każdego cyklu.

Jednym z tego rodzaju środków, nie wyczerpującym rzecz jasna wszystkich w tym względzie możliwości, jest System Magazynowania i Aktualizacji Dokumentów /SMAD/ będący przedmiotem niniejszego artykułu.

2. OGÓLNA CHARAKTERYSTYKA SMAD

SMAD jest systemem programowym wchodzącym w skład oprogramowania S0 141 maszyn ZAM 41, a przeznaczonym do wykonywania operacji na dokumentach. Przez dokument rozumiemy tutaj dowolny w zasadzie tekst o opisanej niżej stronicowo-wierszowej strukturze. Nośnikiem informacji wykorzystywanym przez SMAD do magazynowania dokumentów jest taśma magnetyczna /TM/.

Właściwości funkcjonalne SMAD pozwalają jego użytkownikowi na:

- zapisanie dokumentu na TM,

- odszukanie początku żądanego fragmentu dokumentu zapisanego na TM,
- wykorzystanie informacji, zawartej w odszukanym uprzednio fragmencie dokumentu zapisanego na TM, jako danych wejściowych przez dowolny program,
- aktualizację zapisanego na TM dokumentu połączoną z wydrukowaniem zmienianych jego partii^{**/}.

Zanim przystąpimy do bardziej szczegółowego rozpatrzenia wspomnianych właściwości zajmiemy się omówieniem struktury dokumentów wymaganej przez SMAD.

3. STRONICOWO-WIERSZOWA STRUKTURA DOKUMENTÓW^{***/}

Jak wspomnieliśmy wyżej, dokument będący obiektem działania SMAD musi posiadać określoną strukturę stronicowo-wierszową.

Podział dokumentu na wiersze wyznaczany jest znakami interpretowanymi jako przejście do nowego wiersza /"nowa linia", "koniec karty" itp./ - zatem zadawany jest on automatycznie w trakcie sporządzania dokumentu na urządzeniu perforującym. Określenie podziału dokumentu na strony wymaga wstawienia w miejscach podziału pewnego standartowego tekstu. Tekst ten, w przeciwieństwie do reszty tekstu na stronie zwanej treścią strony, będziemy nazywali nagłówkiem strony /rys. 1/. W skład nagłówka strony wchodzi tzw. identyfikator strony, który można traktować jako uogólniony numer strony. Identyfikator ten może składać się zarówno z cyfr jak i liter, a ponadto grupy znaków literowo-cyfrowych mogą być poprzedzane kropkami.

^{**/} Znane są systemy o zbliżonych właściwościach, na przykład [1], [2].

^{***/} Opisana tutaj struktura dokumentów oparta jest na wprowadzonej przez mgr Hannę i Jerzego Mysiorów koncepcji podziału programów na strony. Istnieje też w ramach SO 141 cały zestaw programów przygotowywania i opracowywania dokumentów o ww strukturze.

W sumie liczba znaków w identyfikatorze łącznie z kropkami nie może przekroczyć ośmiu /identyfikator strony na rys. 1 składa się z trzech znaków/.

Treścią strony może być dowolny tekst spełniający następujące warunki:

- liczba wierszy tego tekstu jest mniejsza od 1000
- liczba znaków w wierszu jest mniejsza od 54
- żaden z wierszy nie rozpoczyna się od dwóch dwukropków.

Dwa pierwsze ograniczenia są konsekwencją przyjęcia pewnej określonej formy dla tworzonej przez SMAD dokumentacji. Trzecie ograniczenie wynika z konieczności odróżnienia tekstu określającego czynność, którą ma wykonać SMAD, tzw. operację, od tekstu stanowiącego przedmiot tej operacji. Przyjęto zasadę, że wiersz rozpoczynający się od dwóch dwukropków oznacza operację.

Rzecz jasna, że przyjęcie omówionej wyżej struktury oraz związanych z nią ograniczeń zawęży w jakimś stopniu zakres zastosowań SMAD. Niemniej jednak dotychczasowe zastosowania SMAD do uruchamiania programów w językach PJEG, PJES, SAS pokazują, że ograniczenia te nie są istotne.

4. OPERACJE SMAD

Czynności jakie ma na danym dokumencie wykonać SMAD zadaje się za pomocą ciągu operacji. Poszczególne operacje wczytywane są z urządzenia wejściowego, określonego w ozołówce problemu [3] /na przykład z czytnika taśmy perforowanej lub z czytnika kart/ i wykonywane w kolejności ich wczytywania.

Zależnie od charakteru wykonywanej pracy SMAD współpracuje z jedną lub dwiema stacjami TM. I tak do zapisania dokumentu na TM lub odszukania określonego miejsca w zapisanym na TM dokumencie wystarcza jedna stacja. Do przeprowadzenia aktualizacji dokumentu potrzebne są dwie stacje taśm magnetycznych.

Na jedną z nich zakłada się wówczas taśmę ze starą wersją dokumentu, a na drugą - taśmę przeznaczoną na nową wersję dokumentu.

Informacje o tym, które ze stacji zostały przeznaczone do współpracy ze SMAD oraz jakie jest przeznaczenie założonych na nie taśm są przekazywane za pośrednictwem czołówki problemu.

5. OPERACJE ZWIĄZANE Z WCZYTYWANIEM I WYSZUKIWANIEM INFORMACJI W DOKUMENCIE

Wczytywanie dokumentów na taśmę magnetyczną odbywa się za pomocą operacji:

:: WCZYTAJ DOKUMENT

Wykonanie tej operacji powoduje umieszczenie na TM ciągu dokumentu zapisanego po tej operacji. Koniec dokumentu wyznacza operacja

:: KONIEC WCZYTYWANIA

Powoduje ona zakończenie wczytywania, odwiniecie TM oraz zakończenie pracy SMAD.

Jednocześnie z wczytywaniem następuje wydrukowanie wczytwanego dokumentu na drukarce wierszowej czyli sporządzenie jego dokumentacji. Dokumentacja ta jest redagowana w dwóch szpaltach i w takiej formie, aby nowa strona rozpoczynała się zawsze u góry arkusza. Dzięki temu, po rozdzieleniu poszczególnych arkuszy, złożeniu każdego arkusza na pół i spięciu arkuszy razem, otrzymuje się dokumentację w formie książki.

Ponieważ liczba wierszy w arkuszu, czyli jego wysokość zależy od rodzaju założonego w drukarce papieru, należy każdorazowo przed przystąpieniem do wczytywania /lub aktualizacji/ dokumentu poinformować SMAD o aktualnej wysokości arkusza papieru drukarki. Służy do tego celu operacja

:: ARKUSZ : n

/gdzie n jest liczbą oznaczającą liczbę wierszy w arkuszu/,
która powinna poprzedzić operację "WCZYTAJ DOKUMENT".

Jeżeli dokumentacja nie jest potrzebna można zrezygnować
z jej drukowania. W tym celu należy operację " WCZYTAJ DOKU-
MENT" poprzedzić operacją

:: NIE DRUKUJ

Jeżeli natomiast poprzedzić operację "WCZYTAJ DOKUMENT"
operacją

:: DRUKUJ : n

gdzie n jest liczbą całkowitą większą od zera, wówczas doku-
mentacja zostanie sporządzona w n egzemplarzach.

Informacje zawarte w zapisanym na TM za pośrednictwem SMAD
dokumencie mogą z kolei stanowić dane dla dowolnego programu,
w szczególności dowolnego translatora języka programowania.
Należy tutaj zaznaczyć, że ta możliwość nie wymaga jakiegokol-
wiek dodatkowego uwzględnienia w samym programie. W systemie
ZAM 41 przyjęto bowiem taką realizację operacji wejścia i wyjś-
cia, że z punktu widzenia programu /translatora/ obojętne jest
skąd pochodzą dane; przesyłanie danych organizuje system opera-
cyjny zgodnie z informacjami podanymi w czołówce problemu [3].

Przed rozpoczęciem wykonywania programu należy na ogół od-
szukać na taśmie magnetycznej początek jego danych.

Do wyszukiwania informacji służą operacje:

:: ODWIŃ

oraz

:: PRZEWIŃ DO STRONY : I

gdzie I jest identyfikatorem strony.

Wykonanie pierwszej z tych operacji powoduje odwiniecie taśmy magnetycznej do pierwszej strony dokumentu.

W wyniku wykonania drugiej operacji taśma zostaje przewinięta do strony o identyfikatorze I. Należy tutaj dodać, że "przewijanie" może odbywać się tylko w jednym kierunku, w kierunku końca taśmy, a więc w kierunku coraz to dalszych stron dokumentu. Zatem, gdy zachodzi potrzeba cofnięcia się do strony wcześniejszej, należy użyć pary operacji: "ODWIŃ", "PRZEWIŃ DO STRONY". Podobnie należy postąpić w wypadku, gdy położenie głowic stacji w stosunku do taśmy nie jest znane, np. po założeniu szpuli na stację. Koniec procesu odszukiwania informacji wyznacza operacja

:: KONIEC ODSZUKIWANIA

która jednocześnie kończy prace SMAD.

Jak wynika z podanego wyżej opisu operacji "ODWIŃ" i "PRZEWIŃ DO STRONY" najmniejszą, możliwą do odszukania za ich pomocą jednostką informacji jest strona. To ograniczenie, podyktowane prostotą realizacji, nie zubaża w jakiś istotny sposób możliwości SMAD. Można bowiem zawsze tak podzielić dokument na strony, by każdy jego fragment, który tworzy pewną logiczną całość i tym samym może stanowić dane dla jakiegoś programu, rozpoczynał się od nowej strony.

6. AKTUALIZACJA DOKUMENTU

Istotną właściwością SMAD, decydującą w głównej mierze o jego użyteczności, jest możliwość wykorzystywania tego systemu do aktualizowania dokumentów zapisanych na taśmach magnetycznych.

Jak już wspomnieliśmy do tego typu pracy potrzebne są dwie stacje i dwie taśmy magnetyczne. W dalszym ciągu taśmę ze starą wersją dokumentu będziemy oznaczali TM1, a taśmę przeznaczoną na nową wersję TMO.

Przebieg procesu aktualizacji określa ciąg złożony z przeznaczonych specjalnie do tego celu operacji, tzw. operacji aktualizujących.

Każda z operacji aktualizujących określa fragment starej wersji dokumentu, który ma ulec zmianie /w szczególności może to być fragment pusty położony na styku dwóch kolejnych wierszy lub stron/, oraz nową treść tego fragmentu /w szczególności może ona być pusta/. Fragmenty te zadaje się za pomocą identyfikatorów stron i numerów wierszy. Należy tutaj dodać, że dla każdej strony wiersze stanowiące treść danej strony są ponumerowane kolejno od 1 wzwyż.

Ażeby uniknąć sortowania przyjęto zasadę, że fragmenty są zmieniane i zapisywane na TMO w takiej kolejności, w jakiej występują w starej wersji dokumentu /tj. na TM1/.

Ponadto, ponieważ operacje są zawsze wykonywane w kolejności ich wczytywania, konieczne jest, ażeby kolejność ta była zgodna z kolejnością rozmieszczenia wyznaczanych przez te operacje fragmentów w starej wersji dokumentu.

Wykonanie każdej operacji aktualizującej można podzielić na dwa etapy.

Etap 1 polega na przeniesieniu ze starej wersji dokumentu do nowej /tj. przepisaniu z TM1 na TMO/ części dokumentu zawartej pomiędzy fragmentem zmienianym przez daną operację a fragmentem zmienionym przez poprzednią operację.

Etap 2 polega na wstawieniu do nowej wersji dokumentu /tj. zapisaniu na TMO/ zmienianego fragmentu w zaktualizowanej postaci. Jednocześnie sporządzana jest dokumentacja zaktualizowanego fragmentu, uzupełnionego ewentualnie jeszcze do pełnej strony.

Ciąg operacji aktualizujących należy poprzedzić operacją

:: POPRAWIANIE

wykonanie tej operacji powoduje przygotowanie do procesu aktualizacji obydwu wykorzystywanych przez SMAD taśm.

Operacje aktualizujące można podzielić na dwie grupy: operacje odnoszące się do stron i operacje odnoszące się do wierszy.

Operacjami odnoszącymi się do stron są:

:: USUŃ STRONY : I-J
:: WSTAW PRZED STRONĄ : J
:: STRONA : J

gdzie I oraz J oznaczają identyfikatory stron.

Operacja "USUŃ STRONY" powoduje usunięcie z dokumentu kolejno stron od I do J, łącznie. Innymi słowy fragment starej wersji dokumentu złożony z ciągu stron zostaje w nowej wersji zastąpiony fragmentem pustym.

Operacja "WSTAW PRZED STRONĄ" pozwala na wstawienie przed stroną o identyfikatorze J ciągu stron zapisanego bezpośrednio za tą operacją.

Mamy tutaj zatem sytuację odwrotną niż w przypadku operacji "USUŃ STRONY". Pusty fragment starej wersji, będący stykiem strony J i strony ją poprzedzającej, zostaje w nowej wersji zastąpiony ciągiem stron. Ciąg strony zapisany po operacji "WSTAW PRZED STRONĄ" kończy się w miejscu wystąpienia kolejnej operacji aktualizującej. Operacja "STRONA" określa stronę o identyfikatorze J jako zmieniany fragment starej wersji dokumentu. Ponadto informuje ona SMAD, że zapisany po niej ciąg kolejnych odnoszących się do wierszy operacji wyznacza nową treść tej strony.

Operacjami odnoszącymi się do wierszy są:

:: SKREŚL WIERSZE : m - n
:: WYMIEN WIERSZE : m - n
:: DOPISZ PO WIERSZU : m

gdzie m oraz n są numerami wierszy.

Operacja "SKREŚL WIERSZE" powoduje usunięcie z określonej uprzednio strony tych wierszy, które w starej wersji miały numery od m do n.

Operacja "WYMIEN WIERSZE" powoduje wstawienie w nowej wersji dokumentu na uprzednio określonej stronie zamiast wierszy, które w starej wersji miały numery od m do n, ciągu wierszy zapisanego bezpośrednio za tą operacją. Podobnie jak koniec ciągu stron tak i koniec wstawianego ciągu wierszy wyznacza kolejna operacja aktualizująca.

Operacja "DOPISZ PO WIERSZU" powoduje wstawienie w nowej wersji dokumentu na określonej uprzednio stronie po wierszu, który w starej wersji miał numer m, ciągu wierszy zapisanego bezpośrednio po tej operacji.

Koniec procesu aktualizacji wyznacza operacja

:: KONIEC

Powoduje ona przeniesienie reszty dokumentu z TM1 na TMO oraz zakończenie pracy SMAD. Pomiedzy operacje aktualizujące można wstawiać również operacje: "ARKUSZ", "DRUKUJ" i "NIE DRUKUJ". Operacje te mają wtedy identyczne znaczenie jak przy wczytywaniu dokumentu.

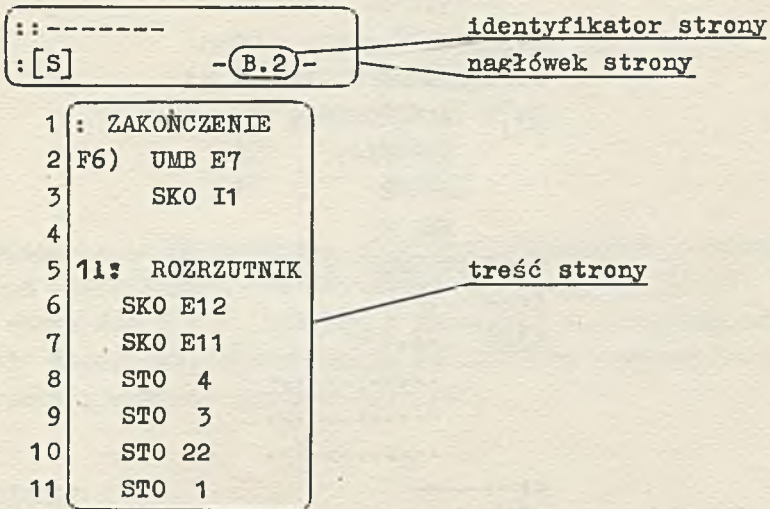
7. PRZYKŁAD AKTUALIZACJI DOKUMENTU

Celem zobrazowania sposobów korzystania ze SMAD do aktualizacji posłużmy się następującym przykładem.

Przypuśćmy, że na taśmie TM1 jest zapisany dokument złożony ze stron kolejno o identyfikatorach:

A.1, A.2, B.1, B.1.1, B.2, B.3, XD1, XD4

Niech przy tym strona B.2 ma postać jak na rys. 1.



Rys. 1

Treścią strony B.2 jest fragment programu w języku PJES. Liczby na początku każdego wiersza nie wchodzi w skład treści strony. Są to numery wierszy wypisywane jedynie przez SMAD w trakcie drukowania dokumentacji.

Nietrudno zauważyć, że wypisanie tej numeracji bardzo pomaga przy pisaniu operacji aktualizacji wierszy.

Przypuśćmy, dalej, że mamy następujący ciąg operacji:

```
::POPRAWIANIE
::ARKUSZ : 72
::DRUKUJ : 2
::USUŃ STRONY : A.2 - B.1.1
::STRONA : B.2
::WYMIENŃ WIERSZE : 1-2
```

F6) UEB E0
POB E15
SKO E4
SKO E4
:SKRESL WIERSZE : 9-11
::DOPISZ PO WIERSZU : 11
E4): ZAKONCZENIE
ODP E5+1
STO 2
SKO F3
::WSTAW PRZED STRONA : XD1
::-----
:[S] -B.4-
.....
.....
.....
::-----
:[S] -B.5-
.....
.....
.....
.....
::KONIEC

W wyniku wykonania przez SMAD tego ciągu operacji na TMO zostanie zapisana nowa wersja ww dokumentu, złożona kolejno ze stron o identyfikatorach: A.1, B.2, B.3, B.4, B.5, XD1, XD4.

Strona B.2 w nowej wersji będzie miała postać jak niżej

::-----
:[S] -B.2-
1 F6) UEB E0
2 POB E15
3 SKO E4
4 SKO E4
5 SKO I1

6	
7	I1): ROZRZUTNIK
8	SKO E12
9	SKO E11
10	STO 4
11	E4): ZAKOŃCZENIE
12	ODP E5 + 1
13	STO 2
14	SKO F3

Jednocześnie na drukarce wierszowej wydrukowana zostanie w dwóch egzemplarzach dokumentacja nowej wersji strony B.2 oraz stron B.4 i B.5. Zgodnie u wyrażonym w programie aktualizacji życzeniem, dokumentacja ta zostanie dopasowana do 72-wierszowego arkusza papieru drukarki.

8. WNIOSKI

SMAD wprowadzono na początku 1968 roku do prac nad oprogramowaniem maszyny ZAM 41, kiedy to rozpoczęto jego wstępną eksploatację. Zastąpił on stosowaną do tej pory tradycyjną metodę uruchamiania programów, operującą perforowaną taśmą papierową jako nośnikiem programów wraz z "nożyczkami i taśmą klejącą" jako narzędziami ich aktualizacji. Zalety SMAD w porównaniu z metodą tradycyjną wynikają z omówionych wyżej jego właściwości. Umożliwiły one skrócenie czasu trwania wszystkich trzech faz cyklu uruchamiania programów. Skrócenie pierwszej fazy wynika z zastąpienia powolnego wejścia z czytnika taśmy perforowanej szybkim wejściem z taśmy magnetycznej, a skrócenie trzeciej fazy ze zautomatyzowania procesu aktualizacji programów. Połączenie procesu aktualizacji z drukowaniem dokumentacji zmieniających fragmentów, konkretnie stron, aktualizowanego dokumentu zapewnia programiście posiadanie bez jakichkolwiek dodatkowych kosztów kompletnej i w pełni aktualnej dokumentacji programu. Możliwość dysponowania w każdej chwili taką dokumentacją ma istotny wpływ na sprawną i szybką lokalizację błędów w programie i w konsekwencji

proceedzi do skrócenia czasu potrzebnego na wykonanie czynności drugiej fazy cyklu uruchamiania programu. Nie oznacza to wszakże, by stosowanie SMAD dawało znaczne korzyści w każdym wypadku. Na ogół, nie opłaca się stosować SMAD do opracowywania programów o objętości poniżej kilkuset wierszy tekstu. Dodatkowe czynności, polegające na konieczności manipulowania taśmami magnetycznymi /zakładanie i zdejmowanie taśm/ oraz pisanie programu aktualizacji mogą w tych wypadkach raczej opóźnić niż przyspieszyć proces uruchamiania.

Należy również zwrócić uwagę, że nawet tam, gdzie skrócenie czasu uruchamiania jest oczywiste, nie musi ono oznaczać jednoczesnego zmniejszenia kosztów tego procesu. Zwróćmy bowiem uwagę na to, że przyspieszenie aktualizacji odbywa się zawsze kosztem stosunkowo drogiego czasu pracy maszyny. Ogólnie można stwierdzić, że największe korzyści przynosi stosowanie SMAD wszędzie tam, gdzie mamy do czynienia z programami obejmującymi od kilkuset wierszy do kilkunastu tysięcy wierszy tekstu, a jednocześnie takimi, że w wypadku stosowania tradycyjnej metody o czasie translacji decyduje przede wszystkim czas wczytywania programu. W maszynie ZAM 41 taką sytuację spotykamy z reguły w wypadku programów napisanych w językach typu assembler, na przykład PJES, PJEG czy SAS. Jak wykazała dotychczasowa eksploatacja SMAD w takich wypadkach zastosowanie tego systemu jest w pełni opłacalne. Najlepsze wyniki dało zastosowanie SMAD do opracowywania programów napisanych w językach PJES i PJEG. Czas aktualizacji w porównaniu z tradycyjną metodą został skrócony średnio czterokrotnie, a czas translacji w wyniku zastosowania wejścia z taśmy magnetycznej 5-7 razy. Oszczędności na czasie translacji zwracają przy tym zawiązką czas maszyny stracony na przeprowadzenie aktualizacji. Podobne wyniki otrzymano w przypadku programów w języku SAS.

W przyszłości przewidziane jest również zastosowanie SMAD do uruchamiania i magazynowania programów napisanych w językach: ALGOL, COBOL, EOL.

Literatura

- [1] WILKES M.K.: A Programmers Utility Filing System, Information Processing Machines, Czechoslovak Academy of Sciences, Praga 1965: 107-114.
- [2] Operating System GEORGE 1, 1900 Series ICT Computer Publications, First Edition August 1966:IV, 17-28.
- [3] ZAM 41 Oprogramowanie. Część pierwsza - Język Operacyjny Maszyny, Instytut Maszyn Matematycznych, Warszawa.

СИСТЕМА ХРАНЕНИЯ И АКТУАЛИЗАЦИИ ДОКУМЕНТОВ

Резюме

Описываемая в статье система хранения и актуализации документов (польское сокращение - SMAD) была прежде всего задумана как инструмент для наладки программ.

Программа, предназначенная для обработки в этой системе, должна быть разделена на страницы, в произвольно избранном месте обозначенном заголовком страниц. Заголовок является текстом формы, который в особенности содержит идентификатор страницы.

Так подготовленную для перфоленты или перфокарт программу переписывается с помощью операции SMAD "WSCZYTAJ" (перепиши) на магнитную ленту. Добавочным эффектом этой операции является копирование любого количества экземпляров документации записанной программы на построичной печатной машине. Форма этой документации учитывает как разделение на страницы так и размеры листа печатной машины, для того чтобы её можно было хранить в виде книжки и легко заместить неактуальные страницы - новыми. Кроме того, в документации составленной SMAD'ом перед каждой строкой программы записан её очередной номер, считан с начала страницы.

Программу, записанную на магнитной ленте можно поправить или подвергнуть трансляции. Во втором случае, благодаря применению входа из магнитной ленты, процесс трансляции проходит скорее чем в случае входа из перфоленты или перфокарт. Для актуализации программы нужны две магнитные ленты, одна из которых содержит старую версию программы, а другая предназначена для новой версии. Внесение поправок производится при помощи соответствующих операций SMAD, например "USUŃ STRONY" (исключи страницы), "WSTAW STRONY" (включи страницы), "SKRĘŚL WIERVSZE" (вычеркни строки), "WYMIENŃ WIERVSZE" (замени строки), "DOPISZ PO WIERVSZU" (допиши после строки). Изменяемые отрезки программы определяются посредством идентификаторов страниц и номеров строк.

Процесс исправления связан с печатанием документации страниц, подвергнутых изменению. Это обеспечивает потребителю SMAD актуальную документацию обрабатываемой программы.

SMAD был внедрён во вычислительную машину ZAM-41 в половине 1968 года. Его эксплуатация оказалась до сих пор вполне полезная для наладки программ. Наибольшие результаты по отношению к ускорению процесса наладки были достигнуты для программ написанных на языках типа ассамблер: PJEK и SAS.

THE SYSTEM OF DOCUMENT STORING AND UPDATING

Summary

The described System of document storing and updating /Polish abbreviation SMAD/ first of all was thought as a tool for program debugging.

The program destined to operate in this system must be first divided into pages. The place of division can be freely chosen and marked by the so-called page headings. The heading is a text of a determined form, which in particular comprises the page identifier.

A program so prepared on a paper tape or punch cards is rewritten on a magnetic tape by means of the SMAD operation "WCZYTAJ" /rewrite/.

An additional effect of this operation is the printing of the rewritten program on a line-printer in a required number of copies. This documentation form considers both; division into pages and the size of the printer sheet so as to be stored in the form of a book and permit to replace easily the outdated pages by new ones. Moreover, every program line of SMAD documentation is provided with its succeeding number, counting from the beginning of the page.

The program written on a magnetic tape can be corrected or translated. In the latter case, due to the application of input from a magnetic tape, the translation is shorter than in the case of input from a paper tape or punch cards. For program updating two magnetic tapes are needed; one - containing the old program version, the other being destined for the new version. Corrections are made by means of adequate operations SMAD, e.g.: "USUŃ STRONY" /remove pages/, "WSTAW STRONY" /put in pages/, "SKREŚL WIERSZ" /cancel line/, "WYMIEN WIERSZE" /exchange lines/, "DOPISZ PO WIERSZU" /add after line/. The changed parts of the program are determined by means page identifiers and line numbers.

The process of correction is connected with changed page printing. This ensures the SMAD user an updated documentation of the processed program.

SMAD was implemented in a ZAM 41 computer in the middle of 1968. Its exploitation proves full usefulness for program debugging. As far as a speed up of the process of debugging programs written in languages of the type of assembler PJEG and SAS are concerned, the results are up till now the best.

OPROGRAMOWANIE UNIWERSALNE
MASZYNY ZAM 41 DO EPD

Hanna MYSIOR

Pracę złożono 21.05.1969

W artykule omówiono pewne elementy oprogramowania uniwersalnego maszyny ZAM 41 dla potrzeb elektronicznego przetwarzania danych. W szczególności przedstawiono zasady działania uniwersalnych systemów stanowiących zespół powiązanych ze sobą programów umożliwiających kompletną lub częściową realizację konkretnego problemu z danej klasy zastosowań, przy czym adaptacja systemu następuje w oparciu o informacje sterujące. Są to tzw. systemy parametryzowane.

1. WSTĘP

Cechą charakterystyczną stosowania elektronicznej techniki obliczeniowej jest z jednej strony duża szybkość uzyskiwania potrzebnych informacji, z drugiej zaś strony długi okres przygotowań, wynikający m.in. z dużej pracochłonności programowania. Szczególnie daje się to zauważyć w systemach elektronicznego przetwarzania danych dla potrzeb zarządzania, w których z reguły mamy do czynienia nie z pojedynczymi programami, lecz z systemami programów opracowywanymi każdorazowo dla konkretnego zagadnienia. Można jednak zauważyć, że systemy te posiadają wiele wspólnych cech, co nasuwa myśl pewnej ich uniwersalizacji. Z tego też względu obserwuje się za granicą dążenie do tworzenia oprogramowania dla poszczególnych modeli maszyn cyfrowych obejmującego uniwersalne elementy zastosowań. Elementy te mogą być w prosty sposób adaptowane do konkretnych

potrzeb, dzięki czemu wydatnie skracaa się czas oprogramowania tych zastosowań.

Zaletą oprogramowania uniwersalnego jest przede wszystkim skrócenie czasu opracowywania systemów oraz praktyczne umożliwienie dokonywania zmian organizacyjnych w eksploatowanych systemach, bez konieczności ponownego zaprogramowania całości systemu.

Jak większość rozwiązań ogólnych, oprogramowanie uniwersalne ma pewne cechy ujemne, wynikające z tego, że nie można nim objąć wszystkich potrzeb realizowanych systemów. Ponadto uniwersalne rozwiązania zmniejszają efektywność opartych na nich systemów.

Zaspokojenie wszystkich potrzeb realizowanych systemów uzyskuje się poprzez wykonanie dodatkowych programów dla konkretnych zastosowań, zaś zmniejszenie efektywności systemów ma coraz mniejsze znaczenie, wobec pojawiania się coraz szybszych maszyn cyfrowych.

Biorąc pod uwagę fakt, że we wszystkich niemal systemach przetwarzania danych występują takie czynności jak: czytanie danych wejściowych, kontrola danych i wstępne ich opracowywanie, redakcja wydawnictwa itp., Zakład Przetwarzania Danych Instytutu Maszyn Matematycznych postanowił rozpocząć pracę nad opracowaniem dla maszyny cyfrowej ZAM 41 uniwersalnego oprogramowania w zakresie wyżej wymienionych czynności.

Na obecnym etapie znajomości omawianego problemu oprogramowania uniwersalnego, wymienione czynności są stosunkowo proste do sformułowania. Bardziej złożone czynności, które obecnie trudno jest nawet sformułować, jak np. aktualizacja ewidencji, zostały odłożone do czasu lepszego rozeznania w tym zakresie.

Realizowane zbiory programów nazwano Systemem Ogólnych Programów Użytkowych /OPUS/ i Systemem ANKIetowania /SANKI/.

2. KLASYFIKACJA OPROGRAMOWANIA UNIWERSALNEGO

Ze względu na sposób korzystania, oprogramowanie uniwersalne można podzielić na:

- podprogramy
- programy samodzielne
- systemy programów.

Podprogramy są to wydzielone fragmenty programowe realizujące najbardziej elementarne czynności. Przykładami takich czynności w elektronicznym przetwarzaniu danych mogą być: czytanie dokumentu, eliminacja zbędnych znaków, przeliczanie liczb, działanie na tekstach, sortowanie, itp. Korzystanie z podprogramu wymaga opracowania tzw. programu głównego, który realizując pewne czynności będzie się odwoływał do tego podprogramu.

Bardziej złożone czynności maszynowe są realizowane przez programy. Wśród nich można wyróżnić: programy specjalizowane, ściśle dostosowane do realizacji konkretnych czynności oraz programy uniwersalne o charakterze ogólniejszym, realizujące konkretne czynności na podstawie danych podawanych z zewnątrz. Korzystanie z programu uniwersalnego wymaga przekazania mu z zewnątrz informacji, które będą sterowały jego działaniem. Informacje te na ogół podawane są do programu w postaci kart przewodnich, z monitora itp. Z tego względu programy te są często nazywane programami parametryzowanymi. Przykładem takiego programu może być program wczytywania danych wejściowych z kontrolą, wstępnym opracowaniem i zapisaniem na taśmie magnetycznej. Parametrami podawanymi z zewnątrz będą: opis postaci danych, opis algorytmu kontroli i wstępnego opracowania oraz opis postaci zapisu na taśmie magnetycznej.

Uniwersalne systemy stanowią zespół powiązanych ze sobą programów, umożliwiających kompletną lub częściową realizację konkretnego problemu z pewnej klasy zastosowań /np. gospodarka materiałowa, opracowanie zbioru ankiet itp./. Adaptacja systemu następuje podobnie jak przy programach uni-

wersalnych w oparciu o informacje sterujące. W dalszej części artykułu będziemy się zajmować tylko elementami oprogramowania zrealizowanymi przez Zakład Przetwarzania Danych.

3. PODPROGRAMY

Ponieważ postanowiono programy wyżej wymienionych systemów opracować w języku PJEK, wymagało to korzystania z uniwersalnych podprogramów w tym języku. Biblioteka Oprogramowania w języku PJEK nie zawierała wszystkich podprogramów potrzebnych do realizacji zaprojektowanych systemów EPD. Wynikła zatem potrzeba opracowania wielu uniwersalnych podprogramów, z których podprogramy współpracy z taśmami magnetycznymi oraz oparty na tej współpracy podprogram sortowania zostaną w dalszym ciągu opisane.

3.1. System współpracy z taśmami magnetycznymi

System TM3 /SZMATA - System Zapisu Magnetycznych TAŚM/ organizuje współpracę z taśmami magnetycznymi zapisanymi wg zasad przyjętego w Instytucie Maszyn Matematycznych standardu.

Najważniejsze cechy tego systemu:

- system składa się z następujących 4 podprogramów:
 - podprogram pisania pliku na taśmie magnetycznej
 - podprogram rozszerzania pliku z taśmy magnetycznej
 - podprogram odczytania pliku z taśmy magnetycznej
 - podprogram zakończenia współpracy z taśmą magnetyczną
- podprogramy mają po kilka różnych wejść związanych z realizacją poszczególnych etapów czynności /np. przygotuj wykonanie, wykonaj, zakończ wykonywanie/
- informacje o budowie pliku na taśmie magnetycznej przekazywane są do podprogramów za pomocą specjalnej tablicy

- informacje z podprogramów do programu głównego /np. o błędach/ są również przekazywane za pomocą ww tablicy. Dodatkowo informacje wypisywane są na monitorze.

Ponadto system TM3 umożliwia współpracę ze zbiorami wieloplikowymi oraz plikami wielospulowymi.

W przypadku zbiorów jednoplikowych umożliwia dopisywanie danych na końcu pliku /rozszerzanie pliku/.

3.2. Podprogram sortowania

Podprogram ST3 służy do sortowania lub łączenia dokumentów na taśmach magnetycznych. Jego realizacja jest wzorowana na programie sortowania dla ICT 1900.

Najważniejszymi cechami tego podprogramu są:

- sortowanie i łączenie wykonuje się wg następujących rodzajów kluczy: znaki alfanumeryczne zapisane w kodzie wewnętrznym sześćo- i ośmiobitowym, liczby stałoprzecinkowe lub liczby zmiennoprzecinkowe
- sortowanie i łączenie można wykonywać wg kluczy rosnących lub malejących.
- klucz sortowania może praktycznie być dowolnie długi, a jego długość jest liczona w jednostkach odpowiednich dla rodzaju klucza /tzn. znakach lub słowach maszyny/
- podprogram umożliwia sortowanie lub łączenie wg dowolnej liczby kluczy
- podprogram dopuszcza sortowanie danych zapisanych na taśmie magnetycznej lub podawanych przez program główny. Podobnie posortowane dokumenty mogą być zapisane na taśmie magnetycznej lub podawane do programu głównego
- podprogram umożliwia opracowywanie dokumentów o równych kluczach

- informacje o budowie danych do sortowania i o sposobie sortowania podaje się do podprogramu za pomocą specjalnej tablicy
- informacje o przebiegu sortowania /m.in. o przybliżonej liczbie przebiegów/ wypisywane są na monitorze.

Podprogram wymaga do pracy co najmniej 4 jednostek taśmy magnetycznej i monitora. Jego współpraca z taśmami magnetycznymi oparta jest na systemie TM3 /SZMATA/.

4. OPUS - SYSTEM OGÓLNYCH PROGRAMÓW UŻYTKOWYCH

4.1. Rodzaje programów

Z punktu widzenia realizowanych czynności programy systemu OPUS można podzielić na następujące grupy:

- programy testowania danych
- programy testowania danych z zapisaniem na taśmie magnetycznej
- program opracowywania danych zapisanych na taśmie magnetycznej
- program redagowania wydawnictw
- program sortowania
- programy pomocnicze

Programy testowania danych wczytują dane zapisane na taśmie papierowej lub na kartach dziurkowanych, sprawdzają ich formalną poprawność i wypisują dane błędne w tzw. wykazie błędów.

Programy testowania danych z zapisaniem na taśmie magnetycznej oprócz wykonywania czynności programów testowania tworzą z danych poprawnych bloki o określonej postaci i zapisują je na taśmie magnetycznej.

Program opracowywania danych modyfikuje dla konkretnych potrzeb dane zapisane na jednej taśmie magnetycznej i w nowej postaci zapisuje je na innej taśmie magnetycznej.

Program redagowania wydawnictw wypisuje na określonym urządzeniu wyjściowym wydawnictwa, korzystając z bloków informacyjnych zapisanych na taśmie magnetycznej.

Program sortowania służy do sortowania lub łączenia dokumentów zapisanych na taśmie magnetycznej.

Programy pomocnicze służą do metrykowania, kontroli, powielania i wypisywania zawartości taśm magnetycznych.

Programy testowania danych i sortowania, jako najbardziej interesujące i najbardziej rozbudowane, zostaną w dalszym ciągu artykułu opisane bardziej szczegółowo.

4.2. Ogólne zasady działania

Działanie programów systemu OPUS składa się z:

- wczytania informacji sterujących zapisanych na tzw. kartach przewodnich /karty te mogą być przygotowywane na taśmie papierowej lub kartach perforowanych/
- opracowania danych opierając się na interpretacji informacji sterujących.

Jednolity sposób podawania informacji sterujących przyjęto tylko dla współpracy z taśmami magnetycznymi, pozostałe informacje sterujące dla poszczególnych typów programów mają różną formę i treść.

Do podawania informacji dla współpracy z taśmą magnetyczną służy odrębna karta przewodnia, tzw. karta opisu pliku. Opisuje ona budowę pliku zapisywanego lub odczytywanego z taśmy magnetycznej oraz numery jednostek taśm magnetycznych, z którymi program współpracuje.

Współpraca z taśmami magnetycznymi w systemach OPUS i SAN-KI oparta jest na systemie TM3.

W większości programów ww systemów ograniczono się do zbiorów jednoplukowych.

Karty przewodnie zawierające informacje sterujące dla poszczególnych typów programów mają następującą treść:

- dla programów testowania podają dokładny opis budowy danych /długość pozycji, liczba elementów w pozycji itp./, sposób ich identyfikacji oraz sposób postępowania w przypadku pozycji poprawnej lub błędnej
- dla programu opracowywania danych, sposób ich opracowania /np. łączenie lub pomijanie danych/ oraz opis budowy danych po opracowaniu
- dla programu wydawniczego podają teksty nagłówkowe dla pisanych wydawnictw oraz sposób ich redagowania /liczba wierszy na stronie, liczba kolumn, sposób sumowania itp./
- dla programu sortowania podają parametry sortowania /numery taśm magnetycznych: wejściowej, wyjściowej i roboczych, nazwy pliku wejściowego i wyjściowego, opisy kolejnych kluczy sortowania itp./.

Na podstawie kart przewodnich tworzone są w pamięci maszyny bloki informacji sterujące działaniem programu.

Projektując system OPUS założono, że do wykonania przez poszczególne programy prostych czynności kart przewodnich powinno być niewiele, a wypełnianie ich powinno być bardzo proste. Bardziej skomplikowane czynności mogą wymagać większej liczby kart o bardziej złożonej strukturze.

4.3. Programy wozytywania danych

Programy wozytywania danych obejmują:

- programy testowania danych

- programy testowania danych z zapisaniem na taśmie magnetycznej.

Dane wejściowe mogą być zapisane na kartach dziurkowanych lub na taśmie papierowej.

Programy testowania danych zapisanych na kartach pozwalają testować zarówno karty 80-kolumnowe jak i 90-kolumnowe.

Zakłada się następującą budowę danych wejściowych:

- dane składają się z pozycji
- pozycje składają się z elementów
- elementy składają się ze znaków, które odpowiadają informacjom zapisanym w pojedynczych kolumnach karty.

Dopuszcza się dwa typy pozycji:

- pozycje jednokartowe, w których informacje mieszczą się na jednej karcie
- pozycje wielokartowe, w których informacje zapisane są na kilku kolejnych kartach.

Jednocześnie można testować kilka rodzajów pozycji jednokartowych lub jeden rodzaj pozycji wielokartowej.

Pozycje jednokartowe mogą być identyfikowane przez podany identyfikator lub przez identyfikator poprzedniej pozycji.

Dopuszcza się też jeden rodzaj pozycji bez identyfikatora.

Pozycje wielokartowe muszą mieć identyfikator pozycji oraz identyfikatory poszczególnych kart w pozycji.

Pozycje formalnie błędne są wypisywane w WYKAZIE BŁĘDÓW z podaniem lokalizacji błędu. Programy testujące dane i zapisujące je na taśmie magnetycznej zapisują tylko dane formalnie poprawne.

Programy testowania danych zapisanych na taśmie papierowej pozwalają testować dane zapisane w dowolnym kodzie.

Zakłada się następującą budowę danych wejściowych:

- dane wejściowe składają się z wierszy rozdzielanych separatorami wiersza
- wiersze składają się z elementów rozdzielonych separatorami elementów
- elementy wiersza składają się z podelementów.

Jednocześnie można testować kilka różnych rodzajów pozycji. Pozycje mogą być identyfikowane przez podany identyfikator, przez identyfikator poprzedniej pozycji lub przez liczbę elementów.

Podobnie jak w przypadku kart, pozycje błędne są wypisywane w WYKAZIE BŁĘDÓW z podaniem lokalizacji błędu, a w przypadku programów z zapisywaniem na taśmie magnetycznej pozycje poprawne zapisywane są na wybranej taśmie.

4.4. Program sortowania

Program sortowania wykonuje na 4 taśmach magnetycznych sortowanie dokumentów zapisanych na taśmie magnetycznej wg standardu IMM. Program korzysta ze zmodyfikowanego podprogramu sortowania ST3.

Dokumenty wejściowe mogą być stałej lub zmiennej długości. Wymagane jest aby:

- każdy dokument zawierał wszystkie klucze sortowania,
- każdy klucz miał stały adres względem początku dokumentu we wszystkich dokumentach,
- każdy klucz miał stałą długość w każdym dokumencie,
- każdy klucz był w całości w jednym miejscu,
- długość klucza była nie większa niż 1023,
- liczba kluczy sortowania była nie większa od 400.

Przy wczytywaniu parametrów sortowania sprawdzana jest ich formalna poprawność i błędy sygnalizowane są wypisaniem odpowiedniego tekstu.

5. SANKI - SYSTEM ANKIETOWANIA

System SANKI /System ANKIetowania/ jest przystosowany do opracowywania określonego typu ankiet i uzyskiwania z góry założonych wydawnictw. Danymi wejściowymi dla tego systemu są wypełnione ankiety. Jedna ankietę może zawierać wiele różnych pytań, na każde pytanie mogą być udzielane różne odpowiedzi. Nie dopuszcza się możliwości zadawania wielu jednakowych pytań oraz podawania własnych uwag lub wyjaśnień.

Programy systemu SANKI funkcjonalnie można podzielić na dwie grupy:

- programy zapisujące informacje z ankiety na taśmie magnetycznej
- programy wydawnicze.

Pierwsza grupa programów, korzystając z informacji zawartych w karcie przewodniej, zapisuje informacje z ankiety na wybranej taśmie magnetycznej.

Druga grupa programów, korzystając z tej taśmy magnetycznej, wypisuje wydawnictwa na dowolnym urządzeniu wyjściowym.

Przewidywane wydawnictwa obejmują zbiorcze zestawienie ankiet oraz tablice współzależności odpowiedzi na wybrane pytania.

Pierwsze z wydawnictw zawiera informacje o ilości różnych odpowiedzi na wszystkie pytania, daje więc jakby obraz całej ankiety, bez podawania związków między odpowiedziami.

Drugie wydawnictwo podaje współzależność odpowiedzi na dwa dowolnie wybrane pytania.

Obydwa wydawnictwa podają liczby ankiet z danym rodzajem odpowiedzi oraz procent w stosunku do różnych określonych parametrów.

6. ZAKOŃCZENIE

Opracowane w Zakładzie Przetwarzania Danych IMM systemy uniwersalnych programów są pierwszymi tego typu systemami w Polsce.

Interesujące jest więc zebranie wniosków z ich realizacji i eksploatacji. Początkowo założono, że okres realizacji wyniesie 1 rok /łącznie z projektem koncepcyjnym/. Nad systemami pracowało 8 osób. Wobec braku doświadczeń krajowych w zakresie wyżej omawianych zagadnień przyjęto interpretacyjną metodę realizacji. Metoda ta okazała się bardzo pracochłonna i wymagająca dużej liczby miejsc pamięci. Związane z tym trudności spowodowały, że systemy opracowywano ok. 1,5 roku.

W wyniku pracy otrzymano system OPUS /14 programów - około 30 tys. rozkazów/ i system SANKI /5 programów - około 10 tys. rozkazów/.

Większość programów jest już dobrze sprawdzona w eksploatacji. Wobec tego, że niektóre czynności programów /znacznie jednak szersze niż funkcje podprogramów/ powinny być zaprogramowane uniwersalnie, wydaje się, że następnym etapem pracy nad oprogramowaniem uniwersalnym do EPD powinno być opracowanie podobnych systemów metodą generacyjną.

УНИВЕРСАЛЬНЫЕ СРЕДСТВА ПРОГРАММИРОВАНИЯ МАШИНЫ ZAM-41 ДЛЯ ОБРАБОТКИ ДАННЫХ

Резюме

Большая трудоёмкость программирования систем электронной обработки данных заставила творцов средств программирования цифровых вычислительных машин создать средства, заключающие универсальные элементы для применений.

Эти элементы можно адаптировать простым способом к конкретным нуждам, благодаря чему время опrogramмирования этих применений значительно сокращается.

В Отделе Обработки Данных Института Математических Машин разработан ряд элементов универсального опrogramмирования, выполняющих основные функции в области обработки данных.

Кроме подпрограмм организующих совместную работу с магнитными лентами, записанными согласно принятому в ИММ стандарту и сортировки на магнитных лентах, были разработаны две системы параметризованных программ: OPUS и BANKI.

Программы входящие в состав этих систем действуют на основе управляющей информации поданной снаружи.

Программы системов OPUS выполняют, между другими, следующие функции:

- проверяют данные записанные на любом носителе информации,
- разрабатывают данные записанные на магнитной ленте,
- размещают в печатной машине издательства приготовленные заранее на магнитной ленте,
- делают сортировку документов записанных на магнитной ленте.

Система BANKI приспособлена к разработке определённого типа анкет и получения издательств положенных заранее.

Обе системы были разработаны 8 людьми в течении 1,5 года, причём получено 19 программ, включающих около 40 тысяч команд.

ZAM 41 COMPUTER UNIVERSAL SOFTWARE FOR DATA PROCESSING

Summary

Great labouriousness of data processing system programming constrained digital computer software builders to find a solution that would embrace universal elements for the purpose of application.

These elements may be adapted to concrete needs, and due to this the time of software developing gets significantly shorter.

A series of universal software elements, realizing basic functions in the field of data processing, was elaborated in the Department of Data Processing of the Institute of Mathematical Machines.

Apart of subroutines that organize the cooperation with magnetic tapes recorded according to the standard accepted in IMM, and sorting on magnetic tapes, two program systems OPUS and SANKI were developed, operating according to parameters given from outside.

The programs of the OPUS system realize, among others, the following functions:

- testing data written on any information carrier,
- processing data written on a magnetic tape,
- editing publications prepared on a magnetic tape formerly,
- sorting documents written on a magnetic tape.

The SANKI system is adapted to elaborate inquiries of a determined type, and to obtain editions assumed in advance.

Both systems were developed by 8 persons during 1 year and a half. 19 programs, comprising about 40 thousand instructions were obtained.

KONSTRUKCJA I TECHNOLOGIA

PRÓBA ZASTOSOWANIA METODY BITTERA DO OCENY
JAKOŚCI ZAPISU MAGNETYCZNEGO NA WARSTWACH
REJESTRUJĄCYCH PAMIĘCI BĘBNOWYCH

Hanna KUŹNICKA

Pracę złożono 9.08.1969

W pracy przedstawiono wyniki próby zastosowania koloidalnej metody Bittera do oceny zapisu magnetycznego na różnych rodzajach warstw rejestrujących. Badania przeprowadzono na warstwach tlenkowych, typowych dla pamięci bębnowych, na warstwie metalicznej Ni-Co oraz na warstwie rejestrującej taśmy cyfrowej Pyral HP 2N 12 N2.

1. WSTĘP

W pracy niniejszej przedstawiono wyniki badań nad zastosowaniem metody Bittera [1] zwanej również metodą koloidalną [2] do kontroli współpracy pola magnetycznego głowicy zapisującej z warstwą rejestrującą pamięci bębnowej. Metoda ta pozwala ujawnić nieciągłości pola magnetycznego warstwy rejestrującej i ocenić gęstość przemagnesowań. Metoda koloidalna była wcześniej stosowana do obserwacji domen, ruchów ścian międzydomenowych, defektów budowy kryształów magnetycznych itp. [1 ÷ 5]. Polega ona na ujawnianiu nieciągłości pola magnetycznego na powierzchni ferro- lub ferrimagnetyka. Czynnikiem ujawniającym zapis magnetyczny na warstwie rejestrującej jest proszek ferromagnetyczny np. magnetyt lub maghemit tworzący zawiesinę w odpowiedniej cieczy. Zmiany natężenia

pola magnetycznego powodują układanie się ferromagnetycznych cząstek zawieszony w figury odwzorowujące niejednorodności pola. Siła działająca na cząstkę ferromagnetyczną jest proporcjonalna do momentu magnetycznego cząstki i zmian pola.

Zgodnie z pracami Kittela [4] o powstawaniu skupisk jednodomenowych cząstek, których energia wynika z funkcji rozkładu Boltzmann [2], [4], stosunek gęstości $n(H, \theta)$ cząstek, których momenty magnetyczne są skierowane pod kątem θ do kierunku pola o natężeniu H , do gęstości cząstek $n(0)$, gdzie natężenie pola jest równe 0 wyraża się zależnością [2]:

$$\frac{n(H, \theta)}{n(0)} = \exp \left[\frac{\mu H \cos \theta}{kT} \right] \quad /1/$$

gdzie:

- μ - moment magnetyczny cząstek
- k - stała Boltzmann
- T - temperatura w skali Kelvina

Całkując $n(H, \theta)$ względem θ w granicach $0 \div \pi$ uzyskuje się wyrażenie:

$$\frac{n(H)}{n(0)} = \frac{1}{4\pi} \int_0^\pi \exp \left[\frac{\mu H \cos \theta}{kT} \right] 2\pi \sin \theta \, d\theta = \frac{\sinh x}{x} \quad /2/$$

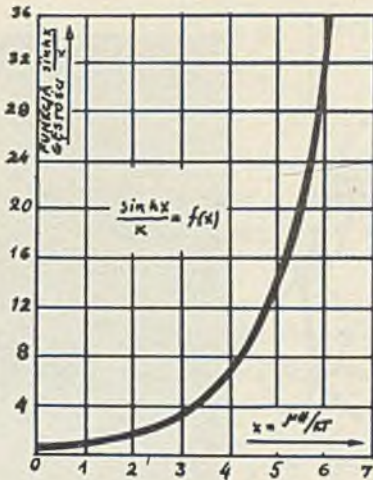
gdzie:

$$x = \frac{\mu H}{kT} \quad /3/$$

Zależność /2/ w funkcji x przedstawiona na rys. 1 pokazuje, że gęstość cząstek szybko rośnie w obszarach gdzie $x > 3$, to jest tam, gdzie:

$$\mu H > 3 kT \quad /4/$$

Stąd, w przypadku cząstek jednodomenowych, dla lokalnych zmian natężenia pola H , warunkiem powstawania figur jest spełnienie zależności /4/.



Rys. 1. Wykres gęstości cząstek $\frac{\sinh x}{x}$ w funkcji x /według Kittela [4]/

Jednodomenowe cząstki w zawieszynie podlegają działaniu własnych pól pojedynczych cząstek oraz pola badanego obszaru powierzchni warstwy.

Pola własne cząstek powodują, że cząsteczki skupiają się w pierścienie lub zwarte skupiska, w których strumień magnetyczny tworzy linię zamkniętą /rys. 2/.

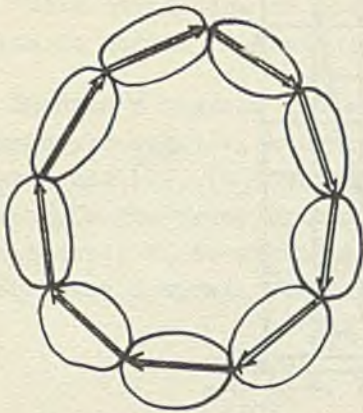
Skupiska o zamkniętym strumieniu magnetycznym, nie uczestnicząc w procesie tworzenia figur, utrudniają uzyskanie czytelnych obrazów odwzorowujących niejednorodności pola magnetycznego badanej powierzchni.

Moment magnetyczny cząstki o średnicy d i magnetyzacji I_s wynosi:

$$\mu = \frac{\pi d^3 I_s}{6} \quad /5/$$

Ponieważ pole między dwiema cząsteczkami jest wyrażone jako:

$$H = \frac{2 \mu}{d^3} \quad /6/$$



a.



b.

Rys. 2. Pierścień o zamkniętym strumieniu magnetycznym złożony z cząstek jednodomenowych

/a/ - schemat, /b/ - mikrofotografia /proszek $\gamma\text{-Fe}_2\text{O}_3$ w powiększeniu 3000 razy/

można z zależności /4/, po podstawieniu /5/ i /6/, wyznaczyć warunki powstawania aglomeratów o zamkniętym strumieniu magnetycznym, ograniczające jednocześnie maksymalną wielkość cząstek tworzących figury:

$$\frac{2 \mu^2}{d^3} > 3 kT$$

$$\frac{d^3 \pi^2 I_s^2}{18} > 3 kT$$

$$d > 3 \left[\frac{2 kT}{\pi^2 I_s^2} \right]^{1/3}$$

171

Wprowadzając pojęcie cząstek uczestniczących efektywnie w tworzeniu figur proszkowych i oznaczając:

$$d_e = 3 \left[\frac{2 kT}{\pi^2 I_s^2} \right]^{1/3} \quad /8/$$

można stwierdzić, że cząstki o średnicy większej niż d_e będą tworzyć pierścienie i aglomeraty o zamkniętym strumieniu magnetycznym i tylko cząstki o średnicy $d < d_e$ będą efektywne w tworzeniu figur.

Dolną granicę średnic cząstek zdolnych do wytworzenia figur wyznaczają warunki /4/ i /5/, czyli

$$\frac{\pi d^3 I_s H}{6} > 3 kT \quad /9/$$

gdzie H określa wielkość natężenia pola magnetycznego na powierzchni próbki.

Minimalna średnica cząstek tworzących figury proszkowe zależy od wartości lokalnych pól magnetycznych na badanej powierzchni oraz od magnetyzacji I_s cząstek w koloidzie.

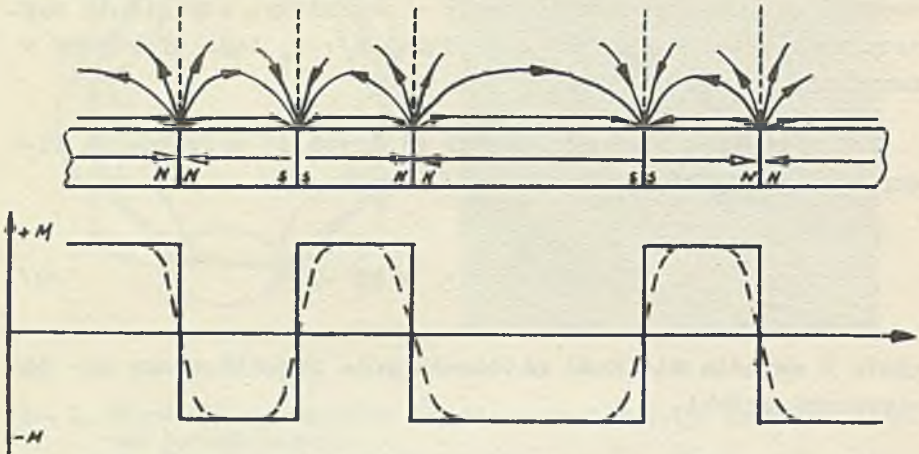
Średnice cząstek stosowanych w zawiesinach powinny więc zawierać się w granicach określonych warunkami /8/ i /9/. Spełnienie tych warunków umożliwia uzyskanie czytelnych figur odwzorowujących nieciągłości pola magnetycznego badanej powierzchni. Metoda koloidalna pozwala badać zmiany magnetyzacji powierzchni warstw rejestrujących poddanych zapisowi.

Schemat rozkładu pola magnetycznego takiej warstwy przedstawia rys. 3.

Siły oddziaływujące na proszek zawieszony w cieczy pokrywającej badaną powierzchnię warstwy są proporcjonalne do zmian pola na tej powierzchni.

Proszek gromadzi się w obszarach przemagnesowań, charakteryzujących się znacznie większą wartością strumienia rozproszenia niż pozostałe obszary warstwy [5], [6]. Obszary prze-

magnesowań mają pewną szerokość, ponieważ zmiana kierunku namagnesowania nie odbywa się w sposób skokowy. Szerokość "paszków" odwzorowujących zmiany namagnesowania zależy od właściwości materiału proszkowego użytego jako "wywoływacz" jak i od właściwości badanej warstwy.



Rys. 3. Schemat rozkładu pola magnetycznego warstwy poddanej zapisowi magnetycznemu

2. CZĘŚĆ DOŚWIADCZALNA

Przedstawione w pracy badania obejmowały:

- obserwację i pomiar gęstości zapisu binarnego w warstwie
- opracowanie łatwej metody ustalenia prawidłowości ustawienia szczeliny głowicy nad warstwą rejestrującą

Badaniom poddano warstwy rejestrujące z $\gamma\text{-Fe}_2\text{O}_3$ na powierzchni bębnow i metaliczne warstwy magnetyczne na powierzchni modelu dysku.

Uzyskane wyniki porównywano z zapisem dokonany w analogicznych warunkach na taśmie cyfrowej PYRAL.

Próby przeprowadzono na kilku rodzajach warstw magnetycznych:

- warstwa z $\gamma\text{-Fe}_2\text{O}_3$ z żywicą epoksydową jako spoiwem /WM - 80/
- warstwa z $\gamma\text{-Fe}_2\text{O}_3$ z żywicą poliwinyllobutyralową jako spoiwem /WM - 88/
- warstwa metaliczna Ni-Co
- taśma cyfrowa PYRAL HP 2N 12 N2

Do wytworzenia zawiesiny stosowano gamma tlenek żelazawy:

- $\gamma\text{-Fe}_2\text{O}_3$ z rozkładu mrówczanu żelazawego
- $\gamma\text{-Fe}_2\text{O}_3$ H-100 firmy EMI
- $\gamma\text{-Fe}_2\text{O}_3$ Szost. 6 - produkcji Szostkińskich Zakładów Chemicznych /ZSRR/
- $\gamma\text{-Fe}_2\text{O}_3$ Szost. 6 - impregnowane monomolekularną warstwą żywicy epoksydowej.

Jako ośrodki dyspersyjne stosowano:

- alkohol etylowy
- heksan
- mieszaninę heksanu i dekanolu
- roztwór kwasu laurylowego w heksanie.

Ciecze te dobierano tak, aby nie wykazywały niszczącego działania na strukturę badanych warstw magnetycznych i charakteryzowały się dużą lotnością.

Proszki ferromagnetyczne były rozpraszane w ośrodku dyspersyjnym przez ucieranie w móżdzierzu agatowym lub mielenie w młynie kulowym, a następnie sączone przez filtr o średnicy oczka 50 μm . Ze względu na tendencję ziaren proszku do aglomeracji, zawiesiny do wywoływania figur stosowano bezpośrednio po ich przygotowaniu. Figury proszkowe otrzymywano przez natryskiwanie badanej powierzchni warstwy magnetycznej zawiesiną. W celu uzyskania wyraźniejszych figur proszkowych stosowano przemycanie warstwy cieczą stanowiącą ośrodek dyspersyjny zawiesiny. Otrzymane figury proszkowe obserwowano w powiększeniu 25x i 75x.

Zróźnicowane właściwości fizyczne i magnetyczne badanych warstw rejestrujących powodowały konieczność doboru składników zawiesiny do uzyskania czytelnego odwzorowania dokonanego zapisu, oraz wyeliminowania składników niszczących badaną warstwę.

Warstwa WM-80 - typowa dla pamięci bębnowych, dopuszcza stosowanie wszystkich podanych wyżej proszków i ośrodków dyspersyjnych. Optymalne wyniki uzyskano przy zastosowaniu zawiesiny γ - Fe_2O_3 H - 100 EMI w alkoholu etylowym. Figury proszkowe odwzorowujące obszary przemagnesowań były bardzo wyraźne i pozwalały się mierzyć. Szerokość obszarów przemagnesowań na warstwie o grubości $10 \mu\text{m}$ rejestrowanych z gęstością 9 b/mm w systemie NRZ wynosiła $35 \pm 5 \mu\text{m}$.

Przy zastosowaniu zawiesiny tlenków γ - Fe_2O_3 Szost. 6 w roztworze kwasu laurylowego w heksanie, szerokość obszaru przemagnesowania zawlewała się w granicach $80 - 150 \mu\text{m}$ w zależności od stężenia kwasu laurylowego.

Rozpraszenie proszków ferromagnetycznych w roztworze kwasu laurylowego w heksanie za pomocą młynka kulowego uznano za niecelowe. Wytworzone zawiesiny nie pozwalały na uzyskanie czytelnych figur. Cząsteczki zawiesiny pokrywały całą powierzchnię badanej warstwy, tworząc tło, które utrudniało określenie pod mikroskopem granic obszarów przemagnesowań.

Warstwa WM-88, której spoiwo stanowi żywica poliwinyllobutyralowa, rozpuszcza się w alkoholu etylowym. Dla wytwarzania figur proszkowych nie można stosować zawiesin zawierających alkohol etylowy. Ze względu na niskie właściwości magnetyczne warstwy WM-88 należy stosować zawiesiny proszków o wyższych wartościach pozostałości magnetycznej np. γ - Fe_2O_3 Szost. 6.

Jako ośrodek dyspersyjny zawiesiny dla WM-88 najlepsze właściwości wykazywał roztwór kwasu laurylowego w heksanie $0,3 \div 0,4 \text{ g}$ kwasu laurylowego w 20 ml heksanu + $0,5 \text{ g}$ γ - Fe_2O_3 /.

Najwyraźniejsze figury otrzymywano z zawieszin przygotowanych w moździerz. Zawiesziny mielone 25 godz. w młynie kulowym nie dawały wyraźnych figur, cząsteczki pokrywając całą powierzchnię warstwy zacierały kontrast, a otrzymanych figur nie można było mierzyć. Figury były nieczytelne bez względu na warunki oświetlenia powierzchni warstwy. Zawiesziny uzyskiwane w wyniku nawet krótkotrwałego mielenia wykazywały tendencję do osiadania na całej powierzchni warstwy.

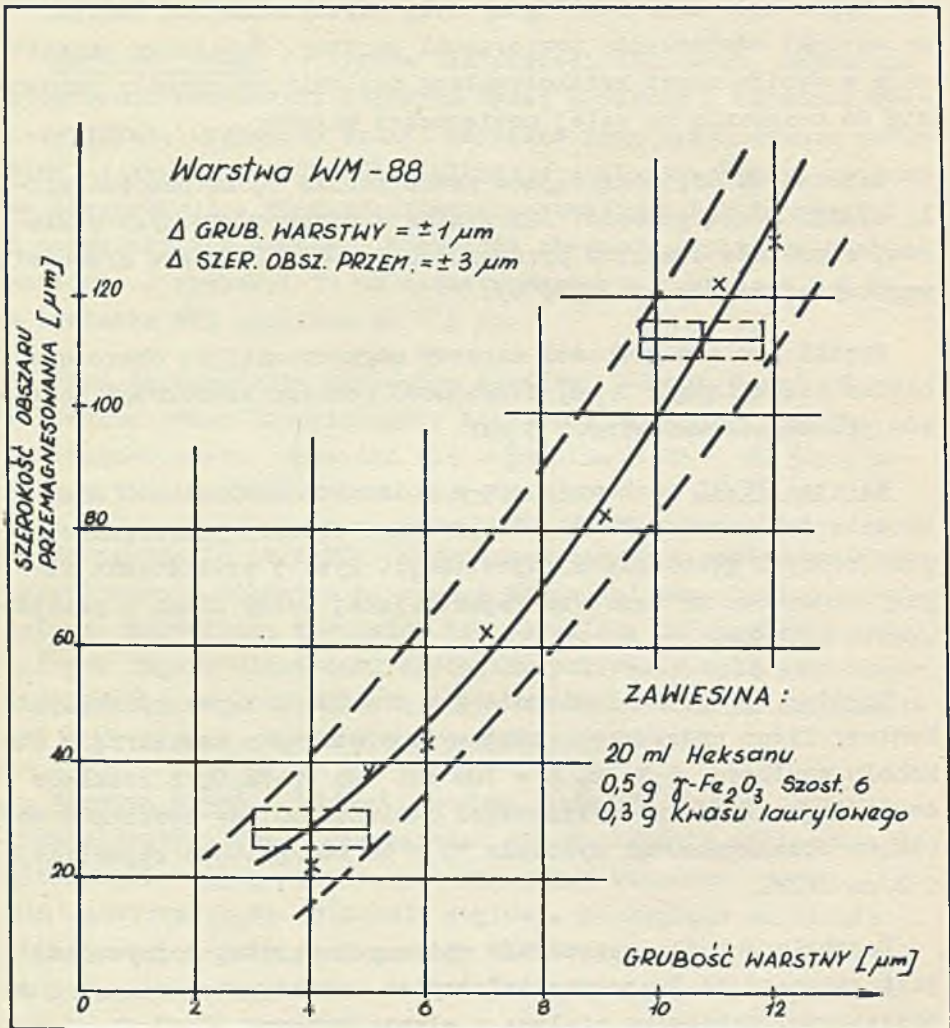
Warstwa WM-88, pokrywająca powierzchnię bębna pamięci miała zróżnicowaną grubość. Jak wynika z przeprowadzonych pomiarów, szerokość obszarów przemagnesowań rośnie wraz z grubością warstwy magnetycznej /rys. 4/.

Wyniki pomiaru grubości warstwy magnetycznej są obciążone błędem bezwzględnym $1 \mu\text{m}$, dokładność pomiaru szerokości obszarów przemagnesowań wynosi $3 \mu\text{m}$:

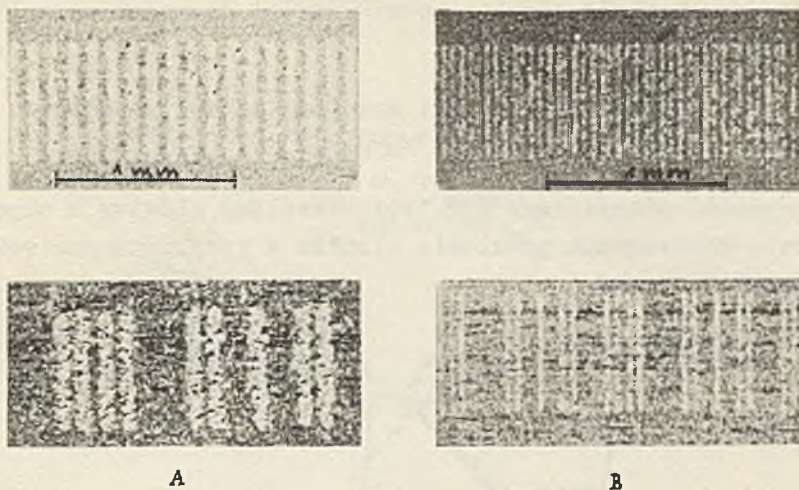
Warstwa PYRAL zachowuje się w opisanych zawieszinach identycznie jak warstwa WM-80. Obserwowano obszary przemagnesowań przy różnych gęstościach rejestracji. Rys. 5 przedstawia figury proszkowe na warstwie rejestrującej taśmy PYRAL w powiększeniu ok. 30x.

Warstwa Ni-Co. Analogicznie jak dla WM-80 najwyraźniejsze kontury figur proszkowych otrzymywano stosując zawieszinę w alkoholu etylowym $\gamma\text{-Fe}_2\text{O}_3$ H - 100 EMI lub $\gamma\text{-Fe}_2\text{O}_3$ z rozkładu dwuwodnego mrówczanu żelazawego. Stwierdzono, że szerokość obszarów przemagnesowań wynosiła 30 - 40 μm /gęstość rejestracji 9 b/mm NRZ/.

Podobnie jak dla poprzednio opisanych warstw, korzystniej jest stosować do "wywoływania" zapisu zawiesziny rozpraszane w moździerz. Zawiesziny mielone w młynie kulowym pokrywały trwałym tłem wywoływany obszar przemagnesowań, co uniemożliwiało pomiar szerokości obszarów przemagnesowań pod mikroskopem. Pomysłne rezultaty w usuwaniu tła z warstwy uzyskiwano przez przemywanie jej czystą cieczą stanowiącą ośrodek dyspersyjny zawiesziny.



Rys. 4. Szerokość obszaru przemagnesowania w funkcji grubości warstwy magnetycznej WM-88



Rys. 5. Figury proszkowe uzyskane na warstwie PYRAL HP 2N 12 N2 powiększenie 30x

A - 8 b/mm, B - 22 b/mm

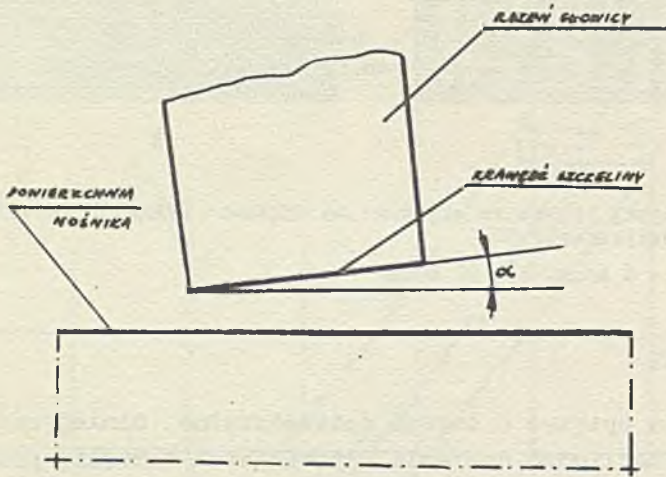
3. WNIOSKI

Badania opisane w części doświadczalnej niniejszej pracy wykazały możliwość dobrania odpowiednich zawiesin proszków ferromagnetycznych, pozwalających uzyskać czytelne odwzorowanie pola magnetycznego zarejestrowanej informacji w postaci wyrazistych figur proszkowych, łatwo usuwalnych z badanej powierzchni. Uwidocznianie zapisu na warstwie może być wielokrotnie powtarzane.

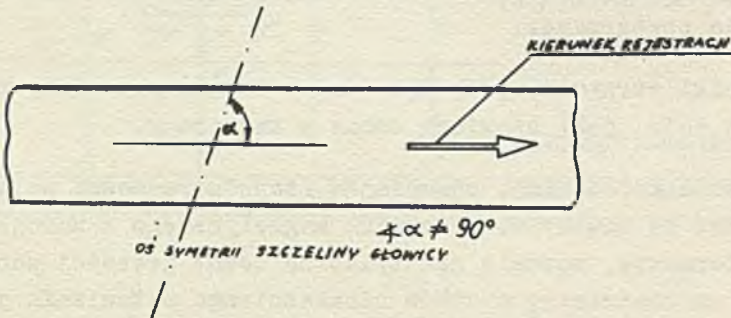
Wyniki otrzymane przy ujawnianiu zapisu pozwoliły na osiągnięcie celu, jaki stawiano sobie w tej pracy.

Metoda koloidalna, ujawniając niejednorodności pola magnetycznego na powierzchni nośnika magnetycznego z zarejestrowaną informacją, pozwala nie tylko na ocenę gęstości zapisu, ale także na obserwację skutków niewłaściwego ustawienia głowicy rejestrującej względem nośnika. Częste wady ustawienia głowicy to:

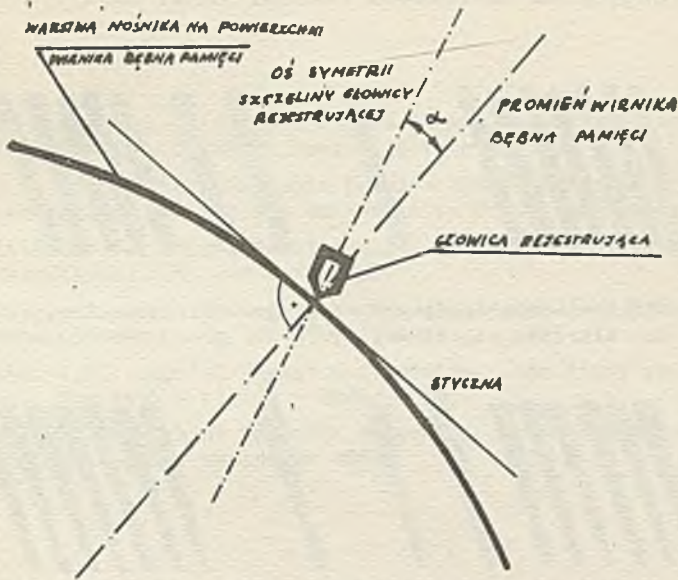
- nierówna odległość głowica-nośnik wzdłuż długości szczeliny /rys. 6/
- nieprostokątne ustawienie osi symetrii szczeliny głowicy względem zapisywanej ścieżki /rys. 7/
- niezgodność płaszczyzny symetrii szczeliny głowicy z odpowiednią płaszczyzną promienia wirnika w przypadku pamięci bębnowej /rys. 8/.



Rys. 6. Nierównoległe ustawienie czoła szczeliny głowicy względem powierzchni nośnika lub tworzącej wirnika bębna pamięci



Rys. 7. Nieprostokątne ustawienie osi symetrii szczeliny głowicy względem kierunku rejestracji



Rys. 8. Niezgodność płaszczyzny symetrii szczeliny głowicy z płaszczyzną odpowiedniego promienia wirnika bębna pamięci

Nierównoległe ustawienie płaszczyzny czoła szczeliny głowicy względem płaszczyzny warstwy nośnika na warstwie czy dysku lub względem odpowiedniej płaszczyzny stycznej do powierzchni wirnika na całej tworzącej, powoduje wystąpienie figur zbliżonych kształtem do trójkątów lub wydłużonych trapezów /rys. 9/.

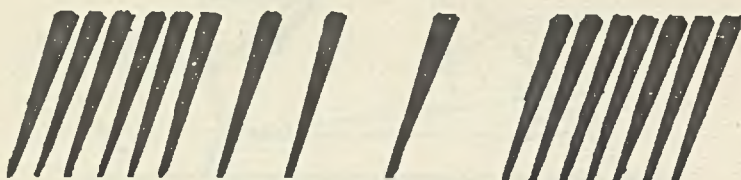


Rys. 9. Efekt nierównoległości płaszczyzny czoła szczeliny głowicy i powierzchni nośnika

Nieprostopadłe ustawienie płaszczyzny symetrii szczeliny głowicy względem kierunku rejestracji informacji objawia się pochyleniem pasków /figur/ odwzorowujących obszary przemagnesowań i zwężeniem szerokości ścieżki /rys. 10/.



Rys. 10. Efekt nieprostopadłości osi symetrii szczeliny głowicy względem kierunku rejestracji



Rys. 11. Sumaryczny efekt niewłaściwego ustawienia głowicy rejestrującej

Rys. 11 przedstawia schematycznie wynik superpozycji obu powodów zniekształceń.

Wada ustawienia głowicy polegająca na odchyleniu płaszczyzny symetrii szczeliny głowicy od płaszczyzny normalnej do powierzchni warstwy nośnika /rys. 8/, powodująca "rozmycie" pasków odwzorowujących obszary przemagnesowań, jest trudna do uchwycenia, gdyż efekty, które powoduje, nie są tak wyraźnie odwzorowane jak opisane wyżej.

Ponieważ w wielu przypadkach kontrola ustawienia głowicy nie jest możliwa ze względu na brak odpowiednich przyrządów optycznych, metoda koloidalna jest użyteczna i wygodna zarówno w przypadku taśm jak bębnow i dysków. Zaletą jej jest nie niszczenie badanej powierzchni.

Prowadzone są prace nad udoskonaleniem metody koloidalnej do uzyskania maksymalnie ostrego rysunku obszarów przemagnesowań oraz skrócenia do minimum czasu badania.

Literatura

- [1] BITTER F.: On Inhomogenities in the Magnetization of Ferromagnetic Materials, Physical Review, November 1931.
- [2] CAREY E., ISAAC E.D.: Magnetic Domains and Techniques for their Observation, The English Universities Press Ltd, London 1966.
- [3] SZCZENIOWSKI S.: Fizyka Doświadczalna cz. III, Elektryczność i Magnetyzm, PWN, Warszawa 1953.
- [4] KITTEL C.: Physical Theory of Ferromagnetic Domains, Bell Telephone Systems Monograph 1909, 1949.
- [5] HOAGLAND A.S.: Digital Magnetic Recording, John Wiley and Sons Inc., New York - London - Sydney 1963.
- [6] MEE C.: The Physics of Magnetic Recording, North - Holland Publishing Company, Amsterdam 1964.

ПОПЫТКА ПРИМЕНЕНИЯ МЕТОДА БИТТЕРА К ОЦЕНКЕ КАЧЕСТВА МАГНИТНОЙ ЗАПИСИ
НА ЗАПИСЫВАЮЩИХ СЛОЯХ ПАМЯТИ НА МАГНИТНОМ БАРАБАНЕ

Резюме

В работе рассматриваются результаты попыток применения коллоидного метода Биттера для оценки магнитной записи на разных типах записывающих слоев.

На основе литературы были рассмотрены условия возникновения порошковых фигур копирующих неоднородности магнитного поля на поверхности ферромагнита.

Исследовано два разные слоя, типовые для памяти на магнитном барабане, металлический слой Ni-Co на диске и записывающий слой ленты Rytal HP 2M 12 H2. Ввиду различных магнитных, механических и химических свойств исследованных слоев констатировано необходимость индивидуально подбирать магнитные порошки и среду дисперсии суспензии для каждого типа исследованных слоев.

Полученные порошковые фигуры наблюдались в увеличении с одновременным измерением ширины области перемангничивания.

Констатировано, что записывающая головка, неправильно установленная к поверхности магнитного слоя, вызывает деформацию порошковых фигур.

Поданы фотографии поверхности с магнитной записью, а также представлены схематические деформации порошковых фигур, возникшие вследствие неправильной установки записывающей головки.

На основе полученных результатов показана попытка оценки пригодности коллоидного метода для скорой и упрощенной установки головки с одновременным измерением ширины.

A TRIAL TO APPLY THE BITTER'S METHOD TO ESTIMATE THE QUALITY OF MAGNETIC RECORD ON DRUM STORAGE RECORDING LAYERS

Summary

The paper discusses the results of trials to apply the colloid Bitter's method to estimate magnetic record on various types of recording layers.

Discussed conditions of colloid pattern origination, mapping the non uniformity of the magnetic field on the surface of a ferromagnetic.

Examined: two different layers, typical for drum storage, a disk metal layer Ni-Co and a recording layer of the type Pyral HP 2N 12 N2.

Because of different magnetic, mechanical and chemical properties of the investigated layers, magnetic powders and suspension dispersion medium had to be individually chosen for each type of investigated layers.

The obtained colloid patterns were examined, magnified 25-75 times. Simultaneously the width of remagnetization areas was measured.

It was ascertained that a wrong arrangement of the recording head to the surface of a magnetic layer causes a deformation of the colloid pattern.

Photographies of the surface with a magnetic record are included, and a schematic colloid pattern deformation, caused by a wrong arrangement of a magnetic head is presented.

On the basis of the obtained results the trial to estimate the usefulness of the colloid method for a quick and simplified control of head arrangement is shown.

AUTORZY ARTYKUŁÓW ZAMIESZCZONYCH W ZESZYCIE 2 "PRAC IMM"

CEMMA - JĘZYK DO SYMULACJI UKŁADÓW DYNAMICZNYCH

Waldemar MADEJ mgr inż. Studia ukończył w 1967 r. na Wydziale Elektroniki Politechniki Warszawskiej. W roku 1967 rozpoczął pracę w Zakładzie Sterowania IMM, gdzie pracuje do chwili obecnej. Początkowo zajmował się zagadnieniami wykorzystania maszyny cyfrowej do badania procesów dynamicznych i opracowywaniem języków symulacyjnych ułatwiających takie badanie. Jest autorem publikacji na ten temat. Od połowy roku 1970 zajmuje się problemami oprogramowania maszyn cyfrowych pracujących w systemie "real-time".

ASTEK I - JĘZYK DO STATYSTYCZNEJ ANALIZY
DANYCH EKSPERYMENTALNYCH

Piotr BIELKOWICZ mgr. W roku 1968 ukończył Wydział Matematyki na Uniwersytecie Warszawskim i rozpoczął pracę w Instytucie Maszyn Matematycznych, gdzie pracuje w Zakładzie Sterowania, zajmując się statystyczną obróbką danych eksperymentalnych.

Piotr PERKOWSKI, mgr inż. ukończył w 1967 r. Wydział Elektroniki Politechniki Warszawskiej. Pracę rozpoczął w 1967 r. w Instytucie Maszyn Matematycznych w Zakładzie Sterowania. Zajmuje się zagadnieniami symulacji procesów dyskretnych oraz statystycznej obróbki danych eksperymentalnych. Jest kierownikiem Pracowni Modelowania w Zakładzie Sterowania IMM.

SYSTEM MAGAZYNOWANIA I AKTUALIZACJI DOKUMENTÓW
I JEGO ZASTOSOWANIE W PROCESIE URUCHAMIANIA PROGRAMÓW

Jacek WITASZEK mgr. W 1958 r. ukończył Wydział Matematyczno-Fizyczno-Chemiczny Uniwersytetu Jagiellońskiego. W Instytucie Maszyn Matematycznych rozpoczął pracę w 1960 roku. Zajmuje się

problematyką związaną z konstrukcją oprogramowania, a zwłaszcza systemów operacyjnych dla maszyn ZAM. Jest autorem szeregu opracowań i publikacji.

OPROGRAMOWANIE UNIWERSALNE MASZYNY ZAM 41 DO EPD

Hanna MYSIOR mgr. Ukończyła w 1959 r. studia na Wydziale Matematycznym Uniwersytetu Warszawskiego. Od 1959 r. pracuje w Instytucie Maszyn Matematycznych początkowo w Biurze Obliczeń i Programów Zakładu Doświadczalnego IMM, następnie w Zakładzie Przetwarzania Danych. Obecnie pracuje na stanowisku adiunkta w Zakładzie Sterowania. Jest autorką szeregu opracowań dotyczących programowania systemów przetwarzania danych.

PRÓBA ZASTOSOWANIA METODY BITTERA DO OCENY JAKOŚCI ZAPISU MAGNETYCZNEGO NA WARSTWACH REJESTRUJĄCYCH PAMIĘCI BĘBNOWYCH

Hanna KUŹNICKA mgr inż. Studia ukończyła w 1963 r. na Wydziale Łączności Politechniki Warszawskiej. W latach 1962-1966 pracowała w Centralnym Biurze Naukowo-Badawczym i Konstrukcyjnym Półprzewodników fabryki półprzewodników "TEWA". Od 1966 r. jest pracownikiem naukowo-badawczym w Instytucie Maszyn Matematycznych, gdzie pracuje nad problematyką miernictwa magnetycznego i doskonaleniem przez polaryzację magnetycznych warstw rejestrujących dla pamięci bębnowych matematycznych maszyn cyfrowych. Jest autorką dwóch publikacji naukowych.



BIBLIOTEKA GŁÓWNA
Politechniki Śląskiej

P 2229 | 70