

Krzysztof TANNENBERG
Andrzej KWIECIEN
Daniel SERAFIN

SYSTEM DYNAMICZNEJ SYMULACJI GRAFICZNEJ

Streszczenie. W artykule omówiono zasadę działania grafoskopu GF-1 oraz jego podstawowe oprogramowanie. Omówiono język dyrektyw, służący do programowego manipulowania obrazem, ze szczególnym uwzględnieniem jego dynamicznych zmian.

Wstęp

Jednym z najbardziej efektywnych sposobów komunikowania się człowieka z maszyną cyfrową jest dialog w oparciu o metody graficzne z wykorzystaniem grafoskopów. Zastosowanie grafoskopu jako urządzenia wyjściowego stanowi ułatwienie w wielu dziedzinach działalności człowieka, szczególnie tam, gdzie działalność ta prowadzona jest w oparciu o elektroniczną technikę obliczeniową. Możliwość śledzenia na bieżąco wyników bądź etapów pracy programu, z możliwością niemalże natychmiastowej ingerencji, mającej na celu wprowadzenie zmian do oglądanego obrazu, stanowi związącą perspektywę zastosowania grafoskopów i prowadzenia prac w tym kierunku.

Technika uzyskiwania obrazów na ekranie grafoskopu jest ściśle związana z jego typem, który jest zdeterminowany przez typ zastosowanej lampy obrazowej. Aktualnie stosuje się trzy rodzaje lamp obrazowych: lampy kinoskopowe, oscyloskopowe i oscyloskopowe z pamięcią.

Grafoskop GF-1 zbudowany jest w oparciu o lampę oscyloskopową. Zastosowanie w nim sprzętowego generatora krzywych drugiego stopnia pozwala wykreślać obrazy składające się z wektorów, okręgów, elips oraz łuków eliptycznych i kołowych. Grafoskop GF-1 nie posiada własnej pamięci obrazowej oraz pióra świetlnego. Rozdzielczość jego wynosi 1024x1024 punktów. Nie jest również wyposażony w interfejs umożliwiający bezpośrednią współpracę z maszynami cyfrowymi ODRA serii 1300. Brak interfejsu nasunął koncepcję przyłączenia grafoskopu do pamięci taśmowej maszyny cyfrowej, a konkretnie do głowicy zapisująco-czytającej jednostki taśmowej PT-3. Tak przyjęte rozwiązanie narzuciło maksymalną liczbę elementów, jakie mogą się pojawić jednocześnie na ekranie. Aby obraz nie migotał, na ekranie nie może pojawić się więcej niż 255 elementów jednocześnie. Wiąże więc to oczywiście z wymaganym dla płynnego obrazu czasem repetycji.

Ogólna zasada działania systemu graficznego jest następująca: komputer po otrzymaniu informacji od użytkownika przetwarza je i kolejne sekwencje obrazu wysyła na taśmę magnetyczną. W trakcie transmisji obraz pojawia się na ekranie grafoskopu. Każdy element pojawiający się na ekranie jest opisany poprzez 68 bitów, co zajmuje w maszynie cyfrowej 3 słowa maszynowe (4 bity są niewykorzystane).

Aby wyświetlić na ekranie 255 elementów, trzeba wysłać na taśmę magnetyczną blok informacji o długości 765 słów. Powoduje to straty czasu związane z koniecznością częstych transmisji.

Brak pióra świetlnego pociąga za sobą konieczność numerowania elementów w obrazie, aby umożliwić jego korektę. Korekta może polegać bądź na wprowadzeniu nowych elementów, bądź na zmianie parametrów elementów już istniejących. Następną konsekwencją braku pióra świetlnego jest konieczność zaprojektowania, przed użyciem grafoskopu, obrazu na papierze i określenie z jakiego rodzaju elementów obraz ma się składać oraz podania parametrów elementów (np. współrzędne środka okręgu, promień i jasność, współrzędne początku i końca wektora, itp.).

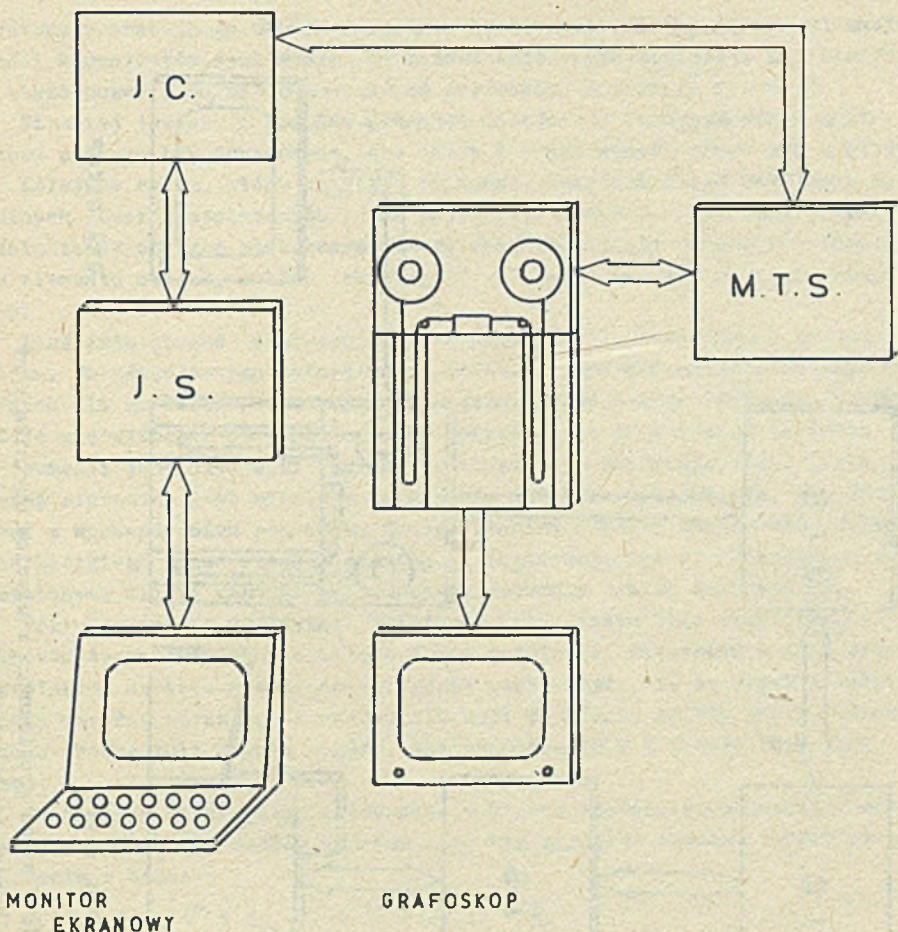
1. Założenia systemu dynamicznej symulacji obrazu

Aby system mógł być efektywnie wykorzystany, jednym z podstawowych celów, jakie należało osiągnąć, jest możliwość pracy konwersacyjnej. Jest to podstawowy wymóg stawiany przed tego rodzaju systemami.

Grafoskop GF-1 nie posiada własnej klawiatury alfanumerycznej i o czym była już mowa, interface-u do bezpośredniej współpracy z jednostką centralną maszyny cyfrowej. Aby umożliwić pracę konwersacyjną, trzeba dysponować oddzielnym monitorem z klawiaturą alfanumeryczną. Do tego celu nadaje się produkowany w kraju monitor ekranowy wraz z jednostką sterującą, współpracujący bezpośrednio z komputerem i stanowiący jego urządzenie dialogowe.

Dysponując wspomnianym monitorem oraz systemem operacyjnym GEORGE-3, można w bardzo łatwy sposób zapewnić bezpośrednią komunikację użytkownika systemu z programem obsługującym, który pracuje pod kontrolą programu EXECUTIVE. Program obsługi grafoskopu może być w każdej chwili zatrzymany i operator za pomocą odpowiednich komunikatów systemowych może ponownie zainicjować jego pracę, dokonując w międzyczasie niezbędnych korekt w oglądanym obrazie. Konfiguracja systemu z wykorzystaniem monitora ekranowego jest przedstawiona na rys. 1.

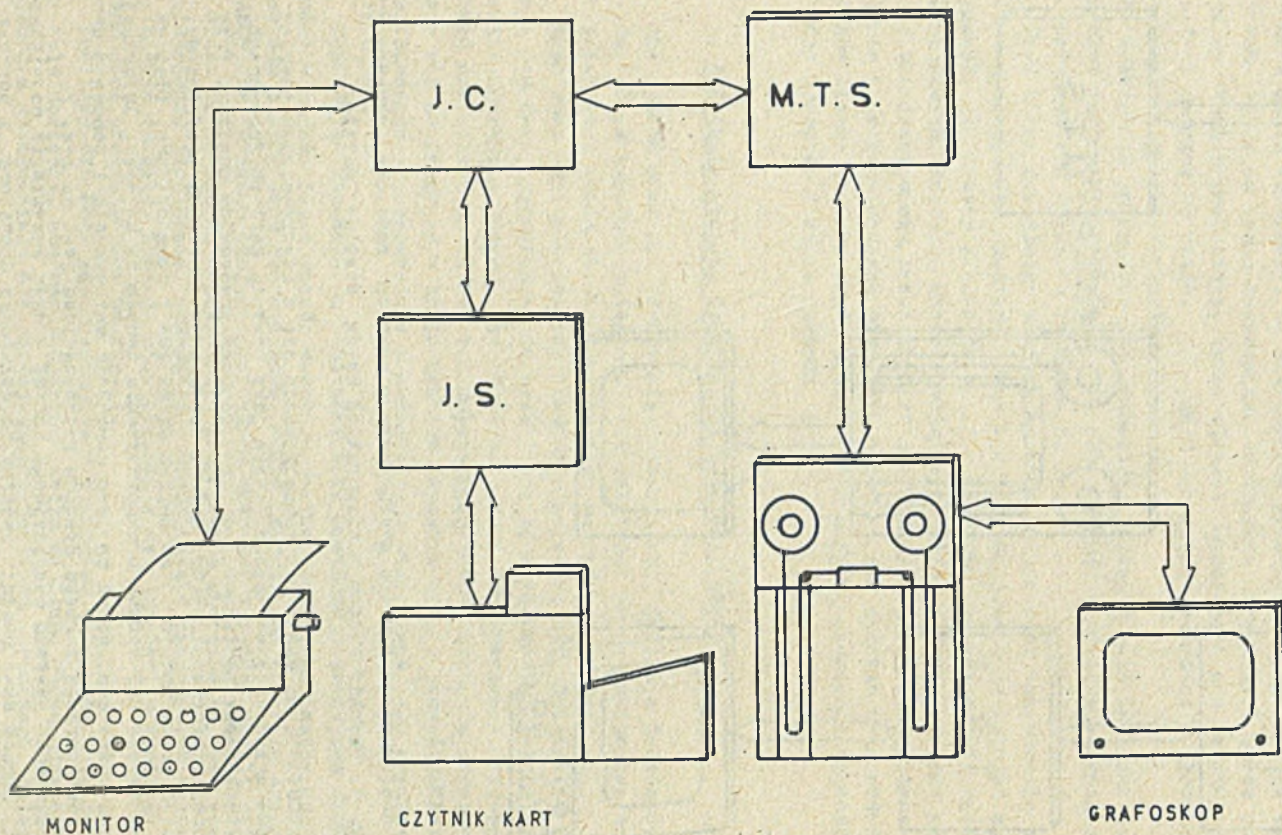
Monitory ekranowe współpracujące z jednostką centralną nie są jeszcze na tyle dostępne, aby móc się oprzeć tylko na nich. Chcąc udostępnić system dla użytkowników nie posiadających monitora ekranowego, należało opracować alternatywne rozwiązanie, dające możliwość pracy zbliżonej do konwersacyjnej. Program obsługi grafoskopu, pracujący pod nadzorem pro-



Rys. 1. Konfiguracja systemu graficznego z użyciem monitora ekranowego
 Fig. 1. Graphic system configuration using display monitor

gramu EXECUTIVE, ma możliwość komunikowania się z użytkownikiem poprzez podstawowe urządzenia wejściowe, takie jak: czytnik kart perforowanych bądź czytnik taśmy papierowej, a także poprzez tradycyjny monitor. Przy dużej porcji informacji, które należy dostarczyć systemowi graficznemu, komunikacja przez monitor jest bardzo uciążliwa i czasochłonna. Dlatego najlepszy wydaje się sposób przygotowywania danych na kartach. Informacje programowi można dostarczyć po jego uprzednim zatrzymaniu i ponownym uruchomieniu. Konfiguracja takiego systemu przedstawiona jest na rys. 2.

Wykorzystanie systemu operacyjnego GEORGE 3 daje oczywiście największe korzyści. Skracza czas reakcji systemu na działanie operatora, a poprzez zastosowanie makroinstrukcji obsługi programu sterującego grafoskopem dia



Rys. 2. Konfiguracja systemu graficznego z użyciem czytnika kart
 Fig. 2. Graphic system configuration using card reader

systemu operacyjnego GEORGE 3 zwalnia praktycznie obsługującego od znajomości komunikatów systemowych (w każdym bądź razie mniejsza ich ilość), a także pozwala mu nie odchodzić od grafoskopu w trakcie pracy.

Niemniej jednak, z powodów podanych wcześniej, oprogramowanie grafoskopu zostało tak opracowane, aby jeden i drugi sposób pracy był możliwy.

Kolejnym celem, który należało osiągnąć, była możliwość kreślenia dowolnych obrazów statycznych i jak najszersze manipulowanie nimi. Przez manipulację obrazem statycznym należy rozumieć zmianę parametrów dowolnego elementu obrazu, takich jak długość wektora, jasność, promień okręgu itp.

Kreślenie obrazów statycznych jest procesem nie stanowiącym dużego problemu. Po zdekodowaniu informacji o obrazie i przygotowaniu przez komputer danych dla grafoskopu następuje ich transmisja do pamięci taśmowej i obraz staje się widoczny na ekranie. Czas potrzebny na proces przetwarzania informacji jest niewielki (parametry obrazu nie zmieniają się), toteż obraz statyczny jest widoczny na ekranie niemalże natychmiast. Podobnie jest z wprowadzeniem poprawek, manipulowaniem obrazem statycznym. Uzyskanie krótkiego czasu reakcji systemu na ingerencję operatora uzyskano między innymi dzięki zastosowaniu w oprogramowaniu języka dyrektyw [2].

Przystępując do realizacji kolejnego celu, jakim było wyświetlanie obrazów dynamicznych, poczyniono pewne założenia. Pierwszym z nich było kreślenie obrazów dynamicznych, gdzie ruch odbywa się na płaszczyźnie. Chcąc uzyskać obraz np. toczącej się kuli po równi, należy podać równania ruchu środka kuli w taki sposób, aby współrzędne X i Y były funkcjami czasu.

Dochodzimy do drugiego założenia, w którym postanowiono dopuścić następujące funkcje, według których mają się zmienić dowolne parametry elementu w czasie:

$$1) Z(t) = A_n \cdot t^n + A_{n-1} \cdot t^{n-1} + \dots + A_0$$

$$2) Z(t) = A \cdot \sin(\omega t + \alpha) + A_0$$

$$3) Z(t) = A \cdot \cos(\omega t + \alpha) + A_0$$

$$4) Z(t) = A \cdot e^{\omega t + \alpha} + A_0$$

Ta grupa funkcji praktycznie pozwala określić dowolną trajektorię ruchu.

Fakt zmiany parametru bądź nawet wszystkich parametrów dowolnej liczby elementów obrazu w czasie powoduje brak czasu maszyny na to, aby generować w czasie rzeczywistym nowe fazy obrazu i aby obraz ten był płynny. Czas ten wydłuża się oczywiście tym bardziej, im więcej elementów obrazu zmienia swe położenia i im bardziej skomplikowany jest ich ruch. Wobec tego należało w pierwszej kolejności przygotować pełną sekwencję kolejnych faz ruchu a dopiero potem przejść do jego wyświetlania.

Kolejne fazy ruchu są przygotowywane przez program i składowane w pamięci zewnętrznej, a następnie odczytywane z niej i wysyłane do grafoskopu.

Takie rozwiązanie spowodowało wydłużenie czasu oczekiwania na obraz.

Wydłużenie czasu oczekiwania na obraz nie zawsze jest tylko związane ze skomplikowanym ruchem elementu czy ich ilością pozostającą w ruchu. Ponieważ przy skalowaniu czasu trwania ruchu przyjęto zasadę, że najkrócej trwający ruch elementu, aby był widoczny na ekranie, ma trwać 3 sekundy, to nawet przy zmianach w czasie dwóch parametrów jednego tylko elementu czas oczekiwania na obraz może być bardzo długi.

2. Język systemu

Język systemu graficznego opartego na grafoskopie GF-1 jest prostym językiem dyrektyw [2]. Dyrektywy, którymi posługuje się użytkownik, nie są przekształcone w program wynikowy, a są jedynie wykonywane przez program użytkowy bezpośrednio po ich odebraniu od operatora. Istotną cechą każdego języka dyrektyw jest jego prostota syntaktyczna. Nie dysponuje on instrukcjami warunkowymi. Ograniczony jest do niezbędnego minimum zbiór typów danych. Każda dyrektywa składa się zazwyczaj z trzech elementów:

- a) kodu określającego rodzaj polecenia przekazywanego do wykonania systemowi,
- b) parametrów dyrektywy oddzielonych spacjami,
- c) ogranicznika dyrektywy.

Prostota języka dyrektyw daje określone korzyści, wynikające z przejrzystości zapisu oraz z szybkiego przyswojenia sobie zasad posługiwania się nim.

2.1. Zasady posługiwania się językiem dyrektyw systemu dynamicznej symulacji graficznej

Każda dyrektywa ma postać następującą:

$$\text{NAME}_{\nabla} \text{NEL}_{\nabla} L_{1\nabla} L_{2\nabla} \dots L_n$$

gdzie:

NAME - jest nazwą dyrektywy,

NEL - numer elementu obrazu, którego dyrektywa dotyczy,

L_1+L_n - lista parametrów dyrektywy.

Każdy element dyrektywy jest oddzielony spacją, a ogranicznikiem dyrektywy jest koniec rekordu, np. koniec karty, natomiast ogranicznikiem pliku dyrektyw znak "x".

2.1.1. Dyrektywy kreślenia statycznych elementów

Przy kreśleniu obrazów statycznych użytkownik dysponuje następującymi dyrektywami:

OK ∇ NE ∇ X₀ ∇ R ∇ Y₀ ∇ JASN

- kreślenie okręgu o promieniu R, współrzędnych środka (X₀, Y₀) i jasności JASN

WE ∇ NE ∇ X_p ∇ X_k ∇ Y_p ∇ Y_k ∇ JASN

- kreślenie wektora o współrzędnych początku (X_p, Y_p), końca (X_k, Y_k) i jasności JASN

EL ∇ NE ∇ X_s ∇ A ∇ Y_s ∇ B ∇ JASN

- kreślenie elipsy o współrzędnych środka (X_s, Y_s), długościach osi A (oś X), B (oś Y) i jasności JASN

FE ∇ NE ∇ X_s ∇ A ∇ Y_s ∇ B ∇ KS ∇ KK ∇ JASN

- kreślenie łuku eliptycznego o środku (X_s, Y_s), długościach osi A (oś X), B (oś Y), kącie startu kreślenia KS i kącie końca kreślenia elipsy¹⁾ KK i jasności JASN

FE ∇ NE ∇ X₀ ∇ R ∇ Y₀ ∇ KS ∇ KK ∇ JASN

- kreślenie łuku kołowego o środku (X₀, Y₀), promieniu R, kącie startu kreślenia KS, kącie końca kreślenia KK¹⁾ i jasności JASN.

Po ciągu tego typu dyrektywach należy umieścić znak "*".

2.1.2. Dyrektywy przeznaczone do manipulowania obrazem statycznym

Chcąc dokonać ingerencji w obraz już nakreślony można posłużyć się poniższymi dyrektywami:

DEL

- usuwanie wszystkich elementów

*

DEL ∇ NE1 ∇ NE2 ∇ NE3 ... NE N

- usuwanie elementów o numerach 1, 2, 3, ... N

*

SCLE ∇ R

- skalowanie obrazu (tzn. zmiana każdej współrzędnej każdego elementu) w skali R

MOVE ∇ X_k ∇ Y_k ∇ NEL

- przesunięcie elementu o numerze NEL o wektor W(X_k, Y_k)

*

MOVE ∇ X_k ∇ Y_k ∇ ALL

- przesunięcie wszystkich elementów o wektor W(X_k, Y_k)

*

(Mówiąc o przesunięciu mamy na myśli przesuwanie początku i końca wektora środka okręgu, elipsy, łuku koł. i elipt.)

SIGN ∇ N

- wyświetlanie kolejno wszystkich elementów obrazu. Liczba W (dotychczas) określa czas trwania na ekranie elementu. W czasie wyświetlania elementu jest wyświetlany również jego numer

PONT NEL

- wyszczególnienie elementu o numerze NEL poprzez na przemian jego gośnięcie i pojawianie się na ekranie.

2.1.3. Dyrektywy służące do dynamicznego wykreślenia obrazu

Dyrektywy tego typu w zasadzie niczym nie różnią się od poprzednich. Jedynie dwie pierwsze litery dyrektywy określają typ funkcji, wg której ma zmieniać się element o numerze NEL. I tak dyrektywa rozpoczynająca się od liter WE oznacza, że mamy do czynienia z funkcją, która jest wielomianem, dyrektywy zaczynające się od liter SI, CO, EX oznaczają, że funkcje są kolejno sinusem, cosinusem, funkcją wykładniczą e^x . Dodatkowo należy podać przedział czasu Δt , w którym dany element ma zmieniać swe parametry.

Postać dyrektyw jest następująca:

a) funkcja wielomianowa typu

$$A(t) = A_n t^n + A_{n-1} t^{n-1} + \dots + A_0$$

np. WELX ∇ NEL ∇ T_{MIN} ∇ T_{MAX} ∇ A_n ∇ A_{n-1} \dots A₀

b) funkcje typu sinus, cosinus, e^x

$$A(t) = A_1 \sin(\omega t + \alpha) + A_0$$

np. SINX ∇ NEL ∇ T_{MIN} ∇ T_{MAX} ∇ A₁ ∇ ω ∇ α ∇ A₀

$$A(t) = A_1 \cos(\omega t + \alpha) + A_0$$

np. COSX ∇ NEL ∇ T_{MIN} ∇ T_{MAX} ∇ A₁ ∇ ω ∇ α ∇ A₀

$$A(t) = A_1 e^{(\omega t + \alpha)} + A_0$$

np. EXPX ∇ NEL ∇ T_{MIN} ∇ T_{MAX} ∇ A₁ ∇ ω ∇ α ∇ A₀

2.1.3.1. Dyrektywy dynamicznego przetwarzania dotyczące wektora

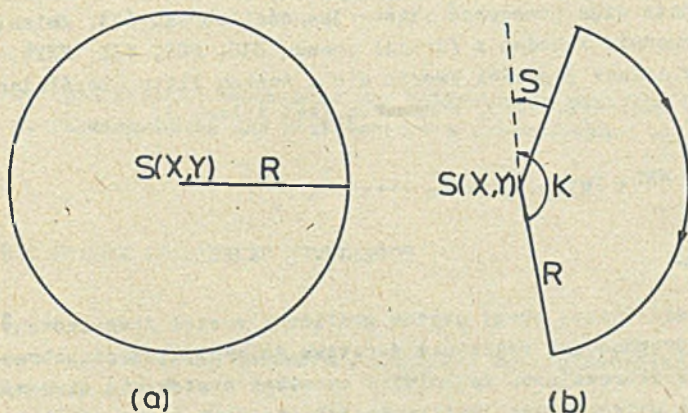
Aby wykreślić wektor należy podać współrzędne jego początku (XP, YP) i końca (XK, YK). Każda z czterech podanych wielkości określających położenie wektora, tzn. XP, YP, XK, YK może zmieniać się według podanych wcześniej funkcji, nazwy dyrektyw będą następujące:

EX|XK, EX|XP, CO|XK, CO|XP, SI|XK, SI|XP, WE|XK, WE|XP.

Powyższe dyrektywy dotyczą zmian współrzędnych X wektora. Analogicznie tworzy się nazwy dyrektyw dotyczących współrzędnych Y, wstawiając w miejsce XP bądź XK, YP oraz YK.

2.1.3.2. Dyrektywy dynamicznego przetwarzania, dotyczące okręgu i łuku kołowego

Definiując okrąg bądź łuk kołowy należy podać (rys. 3) współrzędne środka i promień, a w przypadku łuku kołowego jeszcze dodatkowo kąt początku (S) i końca łuku (K).



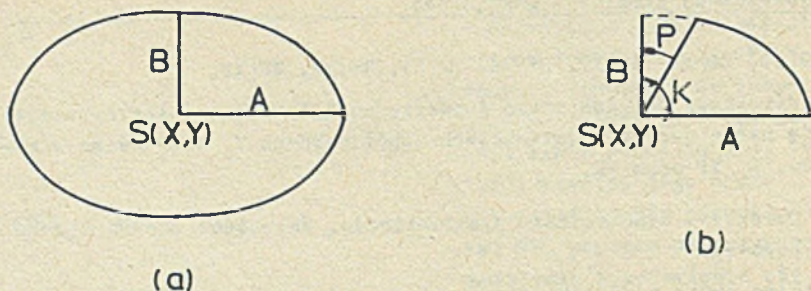
Rys. 3 a) Definicja okręgu, b) Definicja łuku kołowego
Fig. 3 a) Circle definition, b) Circle arc definition

Każdy z wymienionych parametrów, tzn. X, Y, R, S, K, może zmieniać się w czasie. Nazwy dyrektyw tworzy się w ten sposób, że pierwsze trzy litery określają typ funkcji (SIN, COS, EXP, WEL), a czwartą jest litera określająca parametr (X,Y,R,S,K).

2.1.3.3. Dyrektywy dynamicznego przetwarzania elipsy i łuku eliptycznego

Definiując elipsę należy podać współrzędne jej środka (X,Y) oraz długość osi (A,B), a w przypadku łuku eliptycznego dodatkowo jeszcze kąt początku (P) i końca łuku (K) (rys. 4).

Każdy z wymienionych parametrów może zmieniać się w czasie. Nazwy odpowiednich dyrektyw tworzy się w ten sposób, że pierwsze trzy litery określają typ funkcji (SIN, COS, EXP, WEL), a czwartą jest litera określająca parametr (X,Y,S,K,A,B).



Rys. 4 a) Definicja elipsy, b) Definicja łuku eliptycznego
 Fig. 4 a) Ellipse definition, b) Elliptic arc definition

2.1.3.4. Zmiany w czasie jasności obrazu

Umożliwiono korzystanie z 32-stopniowej skali jasności (0-31) obrazu. Programista może więc powodować zmiany jasności obrazu (J). Zmiany te mogą odbywać się zgodnie z jedną z funkcji czasu (SIN, COS, WEL, EXP). Dyrektyw powodujące zmiany jasności tworzy się z trzech liter określających typ funkcji (SIN, COS, WEL, EXP), a czwartą jest litera J .

Na przykład:

$$WEL J \nabla NEL \nabla T_{MIN} \nabla T_{MAX} \nabla A_n \dots A_0$$

3. Zakończenie

Opisany w niniejszej pracy system graficzny został zrealizowany i przetestowany. Zaproponowany repertuar dyrektyw do manipulowania obrazem wydaje się być wystarczający. Jakkolwiek omawiany system nie stanowi idealnego narzędzia o uniwersalnych możliwościach, to jest jednym z nielicznych, dających tak duże możliwości ingerowania w oglądany obraz.

Aktualna wersja systemu nie jest ostatnia. Prowadzone są działania dla jego stałego doskonalenia i to zarówno od strony hardware'owej, jak i software'owej.

Dużą trudnością związaną z dynamicznym kreśleniem obrazu jest brak czasu maszynowego niezbędnego do uzyskiwania obrazów w czasie rzeczywistym. Wprowadzenie wieloczołowego oprogramowania systemu oraz próba wykorzystania innego rodzaju pamięci masowej (np. pamięć bębnowa) pozwolą zmniejszyć czas reakcji systemu na ingerencje z zewnątrz.

LITERATURA

- [1] K. Tannenberga praca doktorska pt.: "Grafoskop ze sprzêtowym generatorem krzywych drugiego stopnia".
- [2] Praca zbiorowa: Grafoskopy w systemach komputerowych. WNT, Warszawa 1980.

Recenzent: Prof. dr hab. inż. Andrzej Grzywák

Wpłynęło do Redakcji: 20.12.1982.

СИСТЕМА ДИНАМИЧЕСКОЙ ГРАФИЧЕСКОЙ ИМИТАЦИИ

Резюме

В статье представлен способ действия графического монитора ГФ-1 а также язык его программирования для статического и динамического манипулирования образом.

THE GRAPHIC SYSTEM OF DYNAMIC SIMULATION

Summary

The paper presents the hardware principles and fundamental software of graphic monitor GF-1.

The programming language used for manipulation of static and dynamic images is described.