

POLSKIE TOWARZYSTWO
ELEKTROTECHNIKI TEORETYCZNEJ I STOSOWANEJ
ODDZIAŁ WARSZAWSKI

WYDZIAŁ ELEKTRYCZNY POLITECHNIKI WARSZAWSKIEJ

PRACE IX SYMPOZJUM

SPD - 9

**SYMULACJA PROCESÓW
DYNAMICZNYCH**

XXXV lat PTETiS

Polana Chochołowska

10 - 14 czerwca 1996 r.

Barbara Kuraś, Jacek Kuraś

GODYS-PC JĘZYK DO SYMULACJI SYSTEMÓW O DZIAŁANIU CIĄGLYM

1. Wprowadzenie

Pod koniec lat siedemdziesiątych w *Institucie Informatyki Uniwersytetu Jagiellońskiego* prowadzone były prace badawczo - projektowe nad językami symulacyjnymi stosowanymi do modelowania systemów ciągłych. Efektem tych prac były: wzorcowy język *Godys-4* [3] (*Graf Oriented Dynamic System Simulator*), oraz język *Godys-5* [1,2] zaimplementowany na komputerach serii *Odra 1300*. Pozytywne doświadczenia zarówno w zespołach badawczych jak też w dydaktyce spowodowały dalsze prace nad rozwojem języka (rozbudowa języka o moduł optymalizacji parametrycznej). W wyniku tych prac powstał język *Godys-6* zaimplementowany na komputerach serii *Riad*. Popularność i dostępność mikrokomputerów osobistych oraz zapotrzebowanie pewnych zespołów badawczych było motywem realizacji kolejnej wersji języka *Godys*. Językiem tym jest *Godys-PC* [4,5]. W referacie tym przedstawiona została najnowsza wersja tego języka.

2. Podstawowe informacje o języku

Język *Godys-PC* zrealizowany został w formie dwóch programów: kompilator języka opisu modelu i system wykonawczy. Podstawą sporządzenia programu w języku *Godys-PC* jest zapis modelu matematycznego badanego systemu, podany w formie układu równań różniczkowych, całkowych i algebraicznych, a zasadniczą częścią programu jest opis modelu. W wyniku kompilacji programu opisującego model symulowanego systemu, otrzymujemy równoważny program w języku pewnej maszyny abstrakcyjnej. Program ten jest natępnie interpretowany przez system wykonawczy. Dla każdego eksperymentu symulacyjnego można w prosty sposób określać parametry modelu badanego systemu i/lub atrybuty symulatora. Istnieje możliwość przeprowadzenia ciągu eksperymentów, zmierzających do automatycznego wyznaczenia optymalnych - w sensie minimalizacji pewnej zadanej funkcji - wartości parametrów modelu. Wyniki eksperymentów można prezentować w postaci przejrzystych tabel i wykresów, a także wykorzystywać postsymulacyjnie w programach analizy statystycznej lub innych. Wspomniane tabele i wykresy mogą być tworzone na monitorze, drukarce lub ploterze. Przebieg eksperymentu symulacyjnego wykonywanego na modelu opisany jest ciągiem dyrektyw, z których każda jest realizowana osobno. Wykonanie kolejnej dyrektywy rozpoczyna się dopiero po całkowitym zakończeniu reali-

zacji dyrektywy bezpośrednio jej poprzedzającej. Wspomniane powyżej programy języka *Godys-PC* uzupełnia program preprocesora podmodeli umożliwiający tworzenie własnych makrodefinicji dla języka opisu modelu. Język pracuje pod kontrolą systemu operacyjnego *MS/PC DOS*. Cały system jest obsługiwany za pomocą bardzo prostego i wygodnego menu.

Zasadniczymi własnościami języka *Godys-PC* są: • prosty i czytelny sposób zapisywania modelu i sterowania przebiegiem symulacji, • różnorodność operacji standardowych, • możliwość wprowadzenia funkcji opcjonalnych (definiowanych przez użytkownika), • różnorodność postaci informacji wyjściowej, • różnorodne metody całkowania numerycznego, • kilka różnych algorytmów optymalizacji parametrycznej, • dokładna i przejrzysta diagnostyka błędów kompilacji i wykonania, • możliwość interakcyjnego wpływania na przebieg procesu symulacji, • elastyczny sposób sterowania drukowaniem, • środowisko sterowane za pomocą wygodnego menu.

3. Struktura programu w języku *Godys-PC*

Program napisany w języku *Godys-PC* składa się z dwóch modułów: 1) opisu modelu symulowanego obiektu, 2) opisu eksperymentów przeprowadzanych na tym modelu.

Moduł opisu modelu ma następującą strukturę: • deklaracja *MODEL* - określa symboliczną nazwę modelu, • deklaracje *MAP*, *NOMAP* - sterują wyprowadzaniem informacji o programie wynikowym, która umożliwia programiście precyzyjną analizę programu. • deklaracje *PREPARE*, *PARAMT*, *CONST*, *DECLARE*, *RENAME* - służą do definiowania obiektów używanych w opisie modelu (zmiennych, parametrów i stałych), • deklaracja *INITIAL* - początek tzw. sekcji inicjalnej opisu modelu, • instrukcje opisu modelu (należące do sekcji inicjalnej), • deklaracja *END* - koniec sekcji inicjalnej, • deklaracja *DYNAMIC* - początek tzw. sekcji dynamicznej opisu modelu, • instrukcje opisu modelu (należące do sekcji dynamicznej), • deklaracja *END* - koniec sekcji dynamicznej i całego opisu modelu.

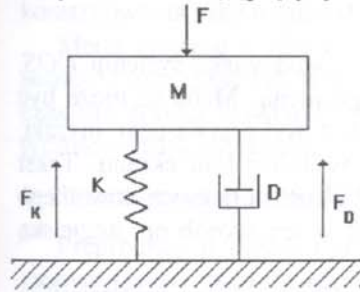
Sekwencję dyrektyw sterujących systemem wykonawczym tworzą: • *LOAD* - identyfikuje symulowany model, • *DATA* - służy do nadawania wartości parametrom symulowanego modelu, • *EXECUTE*, *CONTINUE* - rozpoczęcie nowego lub kontynuacja poprzedniego eksperymentu symulacyjnego, • *HDR* - tworzenie nagłówków opisujących wydawnictwa prezentujące wyniki symulacji, • *PRPLOT*, *PLOTXY*, *GRAPH*, *PRINT* - tworzenie wydawnictw prezentujących wyniki symulacji w postaci wykresów lub tabel, • *SAVE*, *DUMP* - zapamiętywanie w pliku wyników symulacji i programu, • *FINISH* - koniec dyrektyw systemu wykonawczego.

4. Przykładowy program

W tym rozdziale przedstawiony zostanie przykład będący praktyczną ilustracją modelowania i symulacji komputerowej systemów ciągłych w języku *Godys-PC*. Na początku przeprowadzimy analizę problemu, w wyniku której utworzymy model matematyczny badanego systemu. W drugiej części rozdziału przedstawiony jest kompletny program w języku *Godys-PC* napisany na podstawie opisanego wcześniej

modelu matematycznego.

Model matematyczny badanego systemu. Jako symulowany system rozważmy przedstawiony na rys.1 układ mechaniczny, składający się z ciała o *masie* M , na którą działa *sila* F , sprężyny o *współczynniku sprężystości* K oraz tłumika o *współczynniku tłumienia* D .



Rys.1. Układ mechaniczny składający się z masy, sprężyny i tłumika

Przyjmując oznaczenia: x - *położenie masy*, v - *prędkość masy*, a - *przyspieszenie masy*, F_k - *sila sprężystości sprężyny*, F_D - *sila tłumiąca tłumika*, utworzymy model matematyczny układu z rys.1.

Sila sprężystości F_s w ściskanej lub rozciąganej sprężynie ma zwrot przeciwny do przesunięcia masy, a siła tłumiąca F_D tłumika ma zwrot przeciwny do przesunięcia masy (1).

$$F_k = -K \cdot x, \quad F_D = -D \cdot v \quad (1)$$

Druga zasada Newtona, zastosowane do rozważanego układu, ma postać: $F + F_k + F_D = M \cdot a$.

Podstawiając: $v = dx/dt$ i $a = dv/dt = d^2x/dt^2$ otrzymujemy (2)

$$M \cdot \frac{d^2x}{dt^2} + D \cdot \frac{dx}{dt} + K \cdot x = F \Rightarrow \frac{d^2x}{dt^2} = -\left(\frac{K}{M}x + \frac{D}{M}v - \frac{F}{M}\right) \quad (2)$$

będące modelem matematycznym układu z rys.1. Równanie (2) można przedstawić w postaci układu dwu równań 1 - rzędu (3).

$$\dot{x} = v, \quad \dot{v} = -\frac{K}{M}x - \frac{D}{M}v + \frac{F}{M} \quad (3)$$

Zapis modelu w języku Godys-PC. W trakcie eksperymentu symulacyjnego będziemy badali zachowanie się układu mechanicznego, przedstawionego na rys.1, dla siły wymuszającej $F(t) = I(t-t_0)$. Jako model matematyczny przyjmijmy układ równań (2). Zapisany w języku *Godys-PC* model układu z rys.1 ma następującą postać:

```

model SPREZYNA
  prepare x, v
  paramt k, d, m, a1, t0

  dynamic
  x = INTEG (v; 0)
  v = INTEG((a1/m)*STEP(t - t0) - (d/m)*v - (k/m)*x; 0)
  end

load SPREZYNA
data k = 1, d = 0.3, m = 1, a1 = -1, t0 = 10
execute (dt = 0.1, tmax = 50, comdel = 0.1)
hdr      " Układ oscylacyjny drugiego rzędu", ""
plotxy (t = (0, 50), x)
plotxy (t, v)
plotxy (x, v)
data t0 = 5, d = 0.7
execute (tmax = 30, comdel = 0.5)

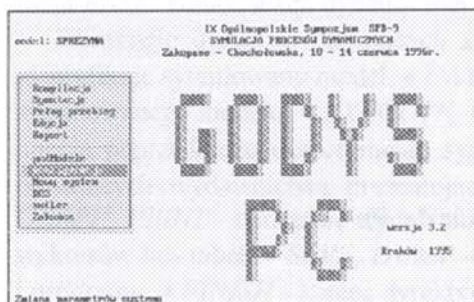
```

```
print (x, v)
prplot (x, v)
graph (x)
finish
```

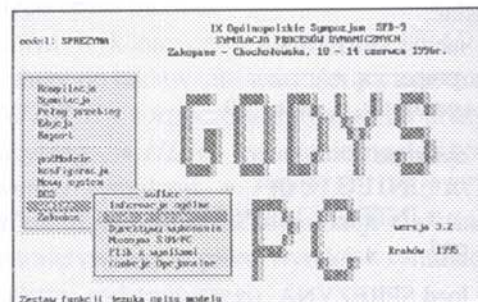
5. Menu systemu Godys-PC

Godys-PC pracuje na mikrokomputerach typu *PC* w środowisku systemu *DOS*. System sterowany jest za pomocą prostego rozwijalnego menu. Menu to może być obsługiwane zarówno za pomocą klawiatury, jak też z wykorzystaniem myszki. Zwięzła semantyka aktualnej opcji menu jest opisana w dolnej linii ekranu. Tekst menu wczytywany jest z pliku tekstowego. Użytkownik może za pomocą dowolnego edytora *ASCII* zmienić zawartość tego pliku i utworzyć w ten sposób np. angielską wersję językową menu.

Menu systemu *Godys-PC* przedstawia rys.2. *Poszczególne opcje* mają następujące znaczenie: • *Kompilacja* - kompilacja aktualnie wybranego modelu (w tym przypadku jest to model o nazwie *SPREZYNA*), • *Symulacja* - rozpoczęcie procesu symulacyjnego aktualnie wybranego modelu (model ten musi być wcześniej skompilowany), • *Pełny przebieg* - automatyczne wykonanie po kolei opcji "*Kompilacja*" i "*Symulacja*", • *Edycja* - wywołanie edytora *ASCII* (może to być standardowy edytor systemu *Godys-PC* lub dowolny wskazany przez użytkownika), • *Raport* - przeglądanie raportu z przebiegu kompilacji oraz kolejnych eksperymentów symulacyjnych, • *podModele* - wywołanie preprocesora podmodeli, • *konFiguracja* - wywołanie pod-menu, w którym modyfikowane są parametry systemu, • *Nowy system* - opcja ta umożliwi utworzenie nowego systemu wykonawczego z dołączonymi własnymi funkcjami użytkownika zrealizowanymi w postaci podprogramów w języku *Fortran*, • *DOS* - tymczasowe wyjście do systemu operacyjnego (*DOS shell*), • *sufLer* - wywołanie podmenu (rys.3) z systemem podpowiedzi dla użytkownika języka *Godys-PC*, • *Zakończ* - wyjście z menu (zakończenie pracy).



Rys.2. Menu główne systemu Godys-PC



Rys.3. Menu "sufLer"

W czasie pracy z systemem *Godys-PC* użytkownik może korzystać z różnych podpowiedzi. Naciśnięcie klawisza *F1* spowoduje wyświetlenie zwięzłej jednoekranowej informacji o tym jak można na bieżąco sterować przebiegiem procesu symulacyjnego. Bardziej szczegółowe podpowiedzi dostępne są w menu "*sufLer*" przedstawionym na rys.3. W menu tym możemy: a) dowiedzieć się gdzie szukać pomocy gdy

mamy problemy, *b*) zobaczyć jaką postać mają podstawowe funkcje języka dostępne w opisie modelu, *c*) oglądnąć przykłady użycia dyrektyw sterujących procesem symulacji, *d*) dowiedzieć się jak wygląda struktura programu wynikowego, *e*) zapoznać się ze strukturą pliku z wynikami symulacji, *f*) poznać sposób konstruowania i dołączania do języka własnych funkcji.

Menu systemu *Godys-PC* podobnie jak wiele programów (m.in. *Norton Commander*) oszczędza kineskop monitora. W przypadku gdy użytkownik przez dłuższy czas nie dotyka klawiatury lub nie manipuluje myszką uruchamiany jest automatycznie wygaszacz ekranu.

6. Preprocesor podmodeli

Preprocesor podmodeli przetwarza tekst źródłowy programu zapisanego w rozszerzonym języku opisu modelu do równoważnej mu postaci akceptowanej przez kompilator języka *Godys-PC*. Rozszerzenie polega na wprowadzeniu do języka możliwości definiowania tzw. podmodeli. Podmodel może być traktowany jako wielowyjściowa funkcja (czego nie ma język *Godys-PC*). Opis podmodelu jest praktycznie makrodefinicją. W opisie podmodelu mogą być używane wszystkie funkcje języka *Godys-PC*, jak też wcześniej zdefiniowane podmodele. Preprocesor podmodeli może korzystać z utworzonej wcześniej biblioteki podmodeli, a także może tę bibliotekę aktualizować. Użytkownik może tworzyć różne biblioteki podmodeli, ale w danym przebiegu preprocesora dostępna jest tylko jedna biblioteka.

7. Uwagi końcowe

Przedstawiony w referacie język *Godys-PC* oraz jego wcześniejsze wersje używane są od wielu lat w dydaktyce i pracach badawczych na uczelniach środowiska krakowskiego oraz w *Politechnice Warszawskiej*. Wieloletnie doświadczenie pozwala wnioskować, że jest on prostym i wygodnym narzędziem. Dodatkową zaletą systemu *Godys-PC* są bardzo skromne wymagania sprzętowe. W odróżnieniu od używanego obecnie "pamięciożernego" oprogramowania pracującego pod kontrolą systemu operacyjnego MS Windows, pełna wersja języka *Godys-PC* zajmuje niecałe 800KB pamięci dyskowej i do sprawnego działania wymaga około 1MB wolnej przestrzeni na dysku twardym. System będzie poprawnie działał nawet na mikrokomputerze *IBM PC/XT* z koprocesorem, pamięcią *RAM 512KB* i kartą graficzną *Hercules, EGA* lub *VGA*.

Ze względu na poważne ograniczenia materiałów konferencyjnych nie było możliwe pełne przedstawienie możliwości języka. Dokładny raport języka *Godys-PC*, wraz z licznymi przykładami symulowanych modeli, przedstawiony jest w [5]. Dodatkowe informacje na temat języka można także uzyskać od autorów: *E-mail kuras@ii.uj.edu.pl*.

Literatura

- [1] J. Król, J. Kuraś, J. Lembas: *Język symulacyjny dla komputerów Odra 1300*. Informatyka 1979, nr 11, ss.16-17.
- [2] J. Król, J. Kuraś, J. Lembas, M. Ślusarek: *Implementacja języka do symulacji układów ciągłych ze zdarzeniami dyskretnymi*. Podstawy sterowania, t.10 (1980), z.1, pp.57-68.
- [3] J. Król, J. Kuraś, J. Lembas, J. Rosek, M. Ślusarek: *Godys-4: an easy to Use Language for the Simulation of Continuous Systems with Discontinuities*. Proc. International AMSE Conference Modelling & Simulation, Paris-Sud 1982, vol.3, Group 3, Simulation Methods, pp.15-17, Imprime par l'AMSE, ed. G.Mesnard.
- [4] J. Kuraś, J. Lembas, M. Skomorowski: *Język symulacji systemów o działaniu ciągłym ze zdarzeniami dyskretnymi*. Informatyka 1989, nr 7, ss.9-11.
- [5] J. Kuraś, J. Lembas, M. Skomorowski: *Wstęp do symulacji komputerowej systemów ciągłych*. Skrypt Uniwersytetu Jagiellońskiego, Kraków 1995.