

PAKIET PROGRAMOWY DO CELÓW IDENTYFIKACJI MODELI DYNAMICZNYCH SIGNAL IDENTIFICATION

Niniejsza praca przedstawia oprogramowanie służące do szeroko rozumianej identyfikacji modeli dynamicznych. Obejmuje ono swym zakresem cały proces identyfikacji począwszy od wstępnej analizy i przetwarzania danych, poprzez estymację i weryfikację modelu, a kończąc na jego analizie. Opisano zaimplementowane typy modeli liniowych, nieliniowych, rozmytych i neuronowych, algorytmy stosowane podczas ich identyfikacji oraz strukturę programu. Zwrócono uwagę na możliwości rozbudowy, elastyczność i prostotę obsługi programu, pokazano możliwe zastosowania.

SOFTWARE PACKAGE FOR IDENTIFICATION FOR DYNAMIC MODELS SIGNAL IDENTIFICATION

The paper describes a software package for a broadly understanding dynamic models identification. The software takes over all identification problems, starting at initial data analysis, through estimation model parameters and verification, up to model analysis. Implemented types of linear, non-linear, fuzzy and neural network models were described. The package offers a user-friendly interface and the possibility to expand its functionality with user written modules.

1. WPROWADZENIE

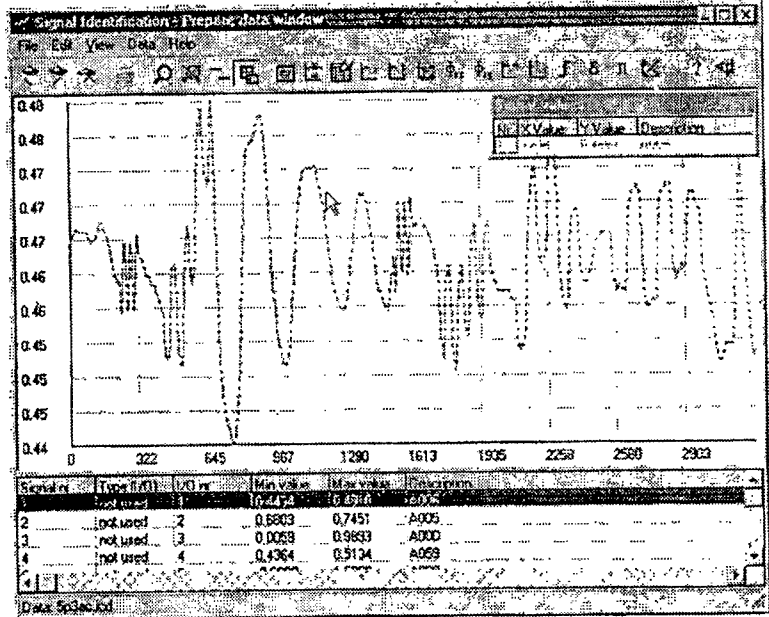
Identyfikacja dynamicznych modeli obiektów sterowania przy wykorzystaniu metod stochastycznych jest zagadnieniem szeroko znanym. Przydatność jej wyników dla celów syntezy układów regulacji jest niekwestionowana. Identyfikacja obiektu może być wykonana przy pomocy różnych metod, poczynając od najstarszej i najlepiej znanej estymacji parametrów transmitancji liniowej ciągłej $G(s)$ lub dyskretnej $G(z)$ obiektu, poprzez bardziej wyrafinowane metody identyfikacji modeli nieliniowych parametrycznych, kończąc na opracowanych w ostatnich latach modelowaniu z wykorzystaniem formalizmu rozmytego czy też sztucznych sieci neuronowych. Każda z wymienionych metod ma swoje wady i zalety, dla każdej z nich istnieje oprogramowanie umożliwiające jej zastosowanie w praktyce oraz bogata literatura [1], [2], [5], [6], [7]. Nie istnieje jeden spójny, łatwy w obsłudze program łączący wszystkie wymienione metody i umożliwiający przeprowadzenie pełnego cyklu identyfikacji modelu, począwszy od przetworzenia danych uczących, a skończywszy na weryfikacji. Lukę tą może wypełnić pakiet *Signal Identification* (SI). Jest on programem pracującym w 32-bitowym środowisku MS Windows 95 / NT / 98. SI daje użytkownikowi możliwość przeprowadzenia pełnej procedury identyfikacji modelu obiektu sterowania. Zgodnie z filozofią wybranego systemu operacyjnego umożliwia on równoległą pracę z

innymi programami, interfejs, poprzez który program komunikuje się z użytkownikiem, jest nazywany „przyjaznym” dla użytkownika, program został wyposażony w system pomocy kontekstowej dostępnej w każdej chwili itp. Program komunikuje się z użytkownikiem (menu, opisy, komunikaty widoczne na ekranie) w języku angielskim.

Opis możliwości programu zostanie podzielony na grupy funkcji odpowiadające bezpośrednio poszczególnym modułom.

2. PRZETWARZANIE WSTĘPNE DANYCH

Poprzez wstępne przetwarzanie danych rozumie się wszystkie operacje na danych pochodzących z procesu przed udostępnieniem ich procedurze identyfikującej czyli odczyt z dysku, przeprowadzenie ich analizy (wyświetlanie sygnałów wraz z możliwościami ich porównania, powiększenia czy też dokładnego odczytu wartości, wyświetlanie gęstości widmowej mocy, wyświetlanie korelacji wzajemnej dwóch sygnałów oraz autokorelacji) oraz różnorodnych przekształceń. Moduł wstępnej obróbki danych pakietu SI umożliwi usunięcie z sygnałów wartości średniej, trendów liniowego i kwadratowego, filtrację sygnałów za pomocą różnych filtrów, pracujących zarówno w dziedzinie częstotliwości (przy wykorzystaniu funkcji cepstrum) jak i czasu (filtry wagowe), całkowanie oraz różniczkowanie cyfrowe połączone z jednoczesną filtracją, agregację sygnałów oraz ich przetwarzanie za pomocą wybranej funkcji matematycznej wprowadzanej przez użytkownika.



Rys 1. Okno wstępnego przetwarzania danych.

3. IDENTYFIKACJA

Pakiet służy do identyfikacji modeli parametrycznych (model obiektu jest tworzony poprzez optymalizację określonego kryterium na podstawie danych pochodzących z procesu tak, aby opisywał on możliwie najdokładniej obiekt). Pakiet dopuszcza wykorzystanie struktury modelu uwzględniającej w celu obliczenia estymowanej wartości wyjścia w chwili k zarówno wartości sygnałów wejściowych u , jak i wartości sygnału wyjściowego z chwil wcześniejszych oraz estymowanego sygnału błędu e . Przy konstrukcji programu przyjęto, że podstawową strukturą będzie model typu *MISO* (wiele wejść - jedno wyjście). Program daje użytkownikowi możliwość wyboru typu modelu spośród dostępnych klas:

- Modele liniowe parametryczne w dziedzinie czasu dyskretnego opisane równaniami różnicowymi typu *AR*, *ARX*, *FIR*, *MAX*, *ARMA*, *ARMAX*. Wyżej wymienione modele można identyfikować przy użyciu estymatorów *LS* (najmniejsza suma kwadratów błędu, ang. *Least Square*), *ELS* (rozszerzona metoda najmniejszych sumy kwadratów, ang. *Enhanced Least Square*), *IV* (metoda zmiennej instrumentalnej, ang. *Instrumental Variable*), *ML* (największej wiarygodności, ang. *Maximum Likelihood*), oraz *LSA* (najmniejsza suma wartości bezwzględnych błędów *Least Sum of Absolute*) korzystając z algorytmu rekursywnego lub *LS*, *ELS* dla algorytmu wsadowego. Dokładny opis równań różnicowych oraz pierwszych czterech estymatorów można znaleźć w pracy [3], natomiast estymator *LSA* został opisany w pracy [4].

- Modele parametryczne nieliniowe względem zmiennych dla czasu dyskretnego opisane równaniem w postaci:

$$y(k) = \alpha_0 + \sum_{i=0}^M a_i v_i(k) + \sum_{i=0}^M \sum_{j=0}^M a_{ij} v_i(k) v_j(k) + \sum_{i=0}^M \sum_{j=0}^M \sum_{l=0}^M a_{ijl} v_i(k) v_j(k) v_l(k), \quad (1)$$

- gdzie $v_i(k)$ jest dowolnym wejściem bądź wyjściem z chwili k lub wcześniejszych lub *arcusem tangensem* następujących zmiennych:

$$v_i \in \left\{ \begin{array}{l} y(k-1), y(k-2), \dots, y(k-r) \\ u_1(k-d_1), \dots, u_1(k-d_1-r) \\ \dots \\ u_n(k-d_n), \dots, u_n(k-d_n-r) \end{array} \right\}, \quad (2)$$

wyznaczane przy pomocy estymatora *LS*.

- Modele liniowe parametryczne w dziedzinie czasu dyskretnego o współczynnikach rozmywanych według opisu *TSK* (*Takagi-Sugeno-Kanga*) w postaci

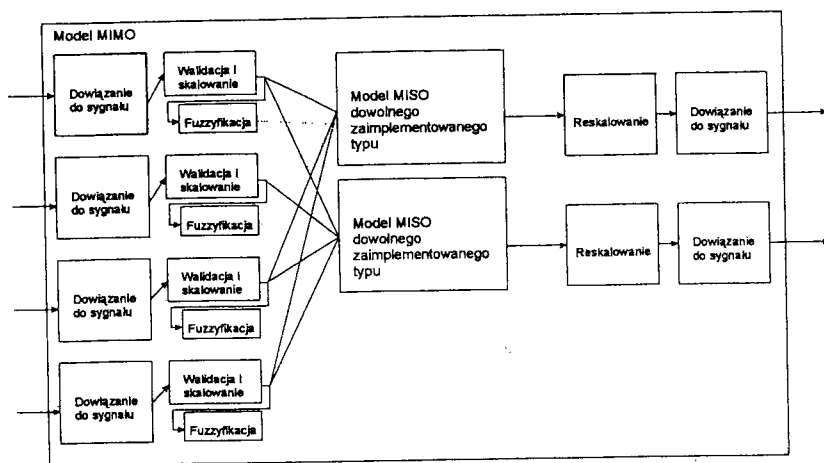
$$\sum_{i=0}^n (A_i \cdot \Phi_i(k)) \cdot y(k) = \sum_{i=0}^n (B_i \cdot \Phi_i(k)) \cdot u(k), \quad (3)$$

gdzie: $\Phi_i(k)$ - funkcja przynależności wejść do zdefiniowanych zbiorów rozmytych. Wyznaczanie ich jest możliwe przy wykorzystaniu estymatorów *LS*, *ELS*, *LSA*.

- Modele neuronowe w dziedzinie czasu dyskretnego - sztuczna sieć neuronowa wielowarstwowa jednokierunkowa, uczona poprzez minimalizację kryterium *LS*. Polecenia obsługi i obliczeń sieci dają możliwość określenia liczby neuronów poszczególnych warstw, wybór zadania sieci (modelowanie lub predykcja) czy też

algorytmu uczenia. Jako podstawową metodę uczenia zastosowano algorytm gradientów sprzężonych, ucieczki z minimów lokalnych są przeprowadzane zgodnie z algorytmem symulowanego wyżarzania, a do inicjacji wag można zastosować algorytm genetyczny lub symulowanego wyżarzania.

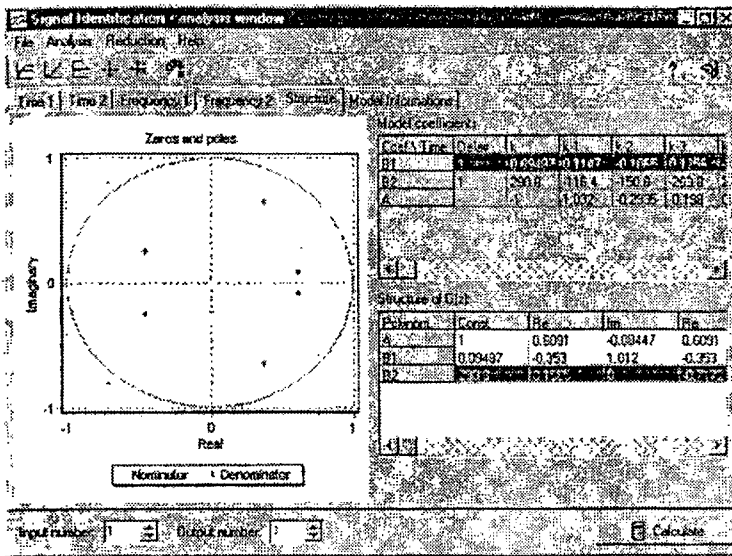
Model dla wielu wyjść (*MIMO*) składa się z wielu modeli typu *MISO*, przy czym możliwe jest ich dowolne łączenie, tzn. wyjście 1 może być identyfikowane jako model liniowy, drugie jako model neuronowy, trzecie jako rozmyte itd... Podczas identyfikacji automatycznie tworzone są również modele autoregresyjne sygnałów wejściowych, przydatne przy przeprowadzaniu predykcji na więcej niż jeden krok. Na rysunku 2 przedstawiono ogólną strukturę modelu wielowymiarowego stosowaną przez pakiet.



Rys 2. Ogólna struktura funkcjonalna modelu *MIMO*

4. ANALIZA MODELU

Następny moduł SI udostępnia podstawowe funkcje analizy modelu. Do tej grupy należy zaliczyć funkcje wyznaczające tzw. aproksymację $3P$ – przybliżony opis za pomocą modelu inercyjnego 1-go rzędu z opóźnieniem, o parametrach: wzmocnienie k , zastępcze opóźnienie T_d , zastępcza stała czasowa T_c , funkcje odpowiedzi modelu na impuls jednostkowy, skok jednostkowy czy dowolne zakłócenie deterministyczne określone przez użytkownika. Ponadto dla modeli liniowych, zarówno występujących osobno, jak i w formie poszczególnych składników modelu rozmytego, dostępna jest możliwość wykreślenia wykresów *Bodego* i *Nyquista*, wyznaczanie biegunów i zer transmitancji modelu oraz przeprowadzenie redukcji rzędu modelu i aproksymacji pomiędzy czasem dyskretnym a ciągłym. Aproksymacja i redukcja rzędu modelu jest przeprowadzana na podstawie dopasowywania charakterystyki widmowej obiektu, tzn.: w pierwszym kroku algorytmu obliczana jest charakterystyka widmowa modelu wzorcowego, następnie są dobierane współczynniki modelu o niższym, zadanym rzędzie tak, aby jego charakterystyka widmowa jak najmniej różniła się od pierwotnej w sensie błędu średniokwadratowego. Błąd ten jednocześnie może stanowić miarę jakości przeprowadzonego przekształcenia. Otrzymany w opisany sposób model powinien mieć zbliżone właściwości do modelu wzorcowego.



Rys 3. Przykładowy wynik analizy – zera i bieguny modelu liniowego

5. WERYFIKACJA

Moduł weryfikacji pozwala na przeprowadzenie modelowania odpowiedzi obiektu na podstawie wcześniej zebranych danych. SI umożliwia przeprowadzenie obliczeń przy pomocy dwu algorytmów: przyjmując, że wartość wyjścia z chwil wcześniejszych jest pobierana z procesu (weryfikacja standardowa) lub że sygnał wyjścia w chwilach wcześniejszych także został obliczony na podstawie modelu (tzw. *model reference simulation*). Modelowanie odpowiedzi może odbywać się z zadaniem dowolnym wyprzedzeniem (standardowo 1 krok). Ponadto program umożliwia automatyczne sprawdzenie grupy modeli na kilku zestawach danych (weryfikacja krzyżowa) i wybór najlepszego ze względu na kryterium *FPE* lub inne, wprowadzane przez użytkownika. Użytkownik może formułować własne, znacznie bardziej uogólnione wskaźniki oceny modelu i na ich podstawie wybierać optymalny w przyjętym sensie.

6. BUDOWA PROGRAMU

Pakiet został zbudowany jako zestaw praktycznie niezależnych modułów, przy pełnym wykorzystaniu programowania obiektowego. Ułatwia to dalszą rozbudowę, zarówno już istniejących części np. o nie zaimplementowane jeszcze grupy modeli czy o nowe klasy filtrów, jak i o całkowicie nowe moduły, np. syntezy regulatora. Język wykorzystany podczas jego tworzenia to C++, przy czym starano się zachować maksymalną zgodność ze standardem ANSI języka, dzięki czemu część *stricte* obliczeniowa może zostać bezproblemowo przeniesiona na komputery o większej mocy obliczeniowej, lub pracujące pod kontrolą innego systemu operacyjnego. Wymagania programu – komputer klasy PC oparty na procesorze Pentium lub następcach, 16 MB RAM, system operacyjny MS Windows 95 / 98 / NT. Wybrany w celu uruchomienia systemu kompilator jest częścią zintegrowanego pakietu szybkiego tworzenia aplikacji RAD (*Rapid Application Development*) C++ Builder 4.0 firmy Borland Inc.

7. PODSUMOWANIE

Opisywany pakiet stanowi aktualnie pełen szkielet pozwalający na łatwą implementację i testowanie nowo opracowywanych algorytmów identyfikacyjnych i innych, związanych z nimi. Dzięki niemu można skupić się na merytorycznej części zadania. W ten sposób np. został zaimplementowany i przetestowany algorytm inteligentnego łączenia modeli liniowych i rozmytych o tej samej strukturze identyfikowanych na bazie innych serii danych. Wspólne funkcje analizy i weryfikacji ułatwiają porównanie różnych typów modeli dla poszczególnych obiektów. Jednolity interfejs użytkownika, zastosowanie w każdym z wykorzystanych rodzajów opisu możliwie największej liczby wspólnych pojęć i określeń oraz zapewnienie wartości domyślnych wszystkich parametrów ułatwiają pracę z programem osobom nie będącym specjalistami w dziedzinie identyfikacji. Dzięki wymienionym powyżej cechom program można zastosować wszędzie tam, gdzie konieczne jest szybkie utworzenie modelu obiektu sterowania. Prosta obsługa i szeroki wachlarz możliwości, uzupełnione obszernym opisem dostępnym interaktywnie jako pomoc (*Help*), ułatwiają wykorzystanie pakietu w obliczeniach inżynierskich oraz dydaktyce.

8. PODZIĘKOWANIA

Obszerne fragmenty pakietu SI zostały przygotowane w ramach programu uczelnianego CATID.

9. LITERATURA

- [1] Dokumentacja: „*System Identification Toolbox User's Guide*”, MathWorks Inc. 1994.
- [2] Driankov D., Hellendoorn H., Reinfrank M.: „*Wprowadzenie do sterowania rozmytego*”, Wydawnictwa Naukowo-Techniczne, Warszawa 1996.
- [3] Janiszowski K.: „*Podstawy wyznaczania opisu i sterowania obiektów dynamicznych*”, WPW, Warszawa 1991.
- [4] Janiszowski K.: „*To estimation in sense of the least sum of absolute errors*”, *Proceedings of the Fifth International Symposium on Methods and Models in Automation and Robotics*, Międzyzdroje 25 – 29 August 1998, Wydawnictwo Uczelniane Politechniki Szczecińskiej, Szczecin 1998.
- [5] Ljung L.: „*System Identification. Theory for the user*”, Prentice Hall, New Jersey 1987.
- [6] Rutkowska D., Piliński M., Rutkowski L.: „*Sieci neuronowe, algorytmy genetyczne i systemy rozmyte*”, Wydawnictwo Naukowe PWN, Warszawa-Lódź 1997.
- [7] Soderstrom T., Stoica P.: „*Identyfikacja systemów*”, PWN, Warszawa 1997.
- [8] Wnuk P.: „*Opracowanie pakietu programowego dla celów identyfikacji modeli dynamicznych*”, Praca Dyplomowa, Instytut Automatyki i Robotyki Politechniki Warszawskiej, Warszawa 1998.