

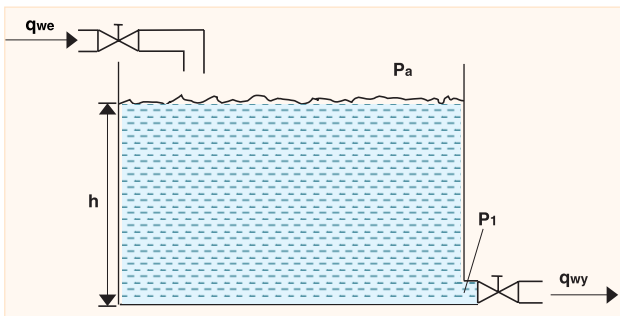
# Zastosowanie programu SIMIT SCE do modelowania właściwości obiektu i połączenia z fizycznym sterownikiem PLC

Artur Król  
Joanna Moczko-Król \*

Rozwiązując problem sterowania procesem fizycznym z użyciem sterownika PLC, należy sporządzić przede wszystkim odpowiedni opis obiektu. Automatyzacja szuka zatem matematycznego modelu w postaci układu zależności zachodzących w procesie, którego rozwiązanie dawałoby wyniki zgodne z obserwacjami.

Współczesne środowisko wspomagania projektowania układów automatyki pomaga użytkownikom w budowie i testowaniu złożonych algorytmów sterowania oraz daje możliwość symulacyjnej weryfikacji pracy zaprojektowanego układu.

Przeprowadźmy analizę układu rozmytego sterowania poziomem cieczy (rys. 1).



Rys. 1. Zbiornik z wodą

Strumień masy (masowe natężenie przepływu) na wyjściu zbiornika można zapisać następującym wzorem:

$$q_{wy} = \frac{k_v}{\sqrt{\rho}} \sqrt{(p_1 - p_a)} \quad (1)$$

$p_1$  – ciśnienie przed zaworem,

$p_a$  – ciśnienie zewnętrzne,

$\frac{\sqrt{\rho}}{k_v}$  – stała zaworu,

$\rho$  – gęstość cieczy.

Stosując zasadę zachowania masy, otrzymamy nieliniowe równanie różniczkowe dla wysokości słupa cieczy:

\* mgr inż. Artur Król – UDT O/Opole,  
mgr inż. Joanna Moczko-Król  
– ECO Opole

$$\dot{h} = \frac{1}{A\rho} (q_{we} - q_{wy}) \quad (2)$$

$q_{we}$  – strumień masy na wejściu układu,

$A$  – powierzchnia przekroju poprzecznego zbiornika.

Do analizy takiego układu wykorzystaliśmy metodę *hardware-in-the-loop* [3] polegającą na podłączeniu, w naszym przypadku, sterownika PLC do komputera, w którym wykonywany jest program symulujący zachowanie obiektu w czasie rzeczywistym.

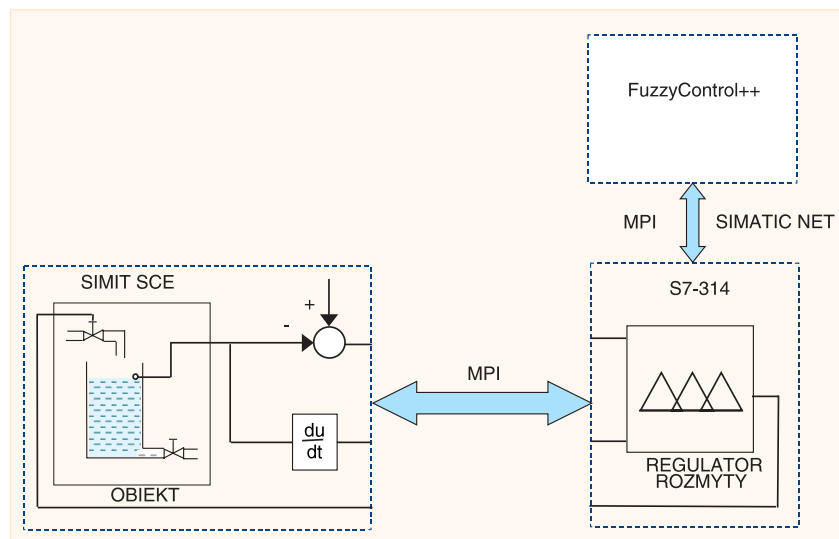
W pierwszej fazie realizacji projektu określiliśmy i sparymetryzowaliśmy podstawowe bloki układu sterowania (rys. 2):

- model obiektu zrealizowany w programie symulacyjnym SIMIT SCE
- regulator zrealizowany za pomocą fizycznego sterownika S7-314.

## Modelowanie właściwości obiektu

Intensywny rozwój zastosowań wszelkiego rodzaju pakietów programowych wspomagających projektowanie układów automatyki zmusił wiele firm, zajmujących się dotychczas produkcją wyłącznie elementów automatyki, do tworzenia programów symulacyjnych.

Jedną z takich ciekawych ofert jest oprogramowanie SIMIT SCE proponowane przez firmę SIEMENS. Jest to pakiet stworzony z myślą o testowaniu różnego rodzaju



Rys. 2. Schemat blokowy układu sterowania

ju układów regulacji przez kadrę inżynierską oraz naukowe grupy badawcze. Pakiet ten pozwala użytkownikowi na:

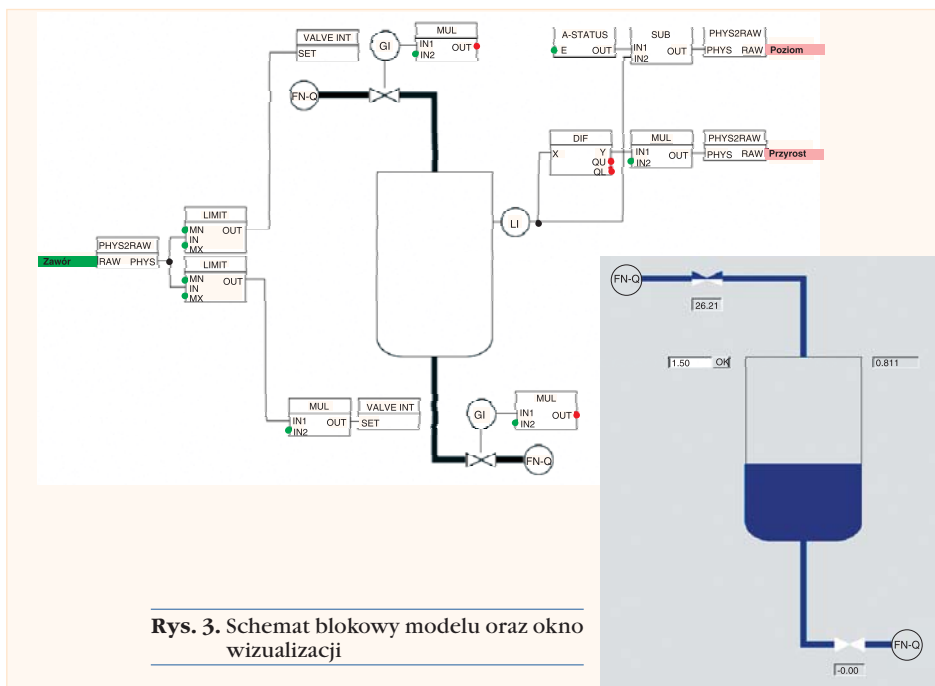
- modelowanie obiektu
- testowanie całego układu automatyki w czasie rzeczywistym
- połączenie z rzeczywistym sterownikiem
- możliwość stworzenia specjalnej nakładki graficznej pozwalającej na zmianę parametrów obiektu
- wizualizację zmiennych dynamicznych
- połączenie z innymi programami
- wykorzystanie bogatej biblioteki gotowych elementów
- tworzenie własnych bibliotek z elementami zdefiniowanymi za pomocą wzorów matematycznych.

Przed rozpoczęciem procesu symulacji program SIMIT SCE wymaga, aby użytkownik zaprojektował tzw. plan, czyli model obiektu, całego układu regulacji lub procesu fizycznego (rys. 3). Tworzy się go przy wykorzystaniu elementów z bibliotek, definiując dodatkowo np. sposób komunikacji z rzeczywistym sterownikiem PLC lub programem pracującym w środowisku Windows. Następnie, na podstawie tak stworzonego planu można wygenerować graficzne okno, dzięki któremu użytkownik wpływa (zmienia), w trakcie pracy *on line*, wybrane wartości parametrów obiektu lub całego układu automatyki (rys. 3). Dodatkowo istnieje możliwość zdefiniowania zmiennych, które dzięki specjalnemu modułowi mogą być wyświetlane i wykreślane w trakcie symulacji w czasie rzeczywistym.

Tak stworzony projekt zostaje następnie poddany procesowi kompilacji, w wyniku czego otrzymujemy specjalne pliki, które umożliwiają jego wykonanie w środowisku Windows w czasie rzeczywistym.

W naszym przykładzie projekt stworzony w programie SIMIT SCE wymagał zdefiniowania następujących elementów (rys. 3):

- modelu zbiornika scharakteryzowanego przez:
  - pojemność
  - pole powierzchni zbiornika
  - sposób ustawienia zbiornika
  - ciśnienie zewnętrzne
  - początkową temperaturę cieczy w zbiorniku
  - początkowy poziom cieczy
- zaworów opisanych przez:
  - charakterystykę zaworu
  - parametry przepływu
- elementów wymuszających przepływ cieczy
- elementów umożliwiających komunikację z zewnętrznym sterownikiem PLC



Rys. 3. Schemat blokowy modelu oraz okno wizualizacji

- elementów umożliwiających pomiar:
  - stopnia otwarcia zaworu
  - poziomu cieczy w zbiorniku
- elementów umożliwiających wykonanie potrzebnych obliczeń matematycznych.

Na podstawie schematu blokowego modelu wygenerowaliśmy okno wizualizacji procesu (rys. 3), które umożliwiło *on line*:

- graficzny podgląd stopnia otwarcia poszczególnych zaworów
- graficzny podgląd poziomu cieczy
- wprowadzanie wartości zadanej poziomu
- cyfrowy podgląd poziomu cieczy w zbiorniku
- wyświetlenie w formie cyfrowej stopnia otwarcia zaworów.

Dla tak przygotowanego schematu blokowego obiektu dodatkowo sparametryzowaliśmy moduł umożliwiający komunikację modelu ze sterownikiem PLC oraz moduł graficzny, który wykreślał przebiegi w interesujących nas punktach (rys. 5).

## Projektowanie regulatora rozmytego

W pierwszej fazie realizacji postawionego zadania zdefiniowaliśmy kryterium, jakie musi spełniać regulator. Założyliśmy, że przy skokowym otwarciu zaworu poziom cieczy w zbiorniku powinien bez przeregulowania osiągnąć poziom zadany.

Proces projektowania regulatora rozmytego (FLC) rozpoczęliśmy od przyjęcia jako zmiennych wejściowych regulatora: wysokości słupa cieczy  $h$  i zmiany  $\Delta h$ , a jako zmiennej wyjściowej: stopnia otwarcia zaworu  $k$ .

W następnym kroku wyznaczyliśmy przestrzenie oraz funkcje przynależności zbiorów rozmytych związanych z etykietami lingwistycznymi dla wybranych zmiennych.

Dla  $h$  i  $\Delta h$  przestrzenie dobraliśmy w przedziale  $(0, 2)$ , a dla  $k$  w przedziale  $(0, 100)$ .

Natomiast funkcje przynależności zbiorów rozmytych są związane ze zbiorami terminów: wysokość słupa cieczy, zmiany poziomu cieczy i stopień otwarcia zaworu. Zbiory terminów podzieliliśmy na następujące etykiety lingwistyczne:

- dla  $h$  – „niski **N**”, „odpowiedni **OK**”, „wysoki **W**”
- dla  $\Delta h$  – „mały **M**”, „bez zmian **BZ**”, „duży **D**”
- dla  $k$  – „zamknij szybko **ZS**”, „zamknij wolno **ZW**”, „nie zmieniaj **NZ**”, „otwórz wolno **OW**”, „otwórz szybko **OS**”.

W opisanym przykładzie funkcja wiążąca zależności  $h$ ,  $\Delta h$  i  $k$  jest wyrażona za pomocą modelu rozmytego i została przez nas wyznaczona na podstawie wiedzy eksperta [7].

Zbiór doświadczeń eksperta odzwierciedlają następujące rozważania:

- jeśli poziom cieczy w zbiorniku jest równy zadaneemu, to nie należy zmieniać stopnia otwarcia zaworu
- jeśli poziom cieczy w zbiorniku jest niższy od zadaneemu, to należy otworzyć maksymalnie zawór
- jeśli poziom cieczy w zbiorniku jest wyższy od zadaneemu, to należy zamknąć zawór
- jeśli poziom cieczy w zbiorniku jest bliski zadaneemu, a jego przyrost jest duży, to należy zawór zamykać powoli
- jeśli poziom cieczy w zbiorniku jest bliski zadaneemu, a jego przyrost jest mały, to należy zawór otwierać powoli.

Na podstawie tych rozważań otrzymaliśmy część wiedzy o systemie w postaci zbioru reguł:

Reguła 1: JEŚLI ( $h = OK$ ), TO ( $k = NZ$ ).

Reguła 2: JEŚLI ( $h = N$ ), TO ( $k = OS$ ).

Reguła 3: JEŚLI ( $h = W$ ), TO ( $k = ZS$ ).

Reguła 4: JEŚLI ( $h = OK$ ) I ( $\Delta h = D$ ), TO ( $k = ZW$ ).

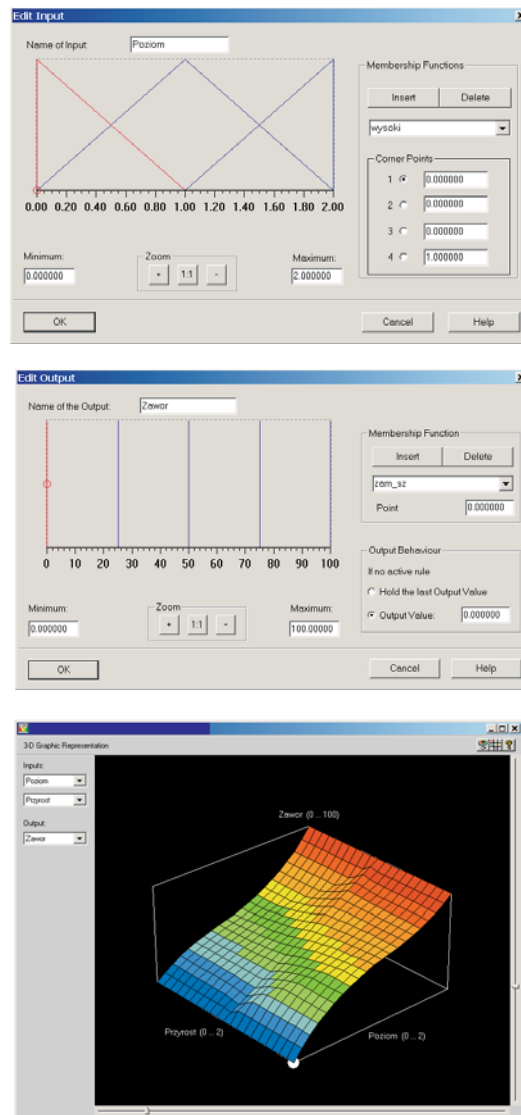
Reguła 5: JEŚLI ( $h = OK$ ) I ( $\Delta h = M$ ), TO ( $k = OW$ ).

Ponieważ zbiór powyższych reguł jest uboższy od mentalnego (zbiór zawierający wiedzę świadomą i podświadomą o systemie rzeczywistym), to wszystkie dodatkowe informacje muszą być uzupełnione na zasadzie domniemania (intuicji) przez kolejnego eksperta, tym razem eksperta modelowania rozmytego.

Jego zadaniem było:

- określenie rodzaju fuzyfikacji (określenie kształtu i parametrów funkcji przynależności)
- określenie na podstawie wyjściowych stopni przynależności tzw. wynikowej funkcji przynależności na podstawie ściśle zdefiniowanych elementów – bazy reguł, mechanizmów inferencji, funkcji przynależności dla wyjścia modelu
- określenie na podstawie wyjściowej funkcji przynależności ostrej wartości wyjścia
- dostrojenie modelu rozmytego.

Do budowy naszej struktury rozmytej wykorzystaliśmy program Fuzzy Control++ firmy SIEMENS [6]. Ponieważ wymusił on pewne ograniczenia, to w naszym zadaniu przyjęliśmy dla wejść  $h$  i  $\Delta h$  trójkątne funkcje przynależności i singletonowe funkcje dla wyjścia  $k$ , rozmieszczone w sposób pokazany na rys. 4.



Rys. 4. Funkcje przynależności dla  $h$  i  $k$  oraz wynikowa płaszczyzna decyzyjna regulatora FL

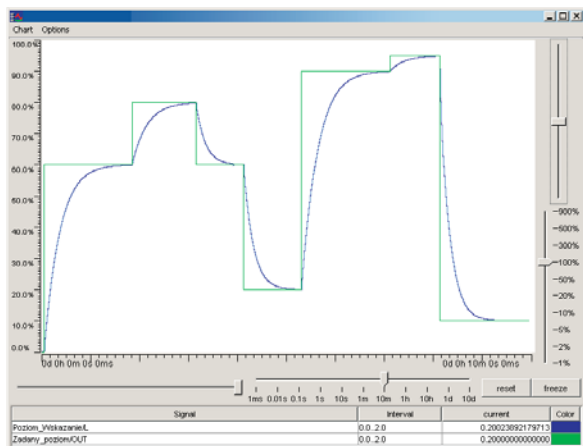
Do określenia ostrej wartości sygnału  $k$  mogliśmy wykorzystać jedynie metodę środka ciężkości, co w rezultacie dało płaszczyznę wiążącą wartości parametrów  $h$ ,  $\Delta h$  i  $k$  przedstawioną na rys. 4. Przy zastosowaniu singletonowych funkcji przynależności wyjścia znacznie uprościł się proces defuzyfikacji oparty na metodzie środka ciężkości. Jest to bardzo ważne, bo pozbywamy się wtedy dużego nakładu obliczeniowego związanego z całkowaniem powierzchni o nieregularnym kształcie.

Do fizycznej realizacji regulatora rozmytego wybraliśmy rzeczywisty sterownik SIMATIC z jednostką centralną S7-314. Następnie za pomocą oprogramowania Fuzzy Control++ skonfigurowaliśmy blok danych (DB30), który zawierał wszelkie informacje potrzebne do obliczeń dla specjalnego bloku funkcyjnego FB30. Blok ten zapisaaliśmy do sterownika, dzięki utworzonemu wcześniej kanałowi w SIMATIC NET S7.

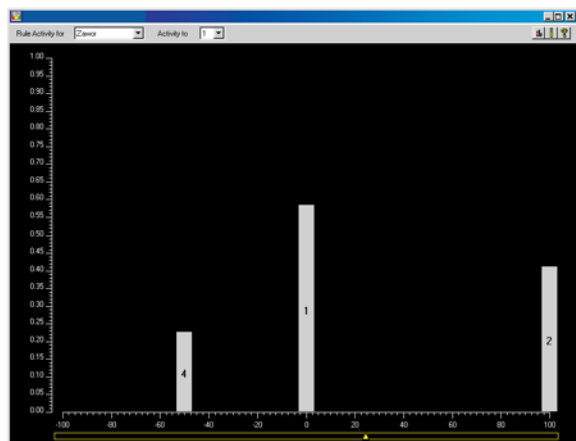
Komunikacja między programem SIMIT SCE a S7-314 odbywała się poprzez interfejs MPI, dzięki czemu zrealizowaliśmy podstawową cechę metody *hardware-in-the-loop*.

Po uruchomieniu symulacji w programie SIMIT SCE, dzięki jego możliwościom graficznym, mogliśmy obserwować zmiany poziomu cieczy w zbiorniku przy jednoczesnych zmianach poziomu zadanego. Jak widać na rys. 5 układ nie wykazuje przeregulowań, a więc zaprojektowany regulator umożliwił zrealizowanie założonego kryterium.

Dodatkowo możliwości programu Fuzzy Control++ pozwoliły na stały podgląd *on line* stopnia aktywacji poszczególnych reguł w procesie dochodzenia do wartości zadanej poziomu cieczy w zbiorniku (rys. 6).



Rys. 5. Przebieg zmian poziomu cieczy w zbiorniku



Rys. 6. Stopnie aktywacji poszczególnych reguł

## Podsumowanie

Zastosowanie metody *hardware-in-the-loop* w znaczny sposób wspomaga proces projektowania prostych, jak i zaawansowanych układów sterowania. Mając dodatkowo dobre oprogramowanie do symulacji wszelkiego rodzaju obiektów, metoda ta pozwala na ocenę działania całego układu regulacji. Ogranicza tym samym czas testowania w warunkach przemysłowych, a dla obiektów nietypowych umożliwia wstępny wybór strategii sterowania.

Dobry program symulacyjny powinien cechować:

- łatwość konstruowania schematów blokowych
- dostępność złożonych funkcji matematycznych, bez konieczności ich modelowania

- czytelność dużych schematów blokowych
- bogata firmowa biblioteka różnorodnych elementów pomocnych do modelowania obiektów, całych procesów technologicznych
- otwartość systemu wyrażająca się możliwością dołączania nowych bibliotek i korzystania z innych programów
- łatwość porównywania wykresów otrzymanych dla różnych wartości zmiennych
- standardowy interfejs do komunikacji z użytkownikiem
- dodatkowo – symulacja regulatora w sposób zapewniający reakcję na sygnały z rzeczywistego obiektu. Takie podejście do wspomaganie projektowania automatyki daje praktycznie nieograniczone możliwości, a jego zastosowanie może być w łatwy sposób rozszerzone na testowanie:
  - zaawansowanych algorytmów PID (z kompensacją opóźnienia, przestrajaniem wzmocnienia, samonastrajających się, wielowymiarowych)
  - regulatorów optymalnych, sterowania predyktywnego i adaptacyjnego
  - algorytmów sztucznej inteligencji (regulatorów wykorzystujących sztuczne sieci neuronowe, algorytmy genetyczne lub logikę rozmytą).

Wydaje się, że w dobie wszelkiego rodzaju specjalizowanych, zaawansowanych układów regulacji, współczesny projektant musi sięgać po niekonwencjonalne metody rozwiązywania skomplikowanych problemów automatyki.

## Bibliografia

1. Fuzzy Control++. User's Manual. SIEMENS AG 1998.
2. Kahlert J., Frank H.: Fuzzy-Logik und Fuzzy-Control. Vieweg & Sohn Verlag GmbH, Braunschweig/Wiesbaden 1994.
3. Kołek K., Sochacki M.: Hardware-in-the-loop w programowaniu sterowników PLC. PAR 12/1999 s. 5–8.
4. Kowal J.: Podstawy automatyki. Uczelniane Wydawnictwo Naukowo-Dydaktyczne. Kraków 2003.
5. Król A., Moczko-Król J.: S5/S7 Windows. Programowanie i symulacja sterowników PLC firmy SIEMENS. Wydawnictwo NAKOM, Poznań 2000.
6. Król A., Moczko-Król J.: Zastosowanie metody hardware-in-the-loop do budowy regulatora adaptacyjnego PI wykorzystującego logikę rozmytą w sterownikach PLC. PAR 1/2003 s. 8–11.
7. Piegat A.: Modelowanie i sterowanie rozmyte. Akademicka Oficyna Wydawnicza EXIT, Warszawa 1999.
8. Rojek R., Bartecki K., Korniak J.: Zastosowanie sztucznych sieci neuronowych i logiki rozmytej w automatyce. Skrypt nr 234 Politechniki Opolskiej. Opole 2000.
9. SIMIT BASIC. Bedienungshandbuch. SIEMENS AG 2003.
10. SIMIT SCE. Telestart Übungsaufgaben. SIEMENS AG 2003. ■