

Sławomir JASZCZAK

WYDZIAŁ INFORMATYKI, ZACHODNIOPOMORSKI UNIWERSYTET TECHNOLOGICZNY,
ul. Żołnierska 49, 71-210 Szczecin

Szybkie prototypowanie cyfrowego algorytmu fuzy PID w sterowniku B&R

Dr inż. Sławomir JASZCZAK

Od 1994 jako asystent na Wydziale Techniki Morskiej Politechniki Szczecińskiej prowadzi badania w zakresie sterowania pojazdami głębinowymi z wykorzystaniem metod sztucznej inteligencji. Od 2002 zatrudniony na stanowisku adiunkta w Zakładzie Sztucznej Inteligencji. Bieżące zainteresowania wiążą się z implementacją złożonych algorytmów sterowania dyskretnego i cyfrowego na platformie wykonawczej PLC.



e-mail: sjaszczyk@wi.zut.edu.pl

Streszczenie

W artykule omówiono zastosowanie biblioteki B&R Automation Studio Toolbox, rozszerzającej program Simulink o możliwość szybkiego prototypowania algorytmów sterowania z przeznaczeniem do urządzeń sterujących B&R. Procedura szybkiego prototypowania algorytmów sterowania cyfrowego została przeprowadzona na przykładzie cyfrowego algorytmu fuzy PID, który po wygenerowaniu z poziomu programu Simulink, przetestowano w rzeczywistym układzie sterowania temperaturą, w oparciu o urządzenie sterujące PP045.

Słowa kluczowe: szybkie prototypowanie układów sterowania, sterowanie rozmyte, algorytm PID, wbudowane funkcje wejście/wyjście.

Rapid prototyping of the digital fuzzy PID algorithm in a B&R controller**Abstract**

In this paper there is described application of the B&R Automation Studio Toolbox, giving users of the Simulink an opportunity of rapid prototyping of digital control algorithms for B&R control devices using original embedded functions. The idea of the rapid prototyping in development of digital control systems is related to automatic generation of an executing code in the C/C++ languages with accordance to the control device architecture and specific conditions connected with realization of the algorithm in the real time. Many potentially good ideas have been thrown away because of a long time connected with implementation in a specific device platform and a risk of getting a completely faulty solution. At the beginning, elements of the system of rapid prototyping (Fig.1) with a detailed step by step procedure (Fig.2) are described. In the next part a conception of the fuzzy PID algorithm (Fig.3), being a modification of the classical PID algorithm, is given. Next, a methodology of development of the fuzzy PID algorithm including: fuzzification (Fig.4), inference and defuzzification (Fig.5) and arrangement of the prototype of the fuzzy PID algorithm (Fig.6) in the Matlab/Simulink environment is presented. In the final part of this paper, selected results of the experimental research, related to the temperature stabilization using the developed and implemented fuzzy PID algorithm according to the presented procedure are given and described.

Keywords: rapid prototyping of control systems, fuzzy control, PID algorithm, embedded input/output functions.

1. Wstęp

Idea szybkiego prototypowania w projektowaniu cyfrowych układów sterowania wiąże się z automatycznym generowaniem kodu wykonawczego algorytmu sterowania z uwzględnieniem architektury platformy sprzętowej oraz wymagań realizacji algorytmu w czasie rzeczywistym. Projektant nie musi koncentrować się na kodowaniu i diagnostyce kodu, w przypadku wystąpienia błędów syntaktycznych i semantycznych, a jedynie na optymalizacji algorytmu sterowania, zgodnie z postawionymi kryteriami jakości. W praktyce istotna jest również możliwość szybkiej modyfikacji kodu wykonawczego.

Wiele potencjalnie dobrych pomysłów jest stosunkowo szybko odrzucanych z powodu nakładu czasu, związanego z implementacją pod konkretną platformę wykonawczą i ryzykiem uzyskania rozwiązania kompletnie nieprzydatnego.

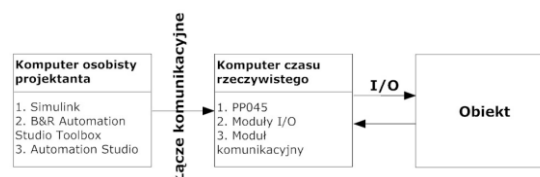
Generowanie kodu wykonawczego na konkretną platformę sprzętową jest ponadto podstawą procedury Hardware In The Loop, która umożliwia testowanie algorytmów sterowania bez ryzyka zniszczenia kosztownych elementów obiektu, urządzenia wykonawczo-nastawczego oraz urządzenia pomiarowego.

W artykule omówiono procedurę szybkiego prototypowania algorytmu fuzy PID przy wykorzystaniu programu Simulink z biblioteką B&R Automation Runtime Target for Simulink, dedykowaną do urządzeń sterujących firmy Bernecker&Reiner.

W rozdziale końcowym przedstawiono wyniki badań eksperymentalnych, polegających na sterowaniu temperaturą z wykorzystaniem zaprojektowanego i zaimplementowanego algorytmu, zgodnie z omówioną procedurą.

2. Szybkie prototypowanie algorytmów sterowania

Szybkie prototypowanie systemów sterowania jest pewną odmianą symulacji Hardware in the Loop [7], przy czym wymaga zastosowania docelowej platformy wykonawczej i rzeczywistego obiektu sterowania. Uogólnioną strukturę stanowiska badawczego przedstawiono na rys. 1.



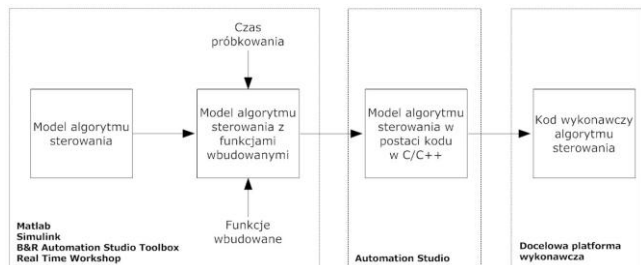
Rys. 1. Typowe elementy systemu szybkiego prototypowania
Fig. 1. Typical elements of the rapid prototyping system

Stanowisko do szybkiego prototypowania algorytmów sterowania obejmuje :

1. oprogramowanie do modelowania matematycznego np. Simulink;
2. bibliotekę funkcji wbudowanych, dedykowanych do określonej architektury sprzętowej urządzeń sterujących np. B&R Automation Studio Target for Simulink;
3. oprogramowanie narzędziowe do programowania i obsługi urządzeń sterujących np. Automation Studio;
4. komputer czasu rzeczywistego z obsługą wejść i wyjść analogowych i dyskretnych np. Power Panel PP045;
5. komputer nadrzędny z oprogramowaniem 1, 2 i 3 oraz połączeniem komunikacyjnym z komputerem czasu rzeczywistego 4.

Biblioteka B&R Automation Studio Target for Simulink stanowi podstawę realizacji procedury szybkiego prototypowania a jej zastosowanie umożliwia automatyczne generowanie kodu programu sterującego, z poziomu oprogramowania MatLab/ Simulink, do urządzeń sterujących firmy Bernecker & Reiner, rozszerzając funkcjonalność standardowej biblioteki Real Time Workshop.

Zgodnie z rys. 2 implementacja algorytmu sterowania jest realizowana w środowisku Matlab/Simulink, co odpowiada praktycznie procedurze Model In The Loop. W kolejnym kroku dodawane są funkcje wbudowane I/O i określany jest czas próbkowania, oznaczający w przypadku urządzeń sterujących B&R przydział do klasy zadaniowej, po czym następuje automatyczne wygenerowanie kodu w postaci tekstowej w języku Ansi C. Uzyskany model algorytmu, dołączany do projektu w programie Automation Studio, można po skompilowaniu przesłać na docelową platformę wykonawczą.



Rys. 2. Typowy proces szybkiego prototypowania
Fig. 2. Typical process of rapid prototyping

Podsumowując można stwierdzić, że B&R Automation Studio Toolbox umożliwia szybką i stosunkowo łatwą transformację skomplikowanych systemów sterowania, opierających się na programie Simulink, w kod źródłowy i jego integrację z projektem w Automation Studio.

3. Koncepcja algorytmu fuzy PID

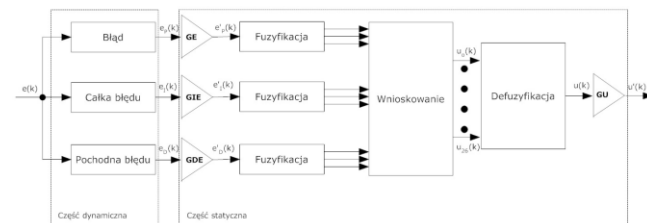
W praktyce przemysłowej klasyczny regulator PID jest nadal dominującym algorytmem, wykorzystywanym do sterowania procesami ciągłymi. W [9] W.J. Parkinson i R.E. Smith stwierdzili nawet, że algorytm PID pozostanie nadal podstawowym rozwiązaniem, wykorzystywanym w praktyce przemysłowej. Okazuje się bowiem, że ujemne sprzężenie zwrotne w powiązaniu z algorytmem PID zapewnia układowi sterowania stosunkowo dużą odporność na: niestacjonarność, nieliniowość i losowość zakłóceń. Przy czym, praktyczne trudności takie jak konieczność długotrwałego strojenia i przestrajania algorytmu PID przy zmieniających się reżimach ruchowych wymuszają na projektantach poszukiwanie alternatywnych metod sterowania i/lub metod poprawy jakości sterowania w układach z algorytmem PID. W ciągu ostatnich dwudziestu lat stopniowo znaczenie na tym polu algorytmy oparte o logikę rozmytą m.in. klasyczne algorytmy Mamdaniego, Takagi-Sugeno, adaptacyjne algorytmy rozmyte (ang. fuzzy adaptive controllers), nadzorujące algorytmy rozmyte (ang. fuzzy supervisory controllers), działające w warstwie sterowania nadrzędnego oraz hybrydowe algorytmy rozmyte tj. wiążące kilka metod sztucznej inteligencji (sztuczne sieci neuronowe, algorytmy genetyczne, uczenie maszynowe itp.).

Zastosowanie logiki rozmytej umożliwiło bowiem wykorzystanie wiedzy eksperckiej w strategii sterowania zapisanej w postaci formuły matematycznej. Inaczej niż w tradycyjnej teorii sterowania synteza rozmytego algorytmu sterowania nie wymaga znajomości dynamiki obiektu sterowania w postaci modelu matematycznego, lecz zapisania wiedzy eksperta, dotyczącej metody sterowania obiektem w postaci reguł lingwistycznych. Z praktycznego punktu widzenia przekłada się to na konieczność implementacji modułu fuzyfikacji, wnioskowania i defuzyfikacji. Niestety zgodnie z [5, 6 i 8] synteza dowolnego algorytmu rozmytego jest prowadzona zazwyczaj metodą prób i błędów i wymaga wielu eksperymentów, aby znaleźć możliwe do zaakceptowania nastawy algorytmu. Ponadto liczba stopni swobody algorytmów rozmytych, wynikająca z liczby parametrów nastawnych stanęła na przeszkodzie do wdrażania ich w praktyce przemysłowej, a dokładniej brak metodyki strojenia parametrów, która istnieje w przypadku algorytmów PID. W [6] i [8] zaproponowano nowatorskie podejścia do zagadnienia projektowania algorytmu fuzy PID, zmierzające do ograniczenia liczby strojonych parametrów nastawnych i powiązania z metodyką projektowania klasycznego algorytmu PID. Istotne jest aby alternatywne rozwiązanie w stosunku do klasycznego algorytmu PID cechowało się [1] łatwością użycia, strojenia, serwisowania i określoną jakością działania.

Poza tym w przypadku aplikacji sterujących, przeznaczonych do zastosowania w przemyśle należy zwrócić uwagę na ograniczenia docelowej platformy wykonawczej algorytmów sterowania, którą zazwyczaj stanowi sterownik programowalny PLC. Ograniczenia te wynikają bezpośrednio z ograniczeń składni języków

programowania sterowników PLC, zazwyczaj LD (ang. ladder diagram) i/lub IL (ang. instruction list) oraz limitowanej pamięci danych i programu, co jest szczególnie odczuwalne w przypadku sterowników klas nano i micro. W przypadku urządzeń sterujących B&R wymienione ograniczenia zostały w dużej mierze usunięte dzięki możliwości programowania w językach wysokiego poziomu: ST (ang. structured text), AB (ang. automation basic), Ansi C oraz możliwości szybkiego prototypowania. Wymienione cechy zdecydowały o wyborze urządzenia sterującego B&R, jako docelowej platformy wykonawczej algorytmów rozmytych.

Za [6], [8] autor proponuje wykorzystać strukturę klasycznego algorytmu PID do skonstruowania algorytmu fuzy PID, którego uogólnioną strukturę przedstawiono na rys. 3.



Rys. 3. Uogólniona struktura algorytmu fuzy PID
Fig. 3. General structure of the fuzzy PID algorithm

Sygnal $e_p(k)$ odpowiada sygnałowi błędowi regulacji, sygnał $e_f(k)$ stanowi całkę błędów regulacji natomiast sygnał $e_d(k)$ określa pochodną błędów regulacji.

Do numerycznego wyznaczenia wartości całki i pochodnej sygnału błędów regulacji zastosowano możliwe najprostsze aproksymacje, przedstawione wzorami (1) i (2):

$$e_f(k) = \sum e(k) \cdot T_s, \quad (1)$$

$$e_d(k) = \frac{e(k) - e(k-1)}{T_s}, \quad (2)$$

gdzie: T_s oznacza czas próbkowania, natomiast $e(k)$ błąd regulacji, określany w każdym kroku próbkowania k jako

$$e(k) = SP(k) - PV(k), \quad (3)$$

gdzie $SP(k)$ to wartość zadana, a $PV(k)$ wartość sygnału sterowanego.

Sygnały ($e_p(k)$, $e_f(k)$, $e_d(k)$) poddaje się fuzyfikacji, przy wykorzystaniu trójkątnych funkcji przynależności, uprzednio poddając je skalowaniu za pomocą współczynników GE , GIE oraz GDE . Współczynniki skalujące umożliwiają stosunkowo szybkie dopasowanie zakresu sygnałów do zakresu, w którym nie występują nasycenia funkcji przynależności, co określono na rys.4, jako zakres pożądany. Niestety podobnie jak w przypadku określenia dziedziny, obejmującej określony sygnał, wartości współczynników trzeba określić eksperymentalnie, co w przypadku obiektów wolnozmiennych znacząco utrudnia proces wstępnego strojenia algorytmu.

W algorytmie zastosowano fuzyfikację z trzema wartościami lingwistycznymi (rys. 4.).

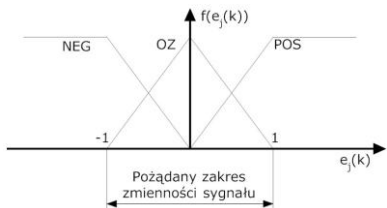
Kombinacjom 27 sfuzyfikowanych wejść algorytmu fuzy PID odpowiada 27 reguł inferencyjnych R_i , o następującej postaci:

$$R1: \text{IF } [e_p(k)=e_{PNEG}(k)] \text{ AND } [e_f(k)=e_{INEG}(k)] \text{ AND } [e_d(k)=e_{DNEG}(k)] \text{ THEN } [U(k)=U_{26}(k)]$$

$$R14: \text{IF } [e_p(k)=e_{POZ}(k)] \text{ AND } [e_f(k)=e_{IOZ}(k)] \text{ AND } [e_d(k)=e_{DOZ}(k)] \text{ THEN } [U(k)=U_{13}(k)]$$

$$R27: \text{IF } [e_p(k)=e_{PPOS}(k)] \text{ AND } [e_f(k)=e_{IPOS}(k)] \text{ AND } [e_d(k)=e_{DPOS}(k)] \text{ THEN } [U(k)=U_0(k)]$$

Przesłanka każdej reguły R_i oceniana jest przy pomocy bloku funkcyjnego, realizującego operację przecięcia zbiorów AND.

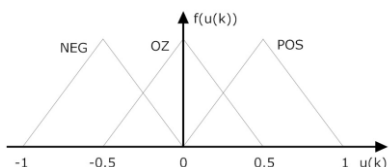


$e_{jNEG}(e_j)$ - funkcje przynależności do wartości lingwistycznej „negative”
 $e_{jOZ}(e_j)$ - funkcje przynależności do wartości lingwistycznej „zero”
 $e_{jPOS}(e_j)$ - funkcje przynależności do wartości lingwistycznej „positive”
 $j: P, I, D$

Rys. 4. Funkcje przynależności w algorytmie fuzzy PID
 Fig. 4. Membership functions in the fuzzy PID algorithm

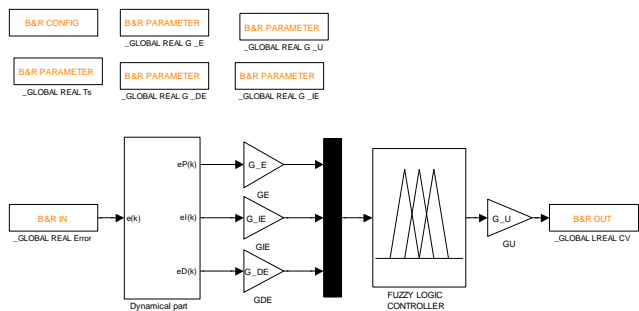
Defuzyfikacja w algorytmie przeprowadzana jest metodą środka ciężkości (ang. Center Of Gravity), przy wykorzystaniu trójkątnych funkcji przynależności (rys. 5) oraz zależności (4).

$$U(k) = \frac{\sum_i f(u_i(k)) \cdot u_i(k)}{\sum_i f(u_i(k))}, \quad (4)$$



Rys. 5. Defuzyfikacja w algorytmie fuzzy PID
 Fig. 5. Defuzzification function in the fuzzy PID algorithm

Wykorzystując Fuzzy Logic Toolbox, Simulink oraz B&R Automation Studio Toolbox opracowano prototyp algorytmu fuzzy PID wraz z funkcjami wbudowanymi, który przedstawiono na rys. 6.



Rys. 6. Prototyp algorytmu fuzzy PID z funkcjami wbudowanymi w programie Simulink
 Fig. 6. Prototype of the fuzzy PID algorithm with embedded functions in the Simulink

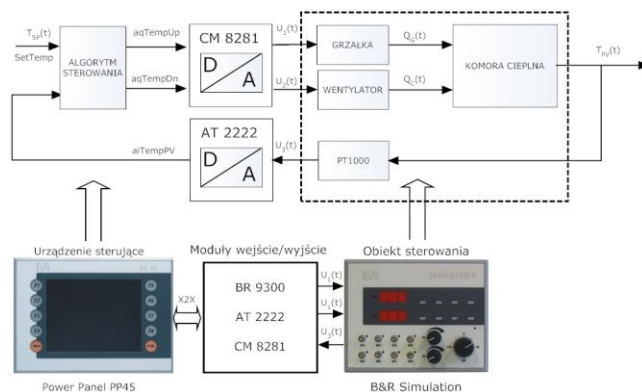
4. Wyniki badań eksperymentalnych

Zaprojektowany algorytm fuzzy PID przetestowano w układzie sterowania temperaturą, który przedstawiono na rys. 7. Urządzenie sterujące stanowi Power Panel PP45 firmy Bernecker&Reiner, które przy wykorzystaniu sieci X2X komunikuje się z układami wejście/wyjście (moduły AC/CA CM8281 oraz moduł temperatury AT2222).

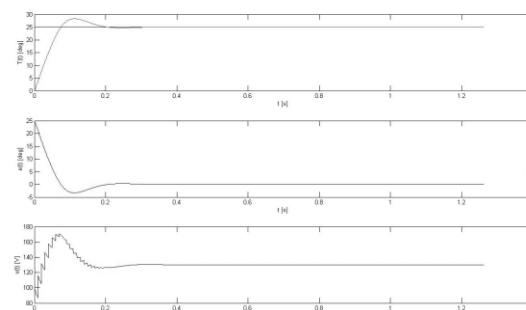
Obiekt sterowania stanowi komora cieplna wraz z układami wykonawczymi (grzałka i wentylator) oraz układem pomiarowym (sonda PT1000) [4]. Sygnał sterowany stanowi napięcie $U_3(t)$, reprezentujące temperaturę $T_{PV}(t)$, natomiast sygnały sterujące stanowią napięcia $U_1(t)$ i $U_2(t)$, odpowiadające za pracę grzałki i wentylatora.

Poniżej przedstawiono wybrany wynik próby skokowej, w układzie stabilizacji temperatury, z zaprojektowanym algorytmem fuzzy PID, przeprowadzonym w czasie rzeczywistym

(rys. 8). Uzyskany przebieg sygnału regulowanego potwierdza poprawność wykonanej procedury na poziomie implementacyjnym.



Rys. 7. Układ sterowania temperaturą z zaprojektowanym algorytmem fuzzy PID
 Fig. 7. Temperature control system with the developed fuzzy PID algorithm



Rys. 8. Wybrana odpowiedź skokowa w układzie zamkniętym z algorytmem fuzzy PID
 Fig. 8. Selected step response in a feedback loop control system with the developed fuzzy PID algorithm

5. Podsumowanie

Na przykładzie zaprojektowanego i zaimplementowanego algorytmu fuzzy PID potwierdzono efektywność procedury szybkiego prototypowania z wykorzystaniem urządzeń sterujących B&R.

Projektowanie algorytmów sterowania, przy wykorzystaniu metodyki szybkiego prototypowania przyspiesza projektowanie i modyfikacje oryginalnych rozwiązań, dzięki zastosowaniu środowiska Matlab/Simulink, co istotnie wpływa na koszt wdrożenia.

6. Literatura

- [1] Åström K.J., Hägglund T.: The future of PID control, Control Engineering practice 9 (2001) Page(s): 1163-1175.
- [2] Brzózka J.: Regulatory cyfrowe w automatyce. Mikom, Warszawa, 2002.
- [3] B&R Automation Runtime Target for Simulink Technical Manual TM140
- [4] B&R Simulation Technical Description 4SIM.00-01, 2007.
- [5] Chopra S., Mitra R., Kumar V.: Fuzzy controller: choosing an appropriate and smallest rule set. International Journal of Computational Cognition, Vol. 3, No. 4, December 2005.
- [6] Jantzen J.: Design Of Fuzzy Controllers. Technical Tutorial, 1998.
- [7] Jaszczak S.: Procedura hardware in the loop w syntezy algorytmów sterowania, PAK 2010.
- [8] Pivoňka P.: Comparative Analysis of Fuzzy PI/PD/PID Controller Based on Classical PID.
- [9] Ross T. J., Booker J.M., Parkinson W. J.: Fuzzy logic and probability applications: bridging the gap. Philadelphia : Amer. Statistical Assoc.: Soc. for Industrial and Appl. Mathematics, 2002.