Logika rozmyta,
Robotyka,
Programowanie

ŁUKASIK Zbigniew[1]
PNIEWSKA Beata[2]
PNIEWSKI Roman[3]

ZASTOSOWANIE LOGIKI ROZMYTEJ W STEROWANIU ROBOTEM LEGO

Zestawy LEGO Mindsotrms oraz NXT umożliwiają konstruowanie robotów i urządzeń mechatronicznych. Oprogramowanie firmowe Lego pozwala na graficzne konstruowanie stosunkowo prostych programów. Kompilator RobotC zapewnia możliwość programowania w języku C, co autorzy wykorzystali do stworzenia algorytmu sterowania z zastosowaniem logiki rozmytej.

APPLICATION FUZZY LOGIC IN CONTROL LEGO ROBOT

LEGO NXTAND Mindsotrms allow the construction of robotic and mechatronic devices. Lego firmware allows the graphical construction of relatively simple programs. RobotC compiler provides the ability to program in C language, which the authors used to create a control algorithm using fuzzy logic.

1. WSTĘP

Zestawy LEGO Mindstorms i NXT stanowią doskonałe narzędzie do nauki robotyki i mechatroniki. Oprogramowanie, dostarczane razem z zestawami umożliwia graficzne tworzenie średnio zaawansowanych algorytmów sterowania. Oprócz oryginalnego oprogramowania Lego powstało wiele innych kompilatorów (dla różnych języków: C, Java, Pascal…..), opracowanych przez inne firmy. Szczególne duże możliwości oferuje system RobotC, wykorzystywany przez autorów przy konstruowaniu algorytmów sterowania z zastosowaniem logiki rozmytej.

[1] Politechnika Radomska, Wydział Transportu i Elektrotechniki; 26-600 Radom; ul. Malczewskiego 29. *Tel: + 48 48 361-77-15, Fax: + 48 48 361-77-42, E-mail: z.lukasik@pr.radom.pl*
[2] Politechnika Radomska, Wydział Transportu i Elektrotechniki; 26-600 Radom; ul. Malczewskiego 29. Tel: + 48 48 361-77-16, Fax: + 48 48 361-77-42, E-mail: b.pniewska@pr.radom.pl
[3] Politechnika Radomska, Wydział Transportu i Elektrotechniki; 26-600 Radom; ul. Malczewskiego 29. Tel: + 48 48 361-77-28, Fax: + 48 48 361-77-42, E-mail: r.pniewski@pr.radom.pl

2. LOGIKA ROZMYTA

Układy sterowania rozmytego należą do klasy sterowań opartych na wiedzy (ang. KBS knowledge based system). Wykorzystanie wiedzy, która nie może być zawarta w analitycznym modelu obiektu, pozwala na zwiększenie niezawodności i odporności na błędy [1]. Układy sterowania oparte o reguły rozmyte mogą realizować sterowanie bezpośrednie lub nadrzędne.

2.1 Podstawy matematyczne

Zbiór: zespół obiektów mających wspólne i rozróżniające cechy. Granice oddzielające elementy należące do danego zbioru i nie należące do niego są jednoznacznie określone.
Funkcję μ_Ω określającą przynależność elementu do zbioru definiuje się następująco:

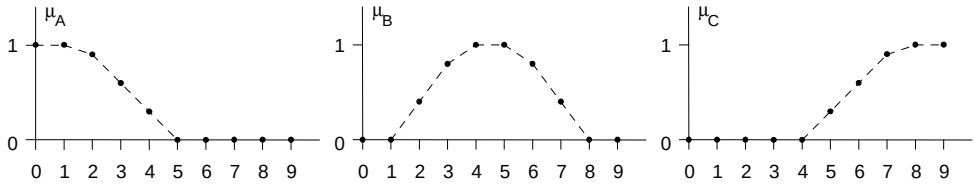
$$\forall u \in U \;\; \mu_\Omega(u) = \begin{cases} 1, & u \in \Omega, \\ 0, & u \notin \Omega, \end{cases} \tag{2.1}$$

gdzie:
U - przestrzeń rozważanych obiektów
Ω- zbiór odpowiadający rozpatrywanej własności.

W przypadku konieczności określenia pojęć „mniej ostrych", trudno jest określić granicę rozdzielającą elementy należące do zbioru od elementów nie spełniających żądanej własności. Rozszerzenie wartości funkcji μ do przedziału [0,1] (w przeciwieństwie do zbioru dwuelementowego {0,1}) umożliwia definiowanie pojęć „mniej ostrych". Zbiór charakteryzowany przez funkcję przynależności μ (ang. membership function) nazywany jest zbiorem rozmytym.
Jako przykład rozpatrzmy następujące trzy własności:
A - cyfra mała
B - cyfra średnia
C - cyfra duża
Przykładowe wartości funkcji przynależności dla powyższych własności są następujące (na rys.1. pokazano odpowiadające im wykresy):
$\mu_A(0) = 1$, $\mu_A(1) = 1$, $\mu_A(2) = 0.9$, $\mu_A(3) = 0.6$, $\mu_A(4) = 0.3$,
$\mu_A(5) = 0$, $\mu_A(6) = 0$, $\mu_A(7) = 0$, $\mu_A(8) = 0$, $\mu_A(9) = 0$,

$\mu_B(0) = 0$, $\mu_B(1) = 0$, $\mu_B(2) = 0.4$, $\mu_B(3) = 0.8$, $\mu_B(4) = 1$,
$\mu_B(5) = 1$, $\mu_B(6) = 0.8$, $\mu_B(7) = 0.4$, $\mu_B(8) = 0$, $\mu_B(9) = 0$,

$\mu_C(0) = 0$, $\mu_C(1) = 0$, $\mu_C(2) = 0$, $\mu_C(3) = 0$, $\mu_C(4) = 0$,
$\mu_C(5) = 0.3$, $\mu_C(6) = 0.6$, $\mu_C(7) = 0.9$, $\mu_C(8) = 1$, $\mu_C(9) = 1$,



Rys.1. Wykresy funkcji przynależności dla własności:
μ_A - mała cyfra, μ_B - średnia cyfra, μ_C - duża cyfra

W działaniach na zbiorach rozmytych wykorzystuje się wiele działań wywodzących się z algebry zbiorów przy wykorzystaniu zasady rozszerzania [1].
Poniżej przedstawiono wybrane działania, które są najczęściej wykorzystywane w układach sterowania:
Dopełnienie (bezwzględne) zbioru rozmytego:

$$\mu_{\neg A}(x) = 1 - \mu_A(x), \qquad \forall x \in X \tag{2.2}$$

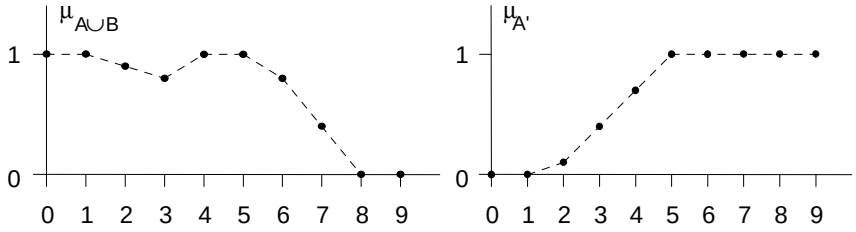
Suma (mnogościowa) zbiorów rozmytych:

$$\mu_{A+B}(x) = \mu_A \vee \mu_B, \qquad \forall x \in X \tag{2.3}$$

Przecięcie zbiorów rozmytych:

$$\mu_{A\cap B} = \mu_A(x) \wedge \mu_B(x), \qquad \forall x \in X \tag{2.4}$$

Na rysunku 2. przedstawiono przykładowe wykresy funkcji przynależności A∪B - cyfra mała lub cyfra średnia i A' - cyfra niemała.



Rys.2. Przykładowe operacje na zborach rozmytych

Konstruowane sterowniki z logiką rozmytą (ang. Fuzzy Logic Controllers) generują sygnał wyjściowy na podstawie relacji logicznych typu: Jeżeli „warunek" To „konkluzja". Wykorzystuje się przy tym jedną z podstawowych tautologii:

$$A \rightarrow B = \overline{A} \vee B \tag{2.5}$$

Przykładowa Baza reguł FLC:

Jeżeli uchyb e(k) jest dodatni **I** zmiana uchybu Δe(k) jest zero
To zmiana sterowania Δu(k) jest dodatnia
Jeżeli uchyb e(k) jest ujemny **I** zmiana uchybu Δe(k) jest zero
To zmiana sterowania Δu(k) jest ujemna

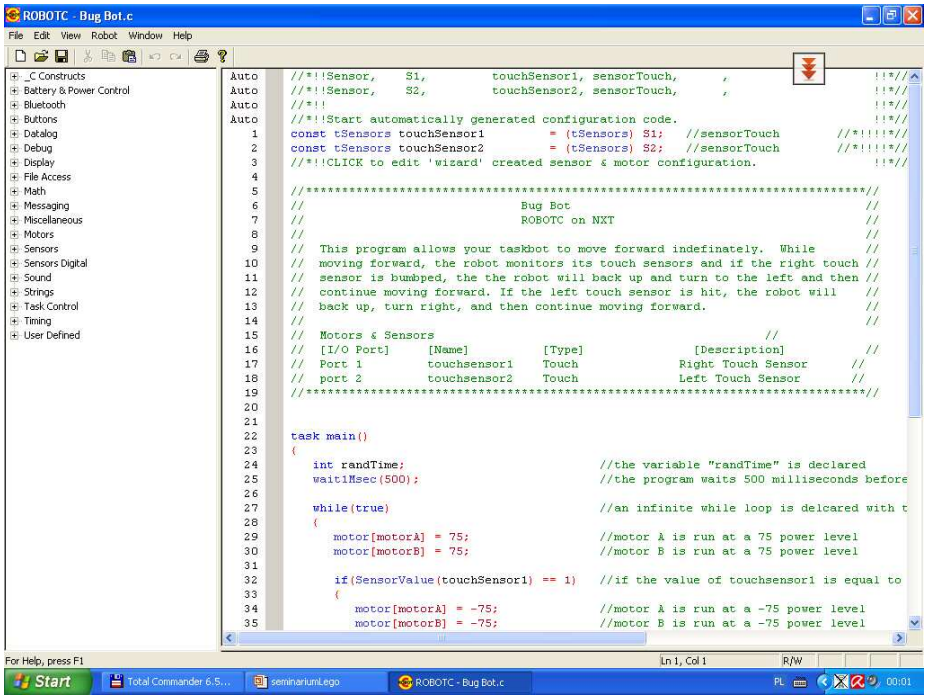
Jeżeli uchyb e(k) jest zero **I** zmiana uchybu Δe(k) jest zero
To zmiana sterowania Δu(k) jest zero
Jeżeli uchyb e(k) jest zero **I** zmiana uchybu Δe(k) jest dodatnia
To zmiana sterowania Δu(k) jest dodatnia
Jeżeli uchyb e(k) jest zero **I** zmiana uchybu Δe(k) jest ujemna
To zmiana sterowania Δu(k) jest ujemna

3. PROGRAMOWANIE ZESTAWU LEGO NXT W JĘZYKU C

3.1. Kompilator RobotC

Oprogramowanie RobotC to narzędzie, przeznaczone do programowania zestawów LEGO Mindstorms i NXT. Charakteryzuje się następującymi cechami:
- Środowisko graficzne,
- zawiera edytor kompilator i debugger,
- możliwość bezpośredniej transmisji programu do sterownika (USB, Bluetooth),

Na rys.3 pokazano przykładowy zrzut ekranu z programu RobotC.



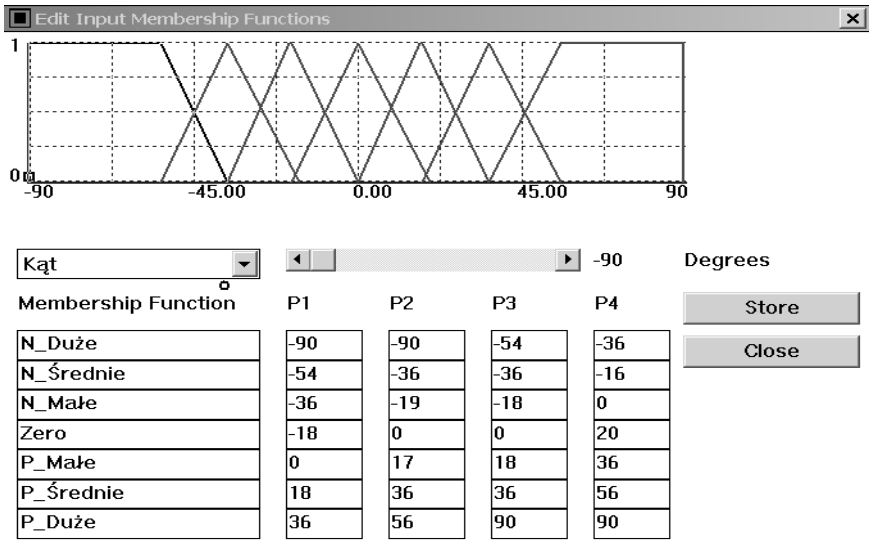
Rys.3. Zrzut ekranu z programu RobotC

4. STRUKTURA OPROGRAMOWANIA

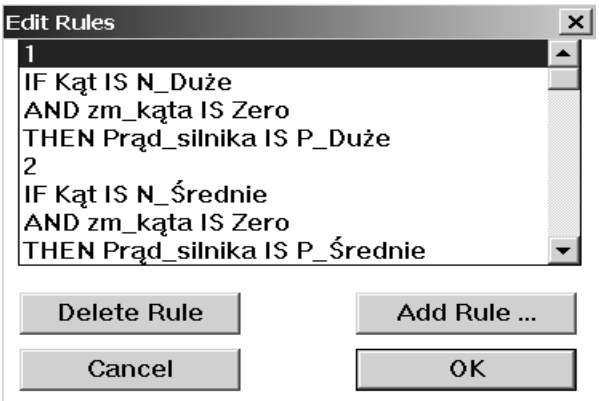
4.1. Edytor relacji rozmytych (FUDGE)

Do syntezy sterownika wykorzystano program FUDGE (FUzzy Design GEnerator), który stworzony został do syntezy sterowników rozmytych, opartych na mikrokontrolerach

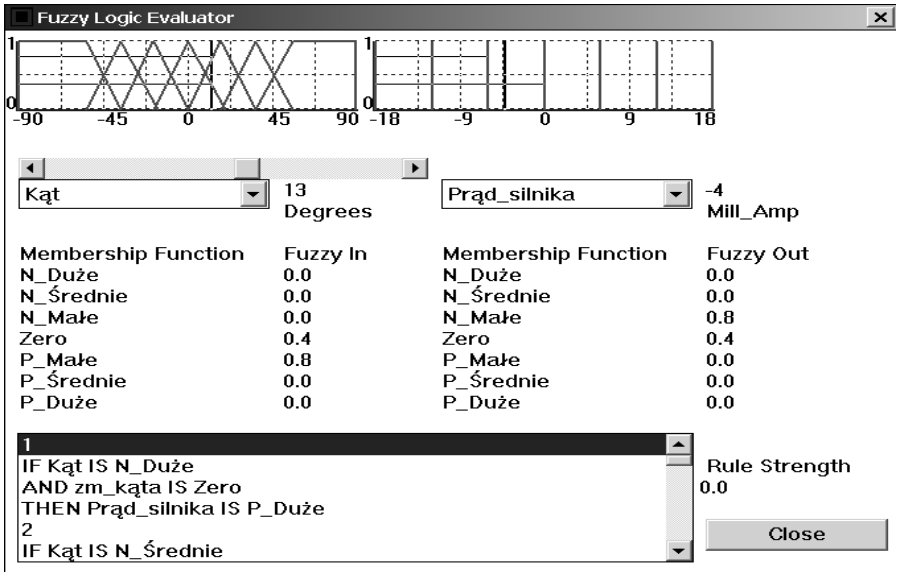
rodzin HC05, HC11 firmy Motorola. W programie, po wprowadzeniu funkcji opisujących zmienne wejściowe i wyjściowe (ang. Membership Function) – rys.4, oraz po zdefiniowaniu relacji (Rules) między funkcjami wejściowymi i wyjściowymi – rys.5 możliwe jest przetestowanie zaprojektowanego sterownika. Największe możliwości kryją się w opcji „Fuzzy Logic Evaluator" (rys.6), gdzie można przetestować reakcję sterownika na wszystkie możliwe zmiany zmiennych wejściowych. Jednocześnie program raportuje wartość generowaną przez wybraną relację. Po dostrojeniu i przetestowaniu sterownika można wygenerować plik wyjściowy. Program tworzy kody źródłowe w asemblerze dla mikrokontrolerów: HC05, HC11, HC16, 68000. Dodatkowo można wygenerować plik źródłowy w języku C (Ansi). Z tej ostatniej możliwości skorzystano w przypadku realizowanego projektu.



Rys.4. Definiowanie funkcji wejściowych



Rys.5 Edycja relacji



Rys.6 Dostrajanie regulatora

5. WNIOSKI

Zaprojektowany układ sterownika realizował założone funkcje. Wadą prezentowanego rozwiązania jest konieczność ponownej kompilacji projektu po każdej zmianie parametrów sterownika. W nowej wersja sterownika, która jest aktualnie realizowana zmieniona została całkowicie koncepcja generowania wartości funkcji i relacji między zmiennymi. W kontrolerze przewidziano pewne graniczne ilości parametrów (dotyczących zmiennych lingwistycznych i relacji). Wartości te, po restarcie kontrolera będą wczytywane z zewnętrznej pamięci EEPROM, co umożliwi przestrojenie kontrolera bez zmiany podstawowego oprogramowania sterownika.

6. BIBLIOGRAFIA

[1] Driankov D., Hellendorn H., Reinfrank M., Wprowadzenie do sterowania rozmytego. WNT 1996

[2] Łukasik Z., Pniewska B., Pniewski R.: *Laboratorium Komputerowych Systemów Sterowania*, Wydawnictwo Politechniki Radomskiej, Radom 2004.