

PNIEWSKA Beata, PNIEWSKI Roman

LOGIKA ROZMYTA W STEROWANIU ROBOTEM LEGO

Streszczenie

W dydaktyce automatyki i robotyki, do realizacji minirobotów wykorzystywane są zestawy LEGO Mindstorms. Pierwotnie przeznaczone były do realizacji prostych układów i nieskomplikowanych algorytmów sterowania (przy wykorzystaniu dołączonego do zestawu oprogramowania graficznego). Ponieważ zestawy te były użytkowane na wielu uczelniach powstało szereg programów umożliwiających oprogramowanie tych zestawów, niezależnych od firmy LEGO. Systemy te umożliwiają implementację rozbudowanych algorytmów sterowania (przy wykorzystaniu różnych języków programowania). W artykule pokazano możliwości wykorzystania logiki rozmytej do realizacji inteligentnych algorytmów sterowania.

WSTĘP

Idea wykorzystania klocków LEGO do budowy robotów, powstała w latach osiemdziesiątych w Massachusetts Institute of Technology. W tym czasie, na rynku były już dostępne takie elementy zestawów LEGO jak: silniki, przekładnie, proste czujniki. Zespół w składzie: Fred Martin, Mitchel Resnick, Rand Sargent, Seymour Papert i Brian Silverman postanowili poszerzyć listę dostępnych składników o mikrokomputer zawarty w klocku. Umożliwiło to budowę złożonych mechanizmów i robotów z zaawansowanym sposobem sterowania. 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 oferują systemy RobotC oraz Brick Command Center, wykorzystywane przez autorów przy konstruowaniu algorytmów sterowania z zastosowaniem logiki rozmytej.

1. ZESTAW LEGO MINDSTORMS

LEGO Mindstorms to zestaw edukacyjny umożliwiający: budowę robotów, budowę układów automatyki komunikację z nimi przez Bluetooth i przez USB.

W skład zestawu wchodzi : 'brick' - komputerowa jednostka centralna, czujniki elektroniczne, serwomechanizmy, klocki LEGO.

Dane techniczne NXT:

32-bit ARM7 microcontroller

256 Kbytes FLASH, 64 Kbytes RAM

8-bit AVR microcontroller

4 Kbytes FLASH, 512 Byte RAM

Bluetooth wireless communication (Bluetooth Class II V2.0 compliant)

USB full speed port (12 Mbit/s)
4 input ports, 6-wire cable digital platform
3 output ports, 6-wire cable digital platform
100 x 64 pixel LCD graphical display



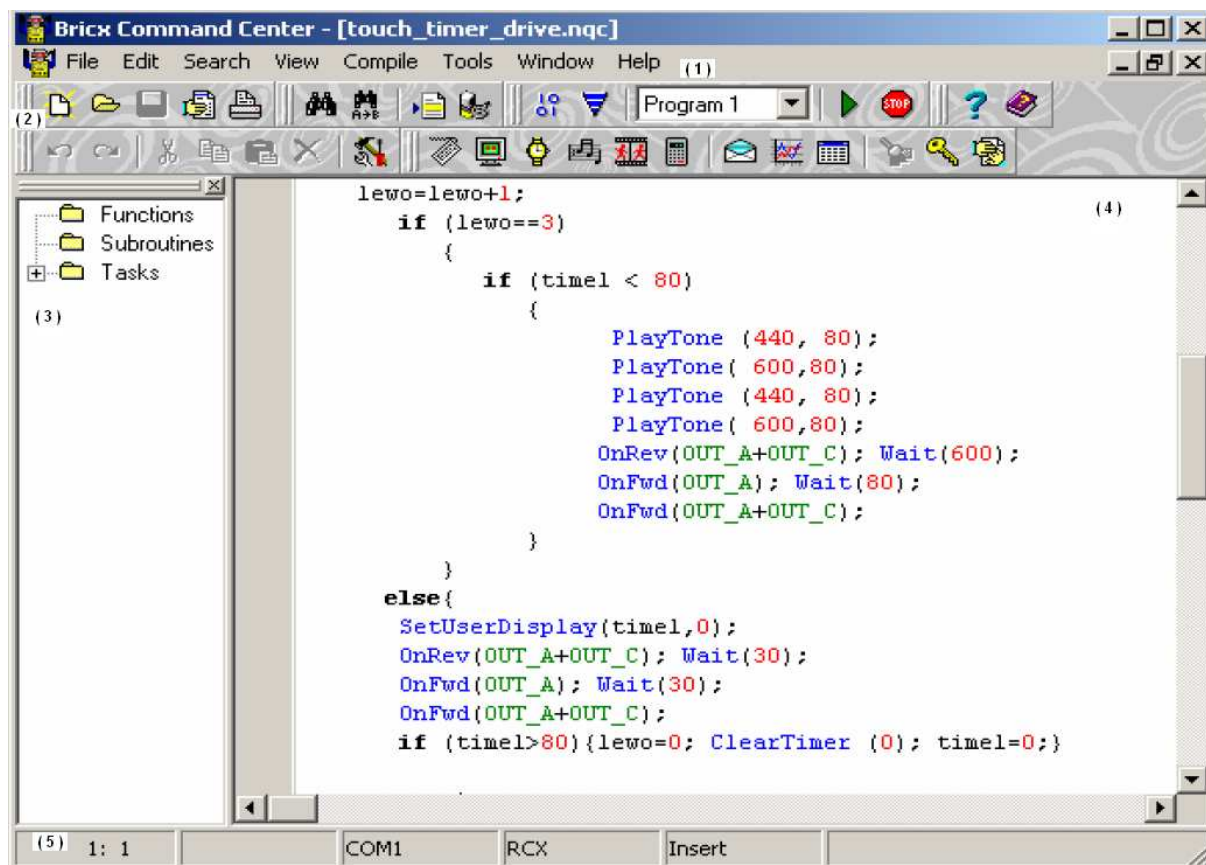
Rys.1. Połączenie czujników i serwomechanizmów z jednostką centralną.

Do zestawu LEGO Mindstorms dołączono oprogramowania LegoMindstorms NXT – jest to środowisko zaprojektowane przez firmę Lego do tworzenia aplikacji przy użyciu schematów blokowych. Umożliwia ono tworzenie prostych algorytmów sterowania. Oprócz „fabrycznego” oprogramowania LEGO powstało wiele innych systemów (darmowych i komercyjnych) umożliwiających tworzenie i uruchamianie programów sterujących robotami LEGO. O możliwościach „poważnego” wykorzystania zestawów LEGO w uruchamianiu i testowaniu nowych algorytmów sterowania w robotyce i automatyce może świadczyć wsparcie przez firmę National Instruments i udostępnienie bibliotek oraz narzędzi, które pozwalają na programowanie robotów w środowisku LabView.

Poniżej, skrótowo przedstawiono dwa darmowe systemy wykorzystane przez autorów przy uruchamianiu oprogramowania zestawów LEGO.

Brics Command Center

Autorem programu jest Mark Overmars. Jest to narzędzie programistyczne, służące do programowania wszystkich mikrokomputerów firmy LEGO (Wszystkie wersje RCX, Spybot, Scout, Cybermaster). Obsługuje takie języki jak NQC, BrickOS, Mindscript. Program działa na platformach wyposażonych w system operacyjny Windows. Jest on ciągle rozwijany i uaktualniany w stosunku do rozwoju technologii i języków programistycznych związanych z robotami LEGO.



Rys.2. Okno programu Brix Command Center

LegOS:

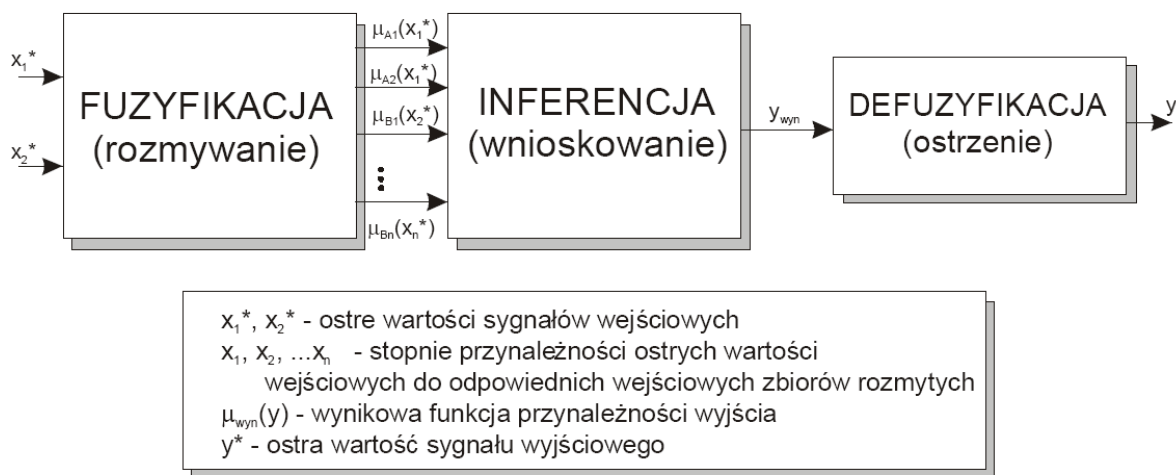
Powstał w 1999 roku, jego autorem jest Markus L. Noga, którego celem było zastąpienie własnym systemem programu firmwarowego firmy LEGO. LegOS umożliwia dzięki pozbyciu się wpływu interpretera kodu bitowego na bezpośrednie działanie na procesorze programem napisanym przez użytkownika. Pozwala to na pełne wykorzystanie całej mocy jaką oferuje sterownik RCX. Językiem programowania LegOSa jest C lub C++, można wykorzystać też dowolny assembler procesora Hitachi 8/3292, skompilowany program jest ładowany do pamięci RAM RCXa i zastępuje standardowy LEGO firmware. Takie rozwiązanie pozwala na efektywne wykorzystanie całego obszaru 32 KB pamięci RAM, również dzięki temu programy działają szybciej, gdyż nie muszą być interpretowane przez oprogramowanie firmowe. LegOS umożliwia na pełne wykorzystanie właściwie wszystkich możliwości sprzętowych mikrokontrolera: bezpośrednie sterowanie pracą silników, czujników, wyświetlacza, przycisków, komunikacji IR, dynamiczne zarządzanie pamięcią, wielowątkowość z wyłączeniem i Semaforów POSIX. Pozwala także na wykorzystanie wszystkich możliwości języka C i adresowanie każdego pojedynczego bitu pamięci RCXa. Pisane programy pod LegOSa mogą wykorzystywać zatem złożone algorytmy działania np. percepcje otoczenia, modelowanie, planowanie, działanie. Mimo ogromnych możliwości jakie oferuje ten system, ma też pewne wady. Instalacja programu jest bardzo kłopotliwa, szczególnie pod system Windows. Jest on napisany pod Unixa więc konieczne będzie zastosowanie emulatorów tego systemu oraz kompilatorów skrótnych.

2. STEROWANIE ROZMYTE

Logika rozmyta (Fuzzy-Logic) stanowi rozszerzenie klasycznej logiki boolowskiej. Wprowadza ona między absolutną prawdę i absolutny fałsz - reprezentowanych najczęściej przez 1 i 0, wartości pośrednie. Pozwala to na określenie „prawdziwości” badanego zjawiska (np. prawie fałsz, w połowie prawda, prawie prawda), i umożliwia zastosowanie rozumienia bardziej ludzkiego, gdzie często nie ma określenia wartości w konkretnych liczbach, a jest przybliżona subiektywna ocena rzeczywistości i przyporządkowanie wielkości badanego zjawiska do abstrakcyjnego przedmiotu. Logika rozmyta została sformułowana przez prof. Lotfi Zadeha. W roku 1965 stworzył on teorię zbiorów rozmytych, która posłużyła do opisanego zjawisk i pojęć, niedefiniowalnych przy pomocy klasycznej.

Reguły które opisują sterowniki rozmyte to reguły typu jeżeli...to.... Ich poprzedniki składają się z koniunkcji lub/i alternatyw wyrażeń lingwistycznych typu x jest A . Oznacza to przynależność danej zmiennej x do zbioru rozmytego określonego przez zmienną lingwistyczną A . Korzystając z teorii zbiorów rozmytych na zmienną lingwistyczną można potraktować jako zbiór termów mających postać zbiorów rozmytych określających jakieś pojęcie z języka naturalnego. Na przykład zmienna lingwistyczna „zmiana kąta” jest określona przez następujące termy: „duża_ujemna”, „średnia_ujemna”, „mała_ujemna”, „zerowa”, „mała_dodatnia”, „średnia_dodatnia”, „duża_dodatnia”. Termy te są scharakteryzowane przez odpowiednie zbiory rozmyte.

Strukturę realizacji procesu sterowania rozmytego pokazano na rys.3.



Rys.3. Sterowania rozmyte [1]

Rozmywanie (fuzyfikacja)

Jest to operacja przekształcająca wejściowe wartości liczbowe, które są powiązane z danymi zmiennymi lingwistycznymi na wartości z dziedziny zbiorów rozmytych. W tym celu dokonuje się wyznaczenia wartości funkcji przynależności dla kolejnych zmiennych lingwistycznych i dla danej rzeczywistej wartości. Najczęstszą metodą rozmywania jest metoda typu singelton.

Wnioskowanie rozmyte

W bloku INFERENCJA zostaje obliczona na podstawie wejściowych stopni przynależności wynikowa funkcja przynależności wyjścia regulatora. Proces ten odbywa się w trzech etapach poprzez:

- Agregacje - operacje która na podstawie stopnia spełnienia przesłanek poszczególnych reguł, oraz z wykorzystaniem operatorów logicznych AND, OR i NOT oblicza stopień spełnienia każdej z reguł.

- Operacje wnioskowania rozmytego, która na podstawie stopnia spełnienia poszczególnych reguł, stosując odpowiedni model wnioskowania, oblicza zbiory rozmyte stanowiące konkluzje tych reguł. Podstawą wnioskowania rozmytego jest reguła „modus ponens”. Jest ona uogólniona w stosunku do klasycznej, dostosowana do teorii zbiorów rozmytych. Reguła ta pozwala na podstawie prawdziwości poprzednika implikacji wnioskować prawdziwość jej następnika. Jednak w odróżnieniu od reguły klasycznej „modus ponens” występuje w niej pojęcie stopnia prawdziwości zarówno poprzednika jak i następnika implikacji jako pewnego predykatu. Implikacja jest dwuargumentową relacją rozmytą, która umożliwia wybór odpowiedniej funkcji która będzie wykonywać implikacje w zależności od potrzeb systemu.

- Kumulacje - jest to operacja zastosowania odpowiedniej s-normy do zbiorów rozmytych otrzymany w fazie wnioskowania i stworzenia pojedynczego zbioru rozmytego.

Przy wnioskowaniu rozmytym są stosowane różne reguły, do najczęściej stosowanych należą:

Wnioskowanie metodą Mamdaniego: opiera się ono na bazie reguł i wykorzystuje operatory lingwistyczne, oraz operacje minimum i maximum. Wynik każdej reguły jest opisany odpowiednią etykietą lingwistyczną, natomiast wynik całego modelu obliczany jest dzięki zastosowaniu superpozycji wyników każdej z reguł po przez: ustalenie wagi każdej reguły, zastosowanie zbioru rozmytego wyznaczanego (wprowadzanego) przez i-tą regułę, za pomocą operacji max przeprowadzona zostaje agregacja pozostałych zbiorów.

Wnioskowanie metodą Larsena: Wnioskowanie to jest podobne do poprzedniego. Wagi reguł oblicza się identycznie jako minimum z wartości $w1 = \min [m_{A1}(x_0), m_{B1}(y_0)]$, ale decyzję stanowi zbiór rozmyty $C1'$ o funkcji przynależności będącej iloczynem $m_{C1'}(z) = w1 * m_{C1}(z)$. Wniosek z obu reguł stanowi zbiór rozmyty C o funkcji przynależności $m_C = \max [w1 * m_{C1}, w2 * m_{C2}]$.

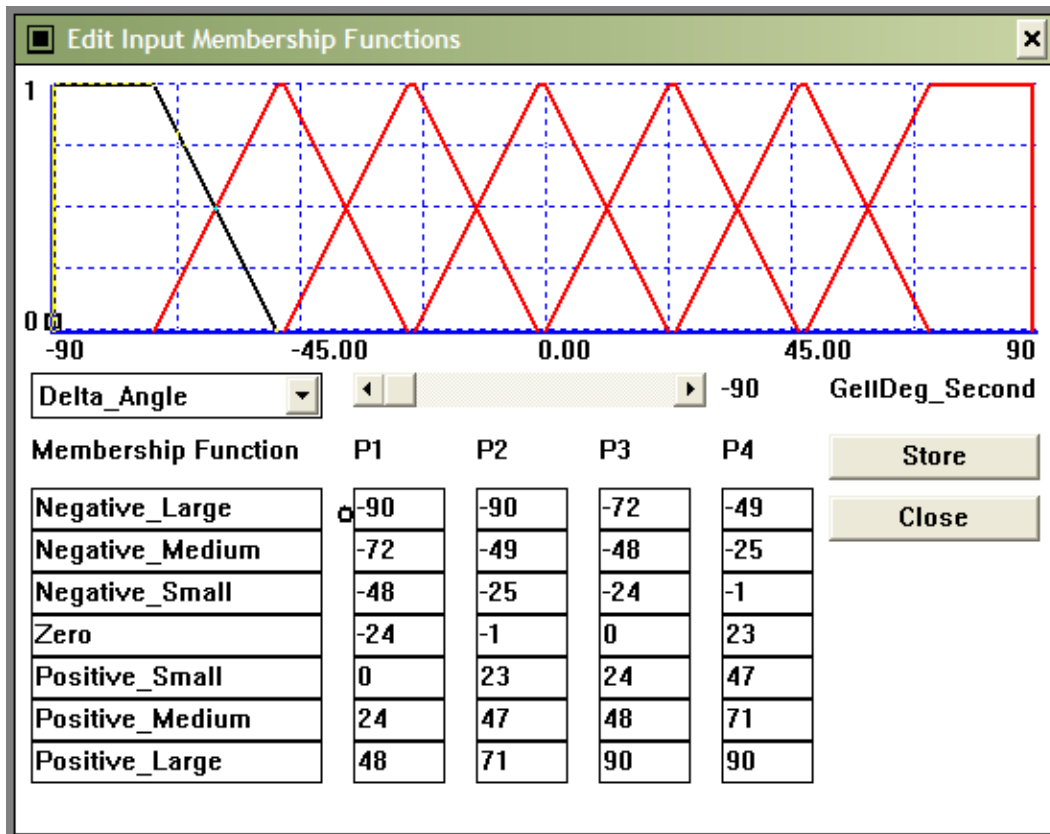
Wnioskowanie metodą Takagi – Sugeno: metoda ta wprowadza w miejsce następników rozmytych następniki typu funkcyjnego. Zaletą tego modelowania jest możliwość wykorzystania analizy liniowych układów dynamicznych w systemie oraz nie wymaga zastosowania procesu wyostrzania w celu obliczenia wyniku wyjściowego.

Wyostrzanie (defuzyfikacja)

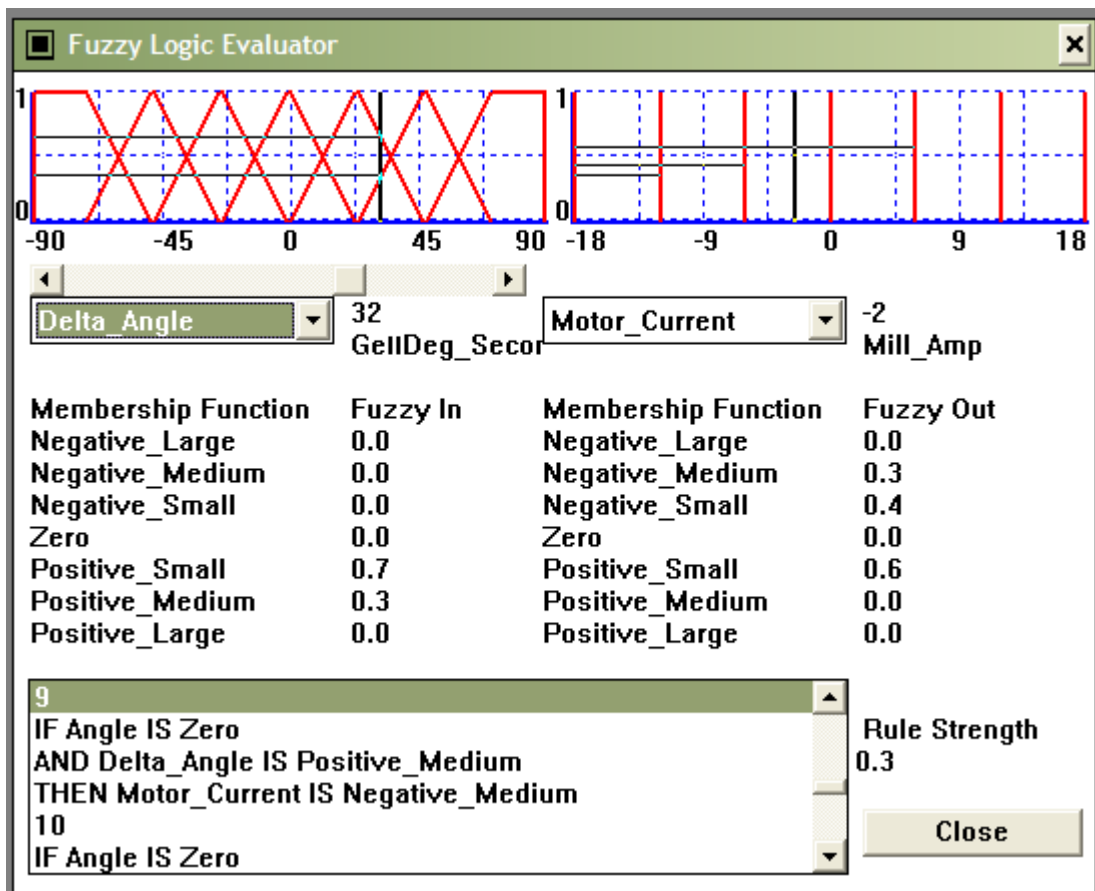
Jest to operacja przekształcenie za pomocą odpowiednio wybranej metody wynikowego zbioru rozmytego na określoną wartość rzeczywistą będącą wartością wyjścia modelu.

3. REALIZACJA STEROWANIA ROBOTEM

Do syntezy sterownika rozmytego wykorzystano program FUDGE firmy Motorola. Oprogramowanie umożliwia dobór zmiennych lingwistycznych dla poszczególnych wejść sterownika (rys.4), opracowanie bazy reguł i sprawdzenie funkcjonowania (okno „evaluatora” – rys.5.). Na podstawie opracowanej bazy reguł i wartości zmiennych lingwistycznych program może wygenerować plik źródłowy dla mikrokontrolerów firmy Motorola lub w języku C (standard ANSI). Przykładowy kod, wygenerowany przez program (dla wybranego algorytmu sterowania robotem - opracowanego w ramach pracy) przedstawiono na końcu rozdziału. Po skompilowaniu kodu źródłowego, plik wynikowy może być wysłany do sterownika RCX i zapewni sterowanie robotem według opracowanych reguł rozmytych.



Rys. 4. Edycja zmiennej „zmiana kąta”



Rys. 5. Dostrajanie regulatora rozmytego

PODSUMOWANIE

Przedstawione w pracy systemy programowania robotów z zestawu Lego MindStorms pozwalają na zastosowanie logiki rozmytej do wyznaczania wartości sterującej (napięcie zasilające silnik robota). Największe możliwości, związane z dopuszczalną liczbą zmiennych w programie i zakresem dostępnych operacji, oferuje system LegOS. Przy wykorzystaniu standardowych narzędzi firmy Lego można implementować jedynie bardzo proste algorytmy sterowania, utrudnione jest także oprogramowanie dodatkowych czujników i elementów wykonawczych. Stosunkowo duże możliwości zapewnia język NQC. Ze względu na składnię tego języka, zbliżoną do języka C możliwe jest proste przeniesienie do tego systemu (po niewielkich modyfikacjach) kodu źródłowego, wygenerowanego przez program FUDGE. Powiązanie programu FUDGE i NQC pozwala na szybkie projektowanie i uruchamianie oprogramowania sterującego dla robotów LEGO.

BIBLIOGRAFIA

1. D. Driankov, H. Hellendoorn, M. Reinfrank : *Wprowadzenie do sterowania rozmytego*, WN-T, Warszawa 1996.
2. A. Piegat: *Modelowanie i sterowanie rozmyte*, Akademicka Oficyna Wydawnicza, Warszawa 1999
3. M. Ferrari, G. Ferrari: *Building Robots With Lego Mindstorms : The Ultimate Tool for Mindstorms Maniacs*, Syngress Publishing 2001r.
4. Łukasik Z., Pniewska B., Pniewski R.: *Laboratorium Komputerowych Systemów Sterowania*, Wydawnictwo Politechniki Radomskiej, Radom 2004.
5. *Zastosowanie logiki rozmytej do wyznaczania optymalnej trajektorii ruchu robota mobilnego* - praca badawcza, 2004/2006. Kierownik pracy B. Pniewska

FUZZY LOGIC CONTROL LEGO ROBOT

Abstract

The teaching of automation and robotics, to achieve minirobots used LEGO Mindstorms. Originally were intended to implement simple systems and simple control algorithms (using the included graphical software.) Since these sets have been used at many universities created a number of software programs for these sets, independent of LEGO. These systems can implement complex control algorithms (using different programming languages). The article shows the possibility of using fuzzy logic to implement intelligent control algorithms.

Autorzy:

dr inż. **Beata Pniewska** – Uniwersytet Technologiczno-Humanistyczny w Radomiu, Wydział Transportu i Elektrotechniki, e-mail: b.pniewska@uthrad.pl

dr inż. **Roman Pniewski** – Uniwersytet Technologiczno-Humanistyczny w Radomiu, Wydział Transportu i Elektrotechniki, e-mail: r.pniewski@uthrad.pl