

## METODY STEROWANIA INTELIGENTNEGO W OPROGRAMOWANIU LABVIEW

## METHODS OF INTELLIGENT CONTROL IN LABVIEW PROGRAMMING

Józef Lisowski<sup>1\*</sup>, Łukasz Sokołowski<sup>2</sup>

Akademia Morska w Gdyni, Morska 81–87, 81-581 Gdynia, Wydział Elektryczny

<sup>1</sup> Katedra Automatyki Okrętowej, e-mail: j.lisowski@we.am.gdynia.pl

<sup>2</sup> Studium Doktoranckie, e-mail: lsokolowski84@gmail.com

\* Adres do korespondencji/Corresponding author

**Streszczenie:** W pracy przedstawiono zasady i przykłady trzech najczęściej stosowanych metod sterowania inteligentnego z wykorzystaniem zestawu narzędziowego ICTL (*Intelligent Control Toolkit for LabVIEW*) – regulatora rozmytego fuzzy logic FLC, regulatora neuro-nowo-rozmytego NF oraz programowania genetycznego GA.

**Słowa kluczowe:** sterowanie rozmyte, sztuczne sieci neuronowe, algorytmy genetyczne, systemy ekspertowe.

**Abstract:** The paper presents the principles and examples of the three most commonly used methods of intelligent control using the toolkit ICTL (*Intelligent Control Toolkit for LabVIEW*) – fuzzy controller FLC fuzzy logic, neuro-fuzzy regulator NF and genetic programming GA.

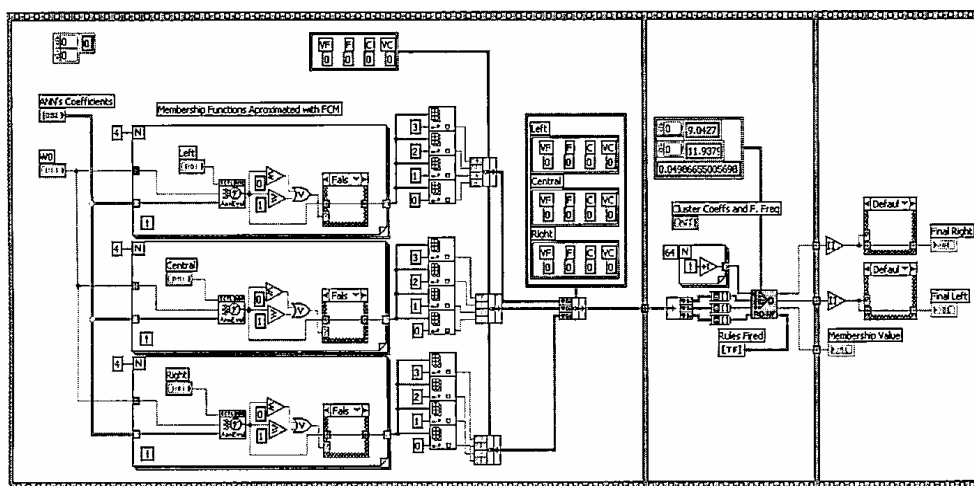
**Keywords:** fuzzy control, artificial neural network, genetic algorithms, expert systems.

### 1. WSTĘP

W celu usprawnienia sterowania procesami przemysłowymi, w tym okrętowymi, wprowadzono metody sterowania inteligentnego, oparte na predykcji ludzkich zachowań i naturalnych procesów w przyrodzie. Obecnie do tworzenia aplikacji inżynierskich wykorzystuje się oprogramowanie LabVIEW (z ang. *Laboratory Virtual Instrument Engineering Workbench*), umożliwiające przeprowadzenie symulacji zaprojektowanego przez użytkownika układu sterowania tak, aby przyspieszyć proces testowania i implementacji w docelowym środowisku [Bitter, Mohiuddin i Nawrocki 2015; Blume 2015; Chruściel 2008].

Sterowanie inteligentne imituje charakterystyki statyczne i dynamiczne obiektów oraz procesów technicznych. Układy sterowania inteligentnego opierają się na poszukiwaniu metod sterowania, które zapewnią wysoki stopień autonomii i inteligencji, pozwalający na osiągnięcie samodzielności podejmowania decyzji przez aplikację i jednocześnie usprawnienie niezawodności oraz skuteczności sterowania procesem [Sokołowski 2015]. Programy *Vis* (*Virtual Instruments*), symulujące rzeczywiste układy, składają się z trzech podstawowych części: frontowego panelu, który stanowi replikę panelu rzeczywistego urządzenia, jego schematu blokowego oraz złącza ikon [NI 2009; Nise 2015; Taczała 2014].

W artykule przedstawiono trzy najczęściej stosowane w praktyce metody sterowania inteligentnego w środowisku LabVIEW – sterowania rozmytego, neuronowo-rozmytego i programowania genetycznego, z użyciem zestawu narzędziowego ICTL (z ang. *Intelligent Control Toolkit for LabVIEW*) [ITESM Mexico 2007] (rys. 1).

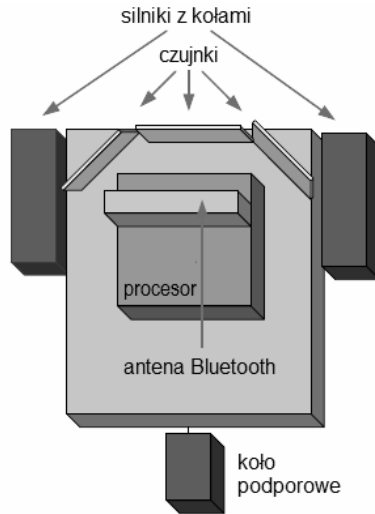


Rys. 1. Schemat zestawu narzędziowego *Intelligent Control Toolkit for LabVIEW* [Ponce-Cruz i Ramirez-Figureoa 2010]

Fig. 1. Block diagram from *Intelligent Control Toolkit for LabVIEW* [Ponce-Cruz and Ramirez-Figureoa 2010]

## 2. REGULATOR FLC (*FUZZY LOGIC CONTROLER*)

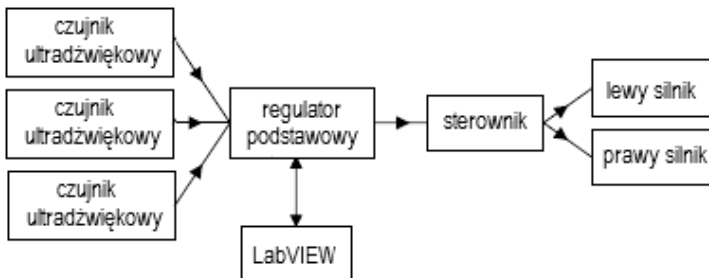
Przykładem zastosowania regulatora rozmytego FLC jest sterowanie robotem kołowym. Przy użyciu zestawu narzędziowego ICTL można zaprogramować robota do pracy za pomocą jednego z trzech regulatorów: Mamdani, Takagi-Sugeno lub Tsukamoto. Implementację technologii FLC w modelu mobilnego robota przedstawiono na rysunku 2.



**Rys. 2.** Części składowe trójkołowego robota WHEEL  
[Ponce-Cruz i Ramirez-Figureoa 2010]

**Fig. 2.** Three-wheeled robot WHEEL parts  
[Ponce-Cruz and Ramirez-Figureoa 2010]

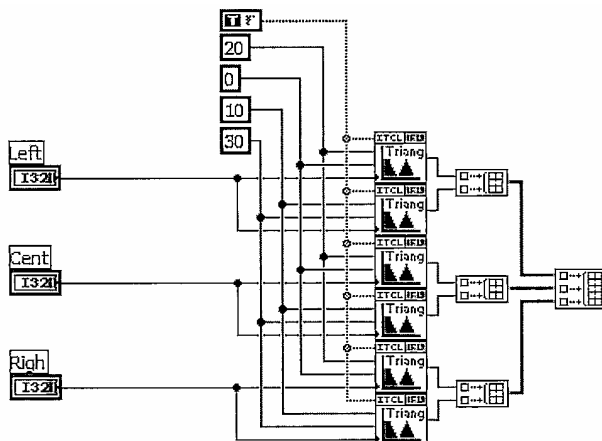
Robot, nazywany WHEEL, poprzez Bluetooth wysyła zmierzoną poprzez trzy czujniki ultradźwiękowe odległość, jako napęd wykorzystuje się dwa serwo-mechanizmy. Silnik jest sterowany poprzez podstawowy regulator Stamp; antena Bluetooth to przetwornik EmbeddedBlue od firmy Parallax, tak samo trzy czujniki i dwa serwomechanizmy (rys. 3).



**Rys. 3.** Schemat blokowy trójkołowego robota

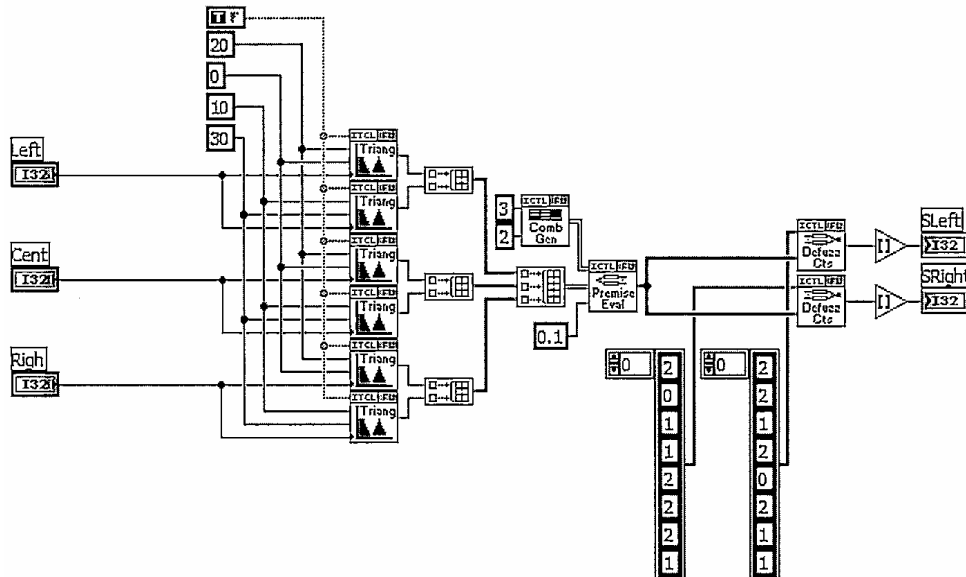
**Fig. 3.** Three-wheeled robot diagram

Do fuzyfikacji (rozmywania) sygnałów wejściowych zastosowano dwie trójkątne funkcje przynależności, następnie utworzono dwuwymiarową tablicę zawierającą 9 reguł, dla trzech wejść – lewo, środkowo, prawo oraz dwóch wyjść – blisko, daleko (rys. 4).



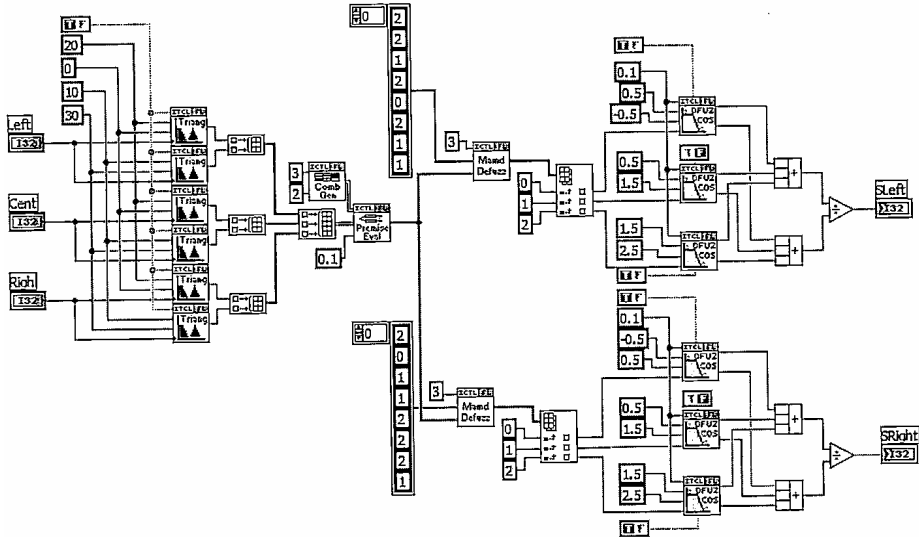
**Rys. 4.** Schemat blokowy realizacji funkcji przynależności [Ponce-Cruz i Ramirez-Figureoa 2010]  
**Fig. 4.** Block diagram of the membership function [Ponce-Cruz and Ramirez-Figureoa 2010]

Następnym etapem było wyznaczenie reguł i defuzyfikacja (wyostrzenie). Aby otrzymać wyostrzoną wartość na wyjściu do zastosowania w regulatorze, należy użyć wyjściowych funkcji zbiorów jednoelementowych bądź wnioskowania Takagi-Sugeno (rys. 5).



**Rys. 5.** Schemat blokowy regulatorów: Takagi-Sugeno oraz Tsukamoto [Ponce-Cruz i Ramirez-Figureoa 2010]  
**Fig. 5.** Block diagram of the Takagi-Sugeno and Tsukamoto controllers [Ponce-Cruz and Ramirez-Figureoa 2010]

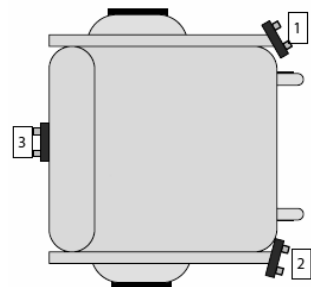
Dla regulatora Mamdani należy zastosować trzy funkcje trójkątne do defuzyfikacji procesów ruchu robota w stanach – lewo, prawo i zatrzymanie (rys. 6).



**Rys. 6.** Schemat blokowy regulatora Mamdani [Ponce-Cruz i Ramirez-Figureoa 2010]  
**Fig. 6.** Block diagram of a Mamdani controller [Ponce-Cruz and Ramirez-Figureoa 2010]

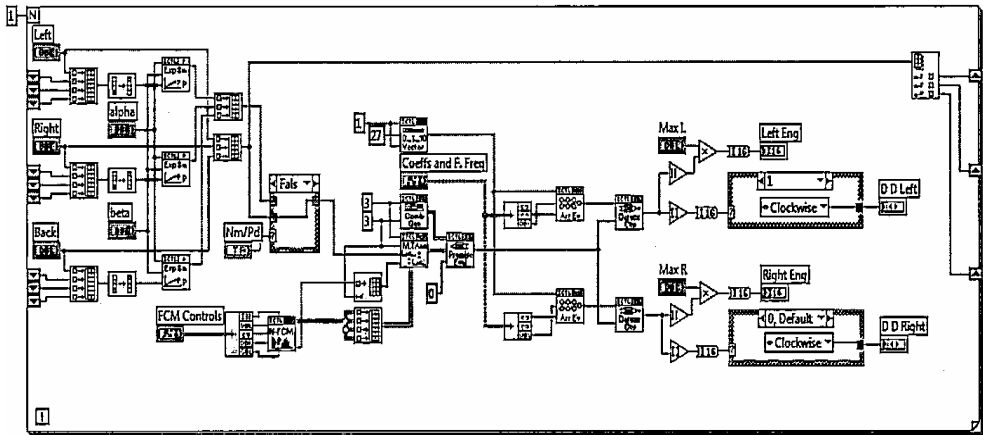
### 3. REGULATOR NF (NEURO-FUZZY)

Przykładem sterowania inteligentnego w LabVIEW jest regulator neuronowo-rozmyty zastosowany do sterowania autonomicznym wózkiem inwalidzkim tak, aby unikać stałych i ruchomych przeszkód. Regulator zbiera informacje z trzech czujników ultradźwiękowych, rozmieszczonych jak na rysunku 7 i mierzących odległości do przeszkód, oraz wybiera najlepszą trajektorię ruchu wózka w celu ich uniknięcia.



**Rys. 7.** Elektryczny wózek inwalidzki z czujnikami odległości: 1 – lewy, 2 – prawy, 3 – tylny [Ponce-Cruz i Ramirez-Figureoa 2010]  
**Fig. 7.** The electric wheelchair with distance sensors: 1 – left, 2 – right, 3 – back [Ponce-Cruz and Ramirez-Figureoa 2010]

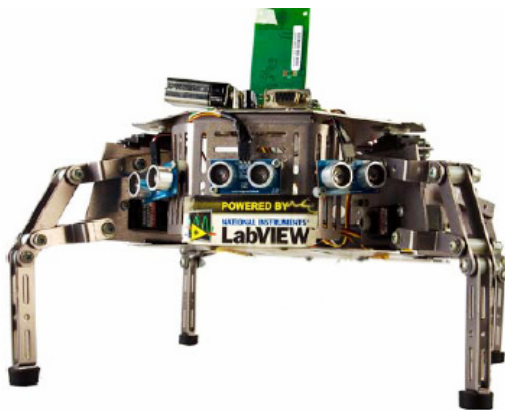
Wyjściami regulatora są napięcia przesyłane do układu, który generuje sygnał z modulacją szerokości impulsów PWM (z ang. *Pulse Width Modulation*), sterujący silnikami napędowymi i ustalający kierunek ich obrotu. Regulator jest oparty na trygonometrycznych sieciach neuronowych T-ANNS (z ang. *Trigonometric Artificial Neural Networks*) i zbiorach rozmytych oraz wykorzystuje metodę wnioskowania Takagi-Sugeno [ITESM Mexico 2007] (rys. 8).



**Rys. 8.** Schemat regulatora neuronowo-rozmytego [Ponce-Cruz i Ramirez-Figureoa 2010]  
**Fig. 8.** Neuro-fuzzy controller block diagram [Ponce-Cruz and Ramirez-Figureoa 2010]

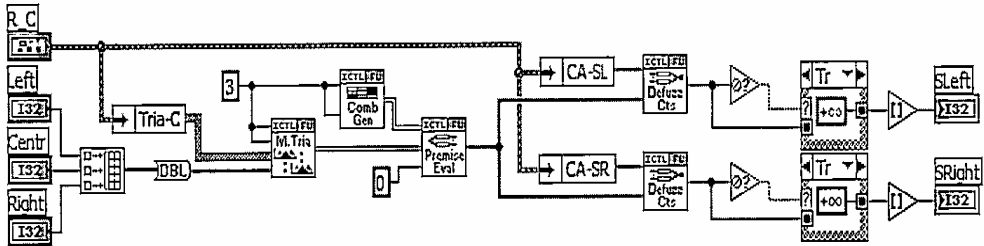
#### 4. ALGORYTM GENETYCZNY GA (*GENETICAL ALGORITHM*)

W zestawie narzędziowym ICTL zastosowano programowanie genetyczne do optymalizacji systemu nawigacyjnego robota mobilnego, pokazanego na rysunku 9.



**Rys. 9.** Robot ZIL I sterowany regulatorem z logiką rozmytą dostrajaną za pomocą algorytmu genetycznego [Ponce-Cruz i Ramirez-Figureoa 2010]  
**Fig. 9.** Robot ZIL I controlled by a fuzzy logic controller adjusted using genetic algorithm [Ponce-Cruz and Ramirez-Figureoa 2010]

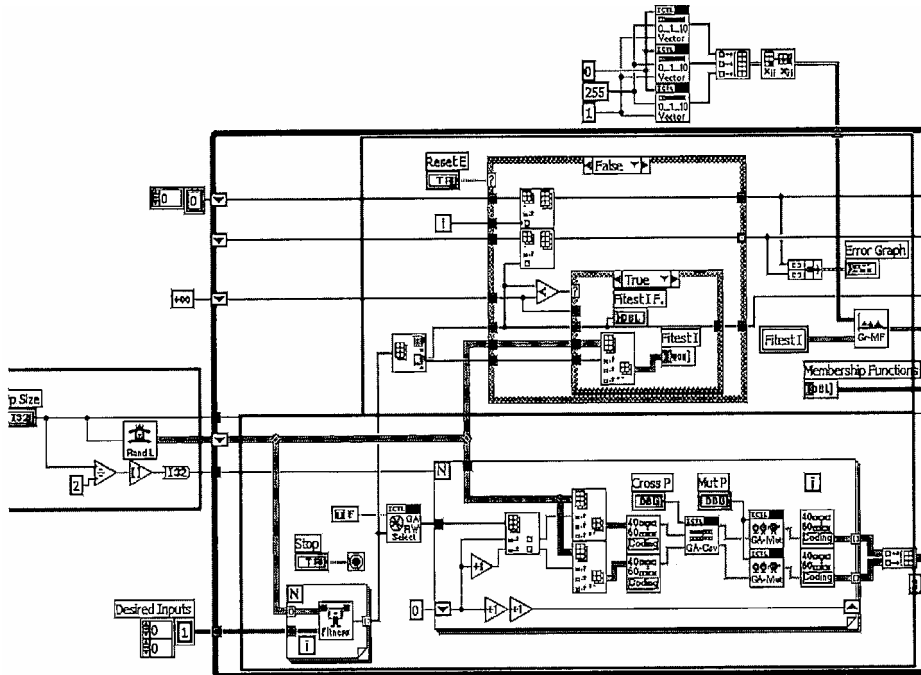
System nawigacyjny czteroosobnego robota ZIL I, zbudowany w technologii Bluetooth, zaprojektowano do pracy w nieznanym otoczeniu. Robot wyposażony w czujniki ultradźwiękowe, pozwalające omijać zarówno przeszkody stałe jak i ruchome, jest sterowany przez regulator rozmyty Takagi-Sugeno, którego budowę pokazano na rysunku 10.



Rys. 10. Schemat blokowy regulatora Takagi-Sugeno [Ponce-Cruz i Ramirez-Figureoa 2010]

Fig. 10. Block diagram of the Takagi-Sugeno controller [Ponce-Cruz and Ramirez-Figureoa 2010]

Logika rozmyta regulatora jest dostrajana za pomocą algorytmu genetycznego GA. Niektóre warunki początkowe zostały określone z wykorzystaniem wiedzy eksperckiej, inne zaś przyjęto losowo (rys. 11).



Rys. 11. Schemat blokowy algorytmu genetycznego GA [Ponce-Cruz i Ramirez-Figureoa 2010]

Fig. 11. Block diagram of the genetic algorithm GA [Ponce-Cruz and Ramirez-Figureoa 2010]

## 5. PODSUMOWANIE

LabVIEW jest środowiskiem programowania wyposażonym w interfejs graficzny. Przedstawia wirtualne instrumenty jako obrazowe reprodukcje urządzeń sprzętowych, takich jak generator oscyloskopu lub przebiegów sygnałów sterujących, wyjściowych i zmiennych stanu procesu automatycznej regulacji. Umożliwia syntezę klasycznych, optymalnych, adaptacyjnych i inteligentnych układów sterowania wykorzystujących systemy ekspertowe, rozmyte, neuronowe i genetyczne.

## LITERATURA

- Bitter, R., Mohiuddin, T., Nawrocki, M., 2015, *LabVIEW Advanced Programming Techniques*, CRC Press, California.
- Blume, P.A., 2015, *The LabVIEW Style Book*, Prentice Hall International, London.
- Chruściel, M., 2008, *LabVIEW w praktyce*, Wydawnictwo BTC, Warszawa.
- ITESM Mexico, 2007, *Intelligent Control Toolkit for LabVIEW*.
- NI, *LabVIEW control design user manual*, National Instruments Corp., Austin 2009.
- Nise, N.S., 2015, *Control Systems Engineering*, Wiley, Pomona.
- Ponce-Cruz, P., Ramirez-Figuroa, F.D., 2010, *Intelligent Control Systems with LabVIEW*, Springer-Verlag, London.
- Sokołowski, Ł., 2015, *Analiza układów sterowania inteligentnego w oprogramowaniu LabVIEW*, praca magisterska, Akademia Morska, Gdynia, [maszynopis niepublikowany].
- Taczała, W., 2014, *Środowisko LabVIEW w eksperymencie wspomaganym komputerowo*, WNT, Warszawa.