

doi:10.15199/48.2018.07.19

Implementacja systemu rozmytego przeznaczonego do sterowania instalacją centralnego ogrzewania

Streszczenie. Artykuł opisuje implementację systemu rozmytego w mikrokontrolerze, dedykowanego dla instalacji centralnego ogrzewania. System ten dedykowany jest do automatyzacji pracy urządzeń instalacji w zależności od ustawień użytkownika. W tym celu dedykowana architektura kontrolera (sterownika) logiki rozmytej została opracowana. System ten, posiada 3 niezależne wejścia i 2 wyjścia; oraz wbudowane 3 wewnętrzne bloki fuzyfikacji, wnioskowania i wyostrzania.

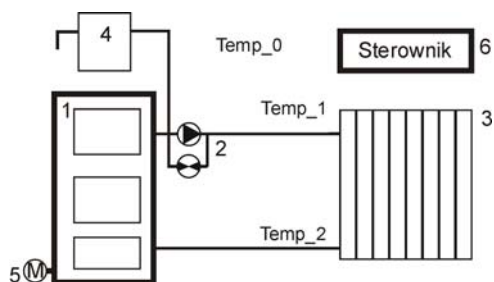
Abstract. This paper describes an implementation of a fuzzy system in the microcontroller, dedicated for the central heating installation. This system is dedicated to automate work of devices of heating installation depending on user settings. For this purpose, a dedicated architecture of a fuzzy logic controller system was elaborated. This system has 3 independent inputs and 2 outputs and is composed of 3 internal blocks: fuzzification, inference and defuzzification. (**Implementation of a Fuzzy System Dedicated for the Central Heating Installation**).

Słowa kluczowe: system rozmyty, mikrokontroler, sterownik, ogrzewanie.

Keywords: fuzzy system, microcontroller, controller, heating .

Wstęp

Sterowanie pracą kotła stałopalnego małej mocy dedykowanego na potrzeby domowej instalacji ogrzewania sprowadza się do utrzymywania zadanej temperatury w instalacji grzewczej na podstawie nastaw użytkownika oraz warunków zewnętrznych [1][2].

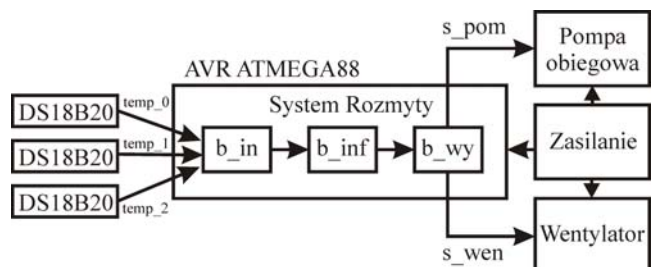


Rys.1. Uproszczony schemat blokowy domowej instalacji kotła stałopalnego, gdzie: 1 – kocioł stałopalny, 2 – pompa wraz z zaworem bezpieczeństwa, 3 – grzejnik (obciążenie), 4 – naczynie przelewowe, 5 – wentylator, 6 – sterownik, Temp_x – odczyty temperatur (gdzie x=0,1,2).

Uproszczony schemat blokowy domowej instalacji kotła stałopalnego przedstawiono na rys. 1. W strukturze tego schematu można wyróżnić podstawowe elementy instalacji. Źródłem ciepła jest kocioł stałopalny (1), który podłączony jest do instalacji grzewczej – grzejników (3), poprzez pompę cyrkulacyjną zadaniem której jest regulacja przepływu wody w instalacji (2). Pomiaru temperatury dokonuje się w oparciu o zainstalowane czujniki temperatury. W tym celu dokonuje się pomiarów temperatury otoczenia – Temp_0, temperatury zasilania – Temp_1 oraz temperatury powrotu – Temp_2. Sterownik utrzymuje zadaną temperaturę tak długo jak paliwo stałe znajduje się w komorze spalania kotła stałopalnego. W zależności od rodzaju paliwa stałopalnego, ogrzewanej kubatury i warunków zewnętrznych proces spalania może trwać od kilku do kilkunastu godzin. Regulacja nadmuchu dopływu powietrza do komory spalania powoduje zmiany ilości tlenu, a to zaś powoduje zmiany intensywności procesu spalania. Zwiększanie ilości powietrza, które dopływa do komory spalania powoduje zwiększenie ilości tlenu w komorze. Przekłada się to docelowo na wzrost temperatury, tak długo jak w komorze spalania znajduje się paliwo stałe. Analogicznie, ograniczenie ilości tlenu w komorze powoduje ograniczenie

procesu spalania. Skrajne zamknięcie dopływu powietrza powoduje wygaszenie kotła. W strukturze blokowej przedstawionej na rys. 1 wyróżnić należy naczynie przelewowe (4). Jego zadaniem jest zabezpieczenie kotła przed niekontrolowanym wzrostem ciśnienia. Zawór obejściowy typu „bypass” połączony równolegle do pompy cyrkulacyjnej (2) zabezpiecza instalację w przypadku awarii pompy lub braku zasilania. Wspomniane elementy dedykowane są do podniesienia bezpieczeństwa eksploatacji domowej instalacji grzewczej.

W artykule przedstawiono opis sterownika dedykowanego do sterowania małą (prostą) instalacją centralnego ogrzewania, który został zrealizowany w oparciu o implementację systemu rozmytego w układzie mikrokontrolera [3]. Na rys. 2 przedstawiono uproszczony schemat blokowy proponowanego sterownika. Głównym elementem sterownika jest układ mikrokontrolera - AVR Atmega88. Mikrokontroler ten współpracuje ze źródłami sygnałów wejściowych zrealizowanych w oparciu o czujniki typu DS18B20 oraz dwoma elementami wykonawczymi tj.: pompą obiegową oraz wentylatorem. W strukturze wewnętrznej mikrokontrolera zaimplementowano system rozmyty Mamdaniego typu MIN-MAX, wraz z bazą wiedzy. Bazę wiedzy stanowią funkcje przynależności wejściowych i wyjściowych zmiennych lingwistycznych oraz baza reguł. Zadanie sterownika sprowadza się do generowania sygnałów sterujących pracą pompy (s_pom) oraz wentylatora (s_wen) na podstawie wejściowych sygnałów temp_x (gdzie x=0,1,2), pochodzących z czujników temperatur.

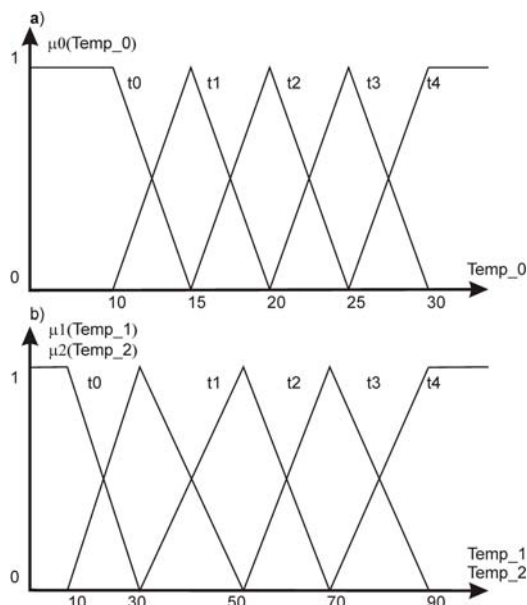


Rys.2. Uproszczony schemat blokowy sterownika domowej instalacji kotła stałopalnego, gdzie: DS18B20 – czujnik temperatury, temp_x – sygnały temperatury (gdzie x=0,1,2), b_in – blok rozmywania, b_inf – blok wnioskowania, b_wy – blok wyostrzania, s_pom – sygnał typu PWM pompy obiegowej, s_wen – sygnał typu PWM wentylatora.

W trakcie pracy kotła można wyróżnić następujące charakterystyczne stany pracy sterownika tj. rozpalenie, praca, przegrzanie, gaszenie oraz czuwanie.

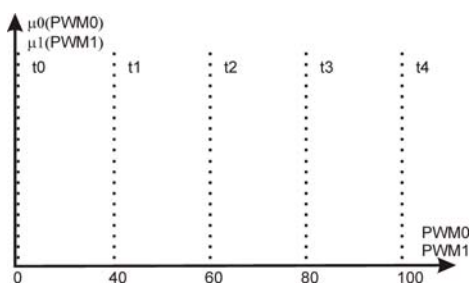
Zmienne lingwistyczne

Na rys.3 przedstawiono wejściowe zmienne lingwistyczne odpowiadające sygnałom temperatur Temp_0 - Temp_2; każda z tych zmiennych opisana jest przy pomocy 5 zbiorów rozmytych o kształcie trójkątnym (t0-t4). Zakres zmian temperatury otoczenia zwiera się w zakresie ~10-30 [°C]. Dla przykładu ostrej wartości temperatury otoczenia równej 20°, odpowiada pełna wartość funkcji przynależności do zbioru rozmytego t2. W przypadku temperatur zasilania i powrotu zakres ich zmian zawiera się w zakresie ~10-90 [°C].



Rys.3. Wejściowe zmienne lingwistyczne a) temperatury otoczenia – Temp_0, b) temperatury zasilania – Temp_1, temperatury powrotu – Temp_2.

Wyjściowe zmienne lingwistyczne przedstawiono na rys. 4. W proponowanym sterowniku proces regulacji prędkości silników napędowych pompy cyrkulacyjnej oraz wentylatora przebiega w oparciu o modulację szerokości impulsu – PWM. Wartości odpowiadające współczynnikowi wypełnienia korespondują do wyjściowych zmiennych lingwistycznych – PWM0 oraz PWM1.



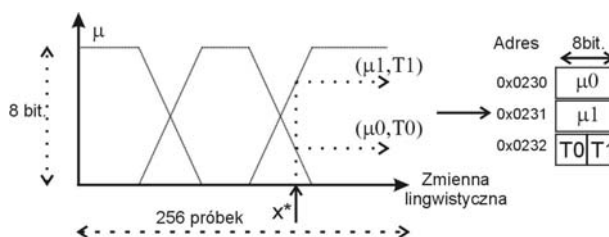
Rys.4. Wyjściowe zmienne lingwistyczne PWM0, PWM1 dedykowane do sterowania pracą pompy cyrkulacyjnej oraz wentylatora.

System rozmyty

Proponowany system rozmyty został zaimplementowany w mikrokontrolerze jednoukładowym AVR Atmega88. Kod programu mikrokontrolera napisano w języku C w oparciu o algorytm wnioskowania typu

Mamdaniego – MIN/MAX. Sygnały temp_0 – temp_2 dołączono do wejść przetworników A/C – ADC0 – ADC2, sygnały wyjściowe s_pom oraz s_wen do wejść PD4,PD5.

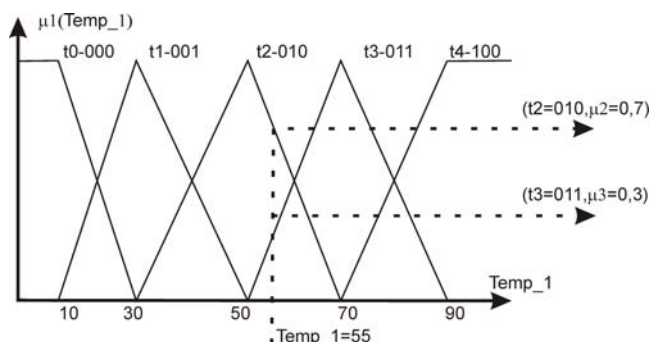
W sterowniku system rozmyty został zrealizowany w oparciu o podział na bloki rozmywania, wnioskowania oraz wyostrzania [4]. Proces rozmywania (fuzyfikacji) polega na sekwencyjnej zmianie wejściowych wartości z dziedziny liczb rzeczywistych na wartość funkcji przynależności μ z dziedziny zbiorów rozmytych. W systemie tym wejściowe zmienne lingwistyczne (Temp_0, Temp_1, Temp_2) opisano 5 termami (zbiorami rozmytymi) a następnie zapisano w pamięci mikrokontrolera w postaci 256 - 3 bajtowych próbek (rys. 5).



Rys.5. Przykład opisu zmiennej lingwistycznej.

Pojedyncza próbka przechowuje wartość dwóch par tj. (μ_0, T_0) oraz (μ_1, T_1). Na podstawie wejściowej wartości dyskretnej x^* zmiennej lingwistycznej (wejścia temperatur) z pamięci systemu odczytywane są wartości funkcji przynależności dwóch zbiorów rozmytych oraz kody aktywowanych zbiorów rozmytych tj. (T_0, μ_0) oraz (T_1, μ_1) dla każdej ze zmiennych lingwistycznych. W przypadku, gdy jedna z wartości funkcji przynależności przyjmuje wartość zerową ($\mu=0$), to w procesie wnioskowania zostaje ona pominięta.

Wejściowe zmienne lingwistyczne opisane są przy pomocy 5 zbiorów rozmytych, których kolejne wartości przedstawione są w postaci wartości binarnych. Na rys.6 przedstawiono przykład ostrej wartości wejściowej zmiennej lingwistycznej Temp_1=55, oraz odpowiadające im dwie pary zawierające kody zbiorów rozmytych oraz wartości funkcji przynależności - ($t_2=010, \mu_2=0,7$), ($t_3=011, \mu_3=0,3$). Proces zamiany ostrej wartości na wartość rozmytą analogicznie przebiega podobnie dla pozostałych zmiennych lingwistycznych.



Rys.6. Przyjęta zasada kodowania zbiorów rozmytych, z zaznaczoną przykładową wartością ostrą.

Następnie wartości rozmyte wejściowych zmiennych lingwistycznych przekazywane są do bloku realizującego proces wnioskowania. Przekazanie wartości następuje w postaci 3 par aktywnych zbiorów rozmytych – gdzie pojedyncza para wygląda następująco - (T_0, μ_0) oraz (T_1, μ_1). Wartości T_0/T_1 odpowiadają aktywowanym zbiorom rozmytym, a wartości μ_0/μ_1 odpowiadającym im

wartością funkcji przynależności. W bloku tym następuje fizyczne sformowanie adresu pod którym zapisano kod wyjściowego zbioru rozmytego oraz obliczenie stopnia zapłonu poszczególnych aktywowanych reguł rozmytych. W trakcie procesu wnioskowania wykonywane są sekwencyjnie operacje formułowania adresu, odczytu wartości ze wskazanego adresu, operacją wyboru najmniejszej wartości – MIN oraz operacją częściowej agregacji typu MAX. Wszystkie wspomniane operacje wykonywane są wyłącznie na podstawie aktywnych wejściowych zbiorów rozmytych.

W pamięci mikrokontrolera zapisano konkluzje reguł rozmytych zbudowanych w oparciu o przesłanki proste. W procesie wnioskowania poszczególne przesłanki proste odpowiadają aktywnym kodom zbiorów rozmytych. Poniżej widoczna jest ogólna reguła kontrolująca pracę silników (pompy i wentylatora):

Ra: jeżeli (T0 = A0) AND (T1 = A1) AND (T2 = A2)
to (y0 = B0)

Rb: jeżeli (T0 = A0) AND (T1 = A1) AND (T2 = A2)
to (y1 = B1)

gdzie:

x0,x1,x2 – wejściowe zmienne lingwistyczne odpowiadające sygnałom temperatury

A0,A1,A2 - termy zmiennych lingwistycznych xi (i=0,1,2),

y0,y1 – wyjściowe zmienne lingwistyczne, odpowiadające sygnałom PWM0 oraz PWM1,

B0,B1 - termy zmiennych lingwistycznych yi (i =0,1)

Liczba reguł jaka opisuje system rozmyty zależy od liczby wejściowych zmiennych lingwistycznych – N, liczby wyjściowych zmiennych lingwistycznych – K, oraz od liczby zbiorów rozmytych opisujących te zmienne – L. Wzrost liczby zbiorów rozmytych powoduje znaczący wzrost całkowitej liczby reguł. W tabeli 1 przedstawiono zestawienie liczby reguł w zależności od liczby L,N i K.

Tabela 1. Przykładowe zestawienie całkowitej liczby reguł opisujących system rozmyty dla systemu o liczbie zbiorów rozmytych L=3,5,7; liczbie wejściowych zmiennych lingwistycznych N=1,2,3,4,5 dla systemu o liczbie wyjść K=1.

L\N	1	2	3	4	5
3	3	9	27	81	243
5	5	25	125	625	3125
7	7	49	343	2401	16807

W proponowanym systemie rozmytym każdorazowo aktywowane są dwa zbiory rozmyte, tym samym całkowita liczba aktywowanych reguł wynosi $K \cdot 2^N$ (w trakcie jednego pełnego procesu wnioskowania). W tabeli 2 przedstawiono zestawienie liczby tych reguł.

Tabela 2. Przykładowe zestawienie całkowitej liczby reguł aktywowanych w pojedynczym procesie wnioskowania, dla systemu rozmytego o liczbie zbiorów rozmytych L=2; liczbie wejściowych zmiennych lingwistycznych N=1,2,3,4,5 dla systemu o liczbie wyjść K=1.

L\N	1	2	3	4	5
3	2	4	8	16	32
5	2	4	8	16	32
7	2	4	8	16	32

Przedstawione zestawienie wartości całkowitej liczby reguł – Tabela 1 oraz odpowiadające im liczby reguł aktywowanych w procesie wnioskowania – Tabela 2, przy założeniu aktywowania wyłącznie dwóch zbiorów rozmytych, wskazuje, że można przetwarzać wyłącznie aktywne reguły. W kolejnych punktach poniżej przedstawiono sposób odczytu wyłącznie aktywnych reguł.

Tabela 3. Tablica generowania adresów konkluzji na podstawie licznika pomocniczego o wagach b2,b2,b0 oraz odczytanych kodach konkluzji

b2	b1	b0	μ_2	μ_1	μ_0	Odczytane kody			Adres
0	0	0	70	84	11	010	000	100	010000100
0	0	1	70	84	89	010	000	011	010000011
0	1	0	70	16	11	010	001	100	010001100
0	1	1	70	16	89	010	001	011	010001011
1	0	0	30	84	11	011	000	100	011000100
1	0	1	30	84	89	011	000	011	011000011
1	1	0	30	16	11	011	001	100	011001100
1	1	1	30	16	89	011	001	011	011001011

W proponowanym systemie całkowita liczba reguł wynosi $2 \cdot 5^3 = 250$ reguł lub 125 par reguł. Każdorazowo w procesie wnioskowania aktywowane jest zaledwie 16 reguł lub 8 par reguł. Proces formowania adresów konkluzji – generowania adresów przebiega w oparciu o wykorzystanie wartości 3 bitowego licznika pomocniczego o wagach b2b1b0. Wykorzystanie tego licznika powoduje, że generowane są wszystkie kombinacje kodów wyjściowych zbiorów rozmytych począwszy od wartości binarnej 000 do 111. Wartości wag licznika mogą przyjmować dwie wartości tj. 0 lub 1, które odpowiadają aktywowanym zbiorom rozmytym. Kody te wprowadzane są na podstawie wartości uzyskanych w procesie rozmywania. Z racji tego, że wykorzystano 3 bitowy licznik, uzyskuje się w ten sposób wszystkie kombinacje kodów zbiorów rozmytych korespondujące do poszczególnych wejściowych zmiennych lingwistycznych. Uzyskane kody są następnie sklejane, tworząc adres, całkowity fizyczny adres uzyskuje się po dodaniu tzw. offsetu. W tabeli 3 przedstawiono zestawienie wszystkich kombinacji wejściowych kodów zbiorów rozmytych. Celem lepszego wyjaśnienia, należy posłużyć się przykładem: Jeżeli b2=0, b1=1, b0=0 a odczytane kody zbiorów wynoszą 010, 001, 100 to w wyniku sklejania otrzyma się adres 010001100. Zmiana licznika do wartości b2=0, b1=1, b0=1, spowoduje odczytanie kodów zbiorów 010, 001, 011, a uzyskany adres wyniesie 010001011. Warto podkreślić, że zmiana wartości najmłodszej wagi licznika pomocniczego z 0 na 1, spowodowała zmianę kolejnego adresu. W proponowanym systemie proces wnioskowania ma charakter iteracyjny, który wykonywany jest tyle razy ile jest reguł aktywnych [5][6]. Z racji, że w systemie występują dwie wyjściowe zmienne lingwistyczne pod wskazanym adresem, zapisywane są dwa 3 bitowe kody konkluzji. W proponowanym systemie rozmytym wszystkie reguły w pamięci systemu zapisano pod ściśle określonym adresem w pamięci mikrokontrolera [7][8]. Uzyskane adresy powstały poprzez sklejanie kodów przesłanek prostych, pochodzących z procesu rozmywania w oparciu o licznik pomocniczy [9].

W trakcie pracy systemu, proces wnioskowania w dużym uproszczeniu składa się z sekwencyjnych operacji formowania adresu konkluzji reguł rozmytych oraz obliczania stopnia spełnienia takiej reguły. Równolegle do procesu generowania kolejnych adresów wykonywane są, w takt wartości wag licznika pomocniczego, operacje typu MIN. Operacja MIN polega na wyborze najmniejszej wartości funkcji przynależności z trzech aktywowanych wejściowych par rozmytych w takt wartości wag licznika. Jeżeli b2=0, b1=1, b0=0 a odczytane wartości funkcji przynależności wynoszą 70, 16, 11 to wynik operacji typu MIN wynosi 11. W przypadku, gdy wartość licznika uzyska kolejny stan tj. b2=0, b1=1, b0=1 spowoduje to odczytanie wartości 70, 11, 89 a wartość operacji MIN wyniesie 11. Należy podkreślić, że w trakcie obliczania stopnia zapłonu reguł rozmytych, niektóre reguły mogą być uruchomione

kilkukrotnie. Dlatego po zakończeniu obliczania wszystkich współczynników zapłonu reguł, wykonywana jest operacja agregacji typu MAX, która polega na wyborze największej wartości funkcji przynależności aktywowanych zbiorów wyjściowych. Wspomniana agregacja częściowa polega na nadpisaniu wartości mniejszej, wartością większą. Całkowity proces wnioskowania wykonywany jest 8 razy, tj. tyle razy ile generowane były kolejne adresy konkluzji. Zakończenie procesu wnioskowania, rozpoczyna proces wyostrzania. W procesie tym na podstawie rozmytych wartości funkcji przynależności wyjściowych zbiorów rozmytych obliczana jest ostra wartość wyjściowa, która przekazywana jest do silników wykonawczych. W systemie tym wykorzystano metodę wyostrzania COG dla singletonów.

Weryfikacja sterownika

Proponowany sterownik został podłączony czujnikami temperatury do rzeczywistej instalacji centralnego ogrzewania z zainstalowanym kotłem stałopalnym o mocy 24 [kW] oraz do elementów wykonawczych. Na podstawie ogólnej wiedzy osoby eksploatującej piec wprowadzono wartości funkcji przynależności poszczególnych zmiennych lingwistycznych, a także bazę reguł. Proces dostrajania wymagał wielokrotnej modyfikacji bazy wiedzy. W tym celu wykorzystano środowisko MATLAB wraz z dodatkową biblioteką - Fuzzy Logic Toolbox. Uzyskana w ten sposób baza została zapisana w pamięci mikrokontrolera. W celu kontroli pracy sterownika wszystkie wartości mierzone przez sterownik przesyłane są do komputera klasy PC celem weryfikacji poprawności działania. Dodatkowo w trakcie eksploatacji proponowany system został rozbudowany o moduł sieciowy oraz stosowną aplikację celem podniesienia wartości użytkowych pracy.

Podsumowanie

Proponowany system rozmyty został zrealizowany i wykorzystany do sterowania kotłem centralnego ogrzewania zainstalowanym w domu jednorodzinnym. Wprowadzenie systemu rozmytego pozwoliło na modyfikację bazy wiedzy sterownika (zachowania sterownika). Docelowo warto opracować kilka modeli wiedzy dedykowanych do konkretnych paliw stałopalnych, tak aby móc dostosować proces spalania do jakości paliwa. Dalsze prace w tym zakresie dotyczyć będą, wykorzystania czujników zawartości tlenu w gazach spalinowych pieca i wykorzystania tych wartości do dostosowania modelu wiedzy systemu rozmytego. Z punktu widzenia technicznego warto podkreślić fakt, wykorzystania techniki

adresowania do wyszukiwania aktywnych reguł. W ten sposób uzyskuje się możliwość znaczącego skrócenia procesu wnioskowania systemu rozmytego. Jednak, aby skrócić czas przetwarzania systemu rozmytego, należy projektowane reguły zapisać w pamięci systemu pod ściśle określonym adresem. Adres ten musi być obliczony w trakcie projektowania systemu. Uzupełnienie systemu w moduł zdalnego dostępu podniosło walory użytkowe pracy.

Autorzy: dr inż. Marek Popławski, Politechnika Koszalińska, Wydział Elektroniki i Informatyki, ul. Śniadeckich 2, 75-453 Koszalin, E-mail: marpop@ie.tu.koszalin.pl; dr hab. inż. Adam Słowik, Politechnika Koszalińska, Wydział Elektroniki i Informatyki, ul. Śniadeckich 2, 75-453 Koszalin, E-mail: aslowik@ie.tu.koszalin.pl.

LITERATURA

- [1] Dounis A.I., Manolakis D.E., Design of a fuzzy system for living space thermal-comfort regulation, *Applied Energy*, 69 (2001), n. 2, 119-144
- [2] Hamdi M., Lachiver G., "A fuzzy control system based on the human sensation of thermal comfort," 1998 IEEE International Conference on Fuzzy Systems Proceedings. IEEE World Congress on Computational Intelligence, Anchorage, AK, 1998, pp. 487-492
- [3] Mendi, F., Boran, K., & Kulekci, M. K., Fuzzy controlled central heating system. *International journal of energy research*, 26 (2002), n. 15, 1313-1322
- [4] Patyra M. J., Grantner J. L., "Hardware implementaions of digital fuzzy logic controller", *Elesiver Information Science*, 113 (1999), 19-54
- [5] Deliparaschos K. M., Nenedakis F. I., Tzafestas S. G. "Design and Implementa tion of a Fast Digital Fuzzy Logic Controller Using FPGA Technology", *Journal of Intelligent and Robotic Systems*, 45 (2006), n. 1, 77-96
- [6] Ikeda, H., Kisu, N., Hiramoto, Y. , Nakamura, S., "A fuzzy inference coprocessor using a flexible active-rule-driven architecture", *IEEE International Conference on Fuzzy Systems*, 34 (1992), 537-544
- [7] Hung D., "Using FPGA Technique for Design and Implementation of a fuzzy Inference System", *Fuzzy Logic and Intelligent Systems*, Kluwer Academic Publishers 1995, 271-288
- [8] Giuseppe Ascia, Vincenzo Catania, Marco Russo, Lorenzo Vita, "Rule-Driven VLSI Fuzzy Processor", *IEEE Micro*, 16 (1996), n. 3, 62-74
- [9] Poplawski, M., Bialko, M., "Implementation of fuzzy logic controller in FPGA circuit for guiding electric wheelchair", *Proceeding ICAISC'12 Proceedings of the 11th international conference on Artificial Intelligence and Soft Computing - Volume Part II*, Springer-Verlag Berlin, Heidelberg 2012, 216-222