Expert Systems with Applications 38 (2011) 9752–9760

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Application of Neuro-Fuzzy Controller for Sumo Robot control

Hamit Erdem [*]

Başkent üniversitesi, Mühendislik fakültesi – EEM bölümü, Bağlıca kampusu. Eskişehir yolu 20 km, Ankara p.kodu (p.box): 06530, Turkey

ARTICLE INFO

Keywords:
Neuro-Fuzzy
Sumo Robot
Robotic competition
NEFCLASS

ABSTRACT

This paper proposes the application of Neuro-Fuzzy (NF) hybrid system for Sumo Robot (SR) control. This robot is frequently designed by engineering students for robotic competition. As the relation between sensors output signals and motors control pulses is highly nonlinear in SR, soft computing techniques can be used to define this nonlinear relation and control of the robot in a competition ring. Application of intelligent methods for SR control not only simplifies robot control and improves robot responses during competition, but also encourages engineering students to use intelligent methods for solving real world's problems. Regarding above rationale, a NF controller for SR control is proposed and implemented. Firstly, a Fuzzy Inference System (FIS) for detecting and tracking of the opponent in the competition ring is developed, which relates sensor output signals to motor control pulses. Secondly, Artificial Neural Networks (ANN) based learning algorithm is used for rule extraction and tuning the FIS parameters. The design approach of the proposed controller is presented in detail, and effectiveness of the controller is demonstrated by hardware implementation and experimental results. The results show that the intelligent control methods can be easily applied in various robot competitions by engineering students.

1. Introduction

Mobile robots as a multidisciplinary field in engineering education provides an attractive platform for combining mechanical, electronic, computer, control, and communication systems to create an integrated system for education and research (Nourbakhsh et al., 2005). This field has evolved quickly over the last decade because of the enormous increase in computing power and the availability of the improved variety of sensors and actuators. From point of the education, robotic enables graduate and undergraduate students to develop a working low cost system with software and hardware integration (Fernandez, Marin, & Wirz, 2007). Parallel to developments in mobile robot technologies, design and implementation of various mobile robots for robotic competition like SR, line tracking and fire fighting robots have been increased. Thus, many engineering faculties encourage their students to take part in designing, constructing, analyzing, and proposing solutions to mobile robot problems with different ways (Maxwell & Meeden, 2000; Murphy, 2001).

Mobile robot design problems can be evaluated in two classes; design of mechanical structure and development of an efficient control algorithm. During the last decade, control of robots has attracted the attention of researchers. Addition to classical control methods, soft computing (SC) based algorithms and Intelligent Control methods have been used for mobile robot control (Pratihar, 2003). These methods can be applied to control of a competitor SR, too. For example, in SR control, besides design of mechanical structure, accuracy of sensors and motors power, correct mapping from sensor output signals to actuator control command space has an important effect in the determination of winner robot.

There are many advanced and midsize robotic competitions for engineering students, and these competitions have become a common occurrence in many engineering faculties (Maxwell & Meeden, 2000; Murphy, 2001). These activities encourage students to learn and utilize all the necessary hardware and software based tools and techniques required to achieve a good performance in competition. Design problems associated with the robotic competitions are often challenging problems that need various solutions in terms of hardware and software choices. The recent developments in robotic technology and mobile robotic competition force engineering students to offer new approaches and strategies in control of competitor robots.

One of the most popular mobile robot competitions is SR competition (Liu & Zhang, 2004). During SR competition, similar to traditional Sumo wrestling, an autonomous mobile robot tries to find the opponent inside the wrestling board by using detection sensors and pushes the opponent out of the ring. Without considering the mechanical structure, two main factors affect the result of competition. The first factor is the application of maximum power to the motors during attack mode depending on motors power, power supply, and motor driver system. The second factor is related to effectiveness of the opponent detecting and the opponent facing

[*] Tel.: +90 312 234 1010/1350; fax: +90 312 2341508.
E-mail address: herdem@baskent.edu.tr
URL: http://www.baskent.edu.tr/~herdem/

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.02.024

algorithm, which is an algorithmic problem and can be improved by utilizing SC methods.

As a SC tool, FIS is well suited for mobile robot control because it is robust even under uncertainty of a model and environmental conditions. This method eliminates the difficulties of modeling problems and allows building robust and smooth controller based on heuristic knowledge and qualitative models. Moreover, FIS determines a nonlinear mapping from its input space to output space by several If–Then rules based on the human knowledge and reasoning mechanism. Application of FIS for mobile robot control has been presented by Montaner and Ramirez-Serrano (1998), Driankov and Saffiotti (2001).

Ever since the FIS was applied in many industrial and robotic applications, construction of a well performing fuzzy system is not always easy. The main disadvantage of FIS is the requirement of an expert knowledge to set up fuzzy parameters in design step. The problem of finding appropriate membership functions (MF) and fuzzy rules are often a tiring process of trial and error. Reducing design step problems and improving FIS performance can be achieved by utilizing data analysis methods and SC techniques. Application of these methods in robotic has been investigated by Tang, Quek, and Ng (2005). As a SC tool, ANN has been used in combination with fuzzy logic to support the process of developing a FIS for a given task (Nauck, 1997). Thus, integration of FIS and ANN, Neuro-Fuzzy with an ANN like structure integrates the learning capability of ANN with interpret-ability of FIS. This integration provides knowledge extraction from system data as FIS rules. There are several approaches to integrate ANN and FIS depending on the application type (Nauck, Klawonn, & Kruse, 1997). Comparison and evaluation of NF systems have been presented by Mitra and Hayashi (2000). Also, Rusu, Petriu, Whalen, Cornell, and Spoelder (2003) proposed a NF controller for indoor application in mobile robot navigation.

SR is a low cost mobile robot, which contains the imperfect low cost IR or ultrasonic sensors for opponent detection. In order to improve sensor output signal quality, sensor conditioning circuit or algorithm namely sensor linearization process can be used. Use of an additional circuitry or an algorithm enlarges hardware structure or increases control code's size, which are unwanted issues in SR design. Furthermore, the opponent detecting and tracking needs localization of the opponent, which can be calculated by using kinematics equations. This calculation requires perfect sensors and a powerful processor. In contrast, FIS suggests a direct mapping from the sensor output signal to motor control voltage space under sensor uncertainty and noisy output.

Neuro-Fuzzy Classification (NEFCLASS) algorithm was introduced by Nauck and Kruse (1995, 1999). This algorithm is one of the NF systems used for learning of fuzzy parameters. They analyzed design of a FIS from available data and overcame the disadvantages of fuzzy without losing the advantages. Nauck and Kruse (2000) applied NEFCLASS-J (java based NEFCLASS algorithm) to Wisconsin Breast Cancer Study data. Their study showed that NEFCLASS-J can be a valuable tool for finding a small set of FIS rules. Also, NEFCLASS was applied by Akcayol and Sagiroglu (2007) in control of a buck converter, and Akay, Akcayol, and Kurt (2008) used this method for rule optimization in a manufacturing problem.

This study aims to utilize of NEFCLASS to develop a SR controller. In the prior study, the same author applied FIS to control of SR with 21 rules Erdem (2007). In the proposed study, author aims to use NEFCLASS for tuning of fuzzy parameters for reducing rules and tuning of fuzzy sets. Application of this method for SR control not only simplifies robot control and improves robot responses in the competition ring, but also will encourage engineering students to use intelligent methods for solving real world's problems by applying this method to control of a simple and known area.

The rest of paper is organized as follows. Section 2 presents the problem definition and discussion of SR operation. Section 3 defines the developed control algorithm and NEFCLASS. Section 4 describes the application of FIS to SR and tracking of the opponent problem. The performance of the designed robot, applied control algorithm and the experimental results are discussed in last section.

2. Problem definition and discussion of SR operation

The architecture of a SR is based on nonholonomic autonomous differential drive robot. This robot collects environment data and sends motion commands to right and left DC motors. The steering of the robot is accomplished by rotating the motors at different angular velocities. When motors moving at the same speed, the robot moves straight, by moving motors in opposite directions the robot turns on the spot. When one motor rotates slower than the other, the robot will move in an arc curving to the side of the slower turning. Fig. 1 shows the behavior of a differential mobile robot and equation (1) defines the relation between motors and robot velocities. In the related equations, VL and VR indicate linear speed of left and right wheels, and V demonstrates the robot linear speed. R (distance from ICC to the midpoint between the two wheels) indicates an instantaneous curvature radius of the robot trajectory, and ω_i show angular velocity of the wheels, respectively. The position and orientation of the robot can be calculated regarded to robot speed in X–Y dimension. Eqs. (2) and (3) define the robot pose based on kinematics relation. Calculation of the robot position and orientation with kinematics equation and path planning needs more accurate sensors, robot model and a powerful processor. Control of the robot with FIS eliminates to need to accurate modeling, and simplifies path planning.

Motors and robot velocity:

$$\begin{aligned}&\omega_L\left(R+\frac{L}{2}\right)=V_L;\quad \omega_R\left(R-\frac{L}{2}\right)=V_R;\quad \omega=\frac{(V_R-V_L)}{L}\\&V_{Robot}=V=\omega.R=\frac{(V_R+V_L)}{2}\end{aligned}\tag{1}$$

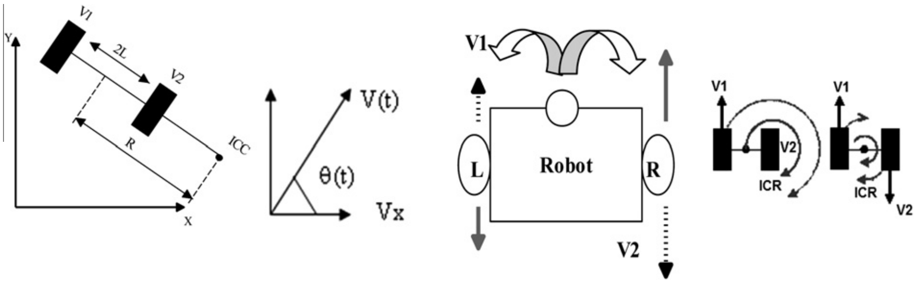

Fig. 1. Differential drive based mobile robot structure.

Position of robot:

$$\begin{aligned} V_x &= V(t) \cdot \cos[\theta(t)] \\ V_y &= V(t) \cdot \sin[\theta(t)] \end{aligned} \tag{2}$$

$$\begin{aligned} x(t) &= \int_o^t V(t) \cdot \cos[\theta(t)]dt + x_0 \\ y(t) &= \int_o^t V(t) \cdot \sin[\theta(t)]dt + y_0 \\ \theta(t) &= \int_o^t \omega(t) \cdot dt + \theta_0 \end{aligned} \tag{3}$$

In SR competition, each robot tries to find the opponent as soon as possible and pushes it out of the ring. As given in Table 1, SR behaviors during competition can be summarized in four operation modes. In search or hunting mode, SR rotates and scans the area to find the opponent. Following to opponent detection by one of the detection sensors, robot tries to localize the opponent and turn toward it by controlling motor velocities (facing mode). Following facing mode, SR applies maximum power to motors and attacks to the opponent in order to push it to out of the ring.

Regardless of mechanical shape and structure, two main factors affect the SR competition result. These factors are efficiency of routing and facing algorithm, which directs SR toward the opponent (facing mode), and applying maximum power to motors during attack mode. The second factor depends on the motors power, power supply and motor driver system, but the first one is an algorithmic problem and can be improved through a control algorithm. As given in Fig. 2, with consideration of SR operation modes, a FIS based control algorithm is expanded for control of the robot during facing mode. In this mode controller accepts continuously detection sensors signals and tries to face toward the opponent as soon as possible. Proposed SR control algorithm and robot hardware is given in Fig. 2.

Table 1
SR Operation modes.

Mode	Definition
Search/ hunting	Robot moves around and scan the ring for opponent detection
Opponent facing	The opponent has been sensed before and robot tries to face the opponent
Attack	Robot drives straight forward at full power to push the opponent out of the ring
Survive	Line sensor has detected the ring border line and robot tries not going off the competition ring

Regards to SR operation modes, a SR is designed and implemented. During competition, FIS receives the outputs of the detection sensors, integrates these signals and generates motor control pulses to direct SR toward the opponent. To detect the opponent, three infrared range sensors (S1, S2, S3) are used. These sensors are mounted on the nearly left, nearly right and middle front of the robot. The main parts of the SR are as follows.

- Ring border detection (IR sensor) and touch sensor (mechanical switches).
- Opponent detection sensors (S1, S2, S3: Sharp IR sensors).
- Right and left DC motors and drivers.
- Main controller (ST52 × 420 fuzzy microcontroller).

2.1. Opponent detection sensor

For localizing and detecting of the opponent, three GP2D12 IR Sharp distance measure sensors are used. This sensor is a compact, self-contained IR ranging system including an IR transmitter, receiver, optics, filter, detection, and amplification circuitry (Sharp Corporation, 1999). GP2D12 generates analog voltage responding to measured distance. The effective sensing range of sensor is between 8 cm and 55 cm, which correspond to 0.4–2.7 V range respectively. The compact sensor uses triangulation method for distance calculation. As shown in Fig. 3 the sensor has a highly nonlinear voltage–distance characteristic. Regarding sensor transfer function, using other control methods, needs application of a sensor linearization algorithm before applying control action. Using a FIS based control eliminates to utilize of linearization approach.

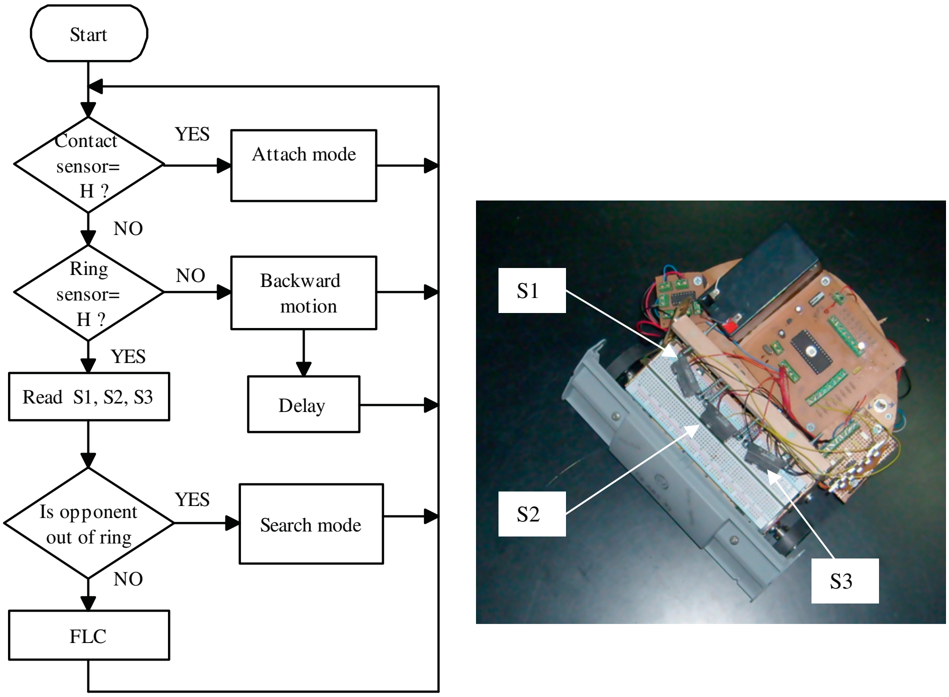

Fig. 2. SR hardware and robot control algorithm.

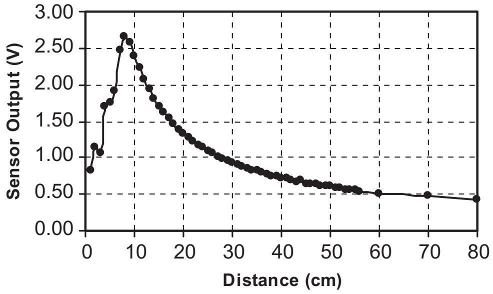

Fig. 3. Relation of measured distance and output voltage for IR sensor.

3. Neuro-Fuzzy systems and NEFCLASS model

FIS and ANN are two fundamental systems in SC and each system has advantages and disadvantages. To overcome limitation of two systems, researchers integrate learning capability of ANN with reasoning capability of FIS as NF hybrid intelligent system (Fakhreddine & Karray, 2004). This integration aims at the generation of a more robust, efficient and easily interpretable system, where the advantages of each model are kept, and their possible disadvantages are removed. From the point of the architecture, the integrated system may be interpreted either as an ANN with fuzzy parameters, or as a FIS implemented in a parallel distributed form. From point of the FIS implementation view, the aim of these systems is providing the fuzzy parameters (rules and sets) from available data. It can be said, NF system is a FIS trained by a (heuristically) learning algorithm derived from neural networks (Nauck & Kruse, 2000). Thus, the learning process in NF is a data-driven approach, and NF simplifies application of FIS to a system which is represented by numerical knowledge. The advantages and disadvantages of the systems are given in Table 2.

NF systems can be divided into three major types based on their topologies and functionalities: cooperative NF systems, the ANN driven FIS, and hybrid neural network-based systems (Fakhreddine & Karray, 2004). NEFCLASS can be evaluated as a hybrid ANN based system. NEFCLASS has an ANN like structure, which uses fuzzy parameters, or it can be interpreted a FIS implemented in a parallel structure. Different NF models are described by Nauck and Kruse (1995), Nauck (1997).

3.1. Architecture of NEFCLASS

NEFCLASS supports FIS designer in construction of a FIS from the system data and provides the interpretability of the developed classifier. NEFCLASS suggests learning algorithms to create the structure (rule base) and tuning the parameters (fuzzy sets) of a fuzzy classifier from system's data. It can be said, NEFCLASS tunes the FIS parameters by adding and deleting the rules (Nauck & Kruse, 1995, Nauck & Kruse, 1997). Although NEFCLASS supports design of a FIS, it is not an automatic FIS creator from system's data. Thus, designer can interpret the learning procedure. The architecture of NEFCLASS is like to a 3 layer feedforward ANN which interprets the FIS structure (Nauck & Kruse, 1997). Although ANN structure has a black box structure and does not allow interpretability, NEFCLASS structure is interpretable by FIS.

Application of NEFCLASS needs a classified dataset. Each feature of data is an input node or fuzzy input variable, and each class of data also defines an output node or a member of output fuzzy set. The fuzzy rules respect to classification have the form "$S1$ is $\mu 1$ and $S2$ is $\mu 2$ and $S3$ is $\mu 3$ then pattern S belongs to class Ci", where $S(s1, s2, s3)$ is the feather vector, and $\mu 1, \mu 2, \ldots \mu m$ are fuzzy sets and $C1, C2, \ldots C3$ are classes. NEFCLASS classifies input patterns into output classes by using linguistic rules (Nauck & Kruse, 1995). Table 3 summarizes the main features of NEFCLASS. In the proposed system, the inputs of the NEFCLASS are the outputs of IR sensors and the output of the model is five classes for control of robot direction and position.

The architecture of NEFCLASS for proposed problem is shown in Fig. 4. As shown in figure, the first layer U1 contains the input pattern features (sensor outputs), the hidden layer U2 holds the rules, and output layer U3 shows fuzzy outputs (five classes). Regarding these layers, Eqs. (4) and (5) define the input pattern q, and the activation functions of nodes in hidden and output node. In given equations, $W(x, R)$ is the weight on the connection from input to hidden layer, and $W(R, c)$ indicates a weight from the hidden to the output layer in NEFCLASS structure. These weights are fixed at one or zero (Nauck & Kruse, 1995).

$$a_R^{(q)} = \min x \in U_1\{W(X,R)(a_x^{(q)})\} \tag{4}$$

$$a_c^{(q)} = \sum_{R\in U_2} \frac{W(R,C)\cdot a_R^{(q)}}{W(R,C)} \text{ or } a_c^{(q)} = \max R \in U_2\{a_R^{(q)}\} \tag{5}$$

3.1.1. Learning approaches in NEFCLASS

Learning procedure in NEFCLASS needs training dataset containing input output data classes, and learning contains two steps: obtaining fuzzy classification rules from labeled dataset and tuning of the predefined fuzzy sets. The first step is rule generation from available dataset or modification of the initially suggested rule base by adding or deleting the rules. Rule learning which is a structure oriented approach is achieved by an approaches, which suggested by Wang and Mendel (1992). During rule generation if there is no initial rule base, NEFCLASS starts with no hidden layer and hidden layer neurons are created during the learning task.

Table 2
Comparison of FIS and ANN systems.

	Fuzzy	Neural networks
Advantages	– No mathematical model – Prior knowledge can be used – Simple implementation – Human like reasoning – Handle uncertainty, ambiguity – White box (Interpretability) – Verification is possible – Linguistic representation of knowledge	– No mathematical process model required – No rule-based knowledge required – Different learning algorithms available – Adaptive – Massive parallelism – Learning capability – Fault tolerance – Knowledge distributed within network
Disadvantages	– Knowledge of system must be available – No learning – Design time is more (trial and error) – Tuning is difficult – Needs expert knowledge	– Rules cannot be extracted – Determine heuristic parameters – Adaptation to modified environment is difficult – No guarantee learning converges – Black Box (difficult to interpret) – Prior knowledge cannot be used

Table 3
The features of NEFCLASS.

Layers	Input layer: Inputs of fuzzy system	Hidden layer: rules	Output layer: classes (1 neuron per class)
Learning: 2 step	Step1: Rule base creation (structure learning)	Step2: Fuzzy set tuning (error based learning)	Pruning (Rule, variable reduction)
Number of neurons	Numbers of fuzzy variables (input layer)	Numbers of rules (hidden layer)	Number of classes (output layer)
Weights	Input to hidden (fuzzy sets)	Hidden to output (linear = 1/0)	Initial fuzzy sets defined by user
Activation function	Input variables directly connected	Fuzzy min.operation (t-norm)	Max operation (t-conorms)

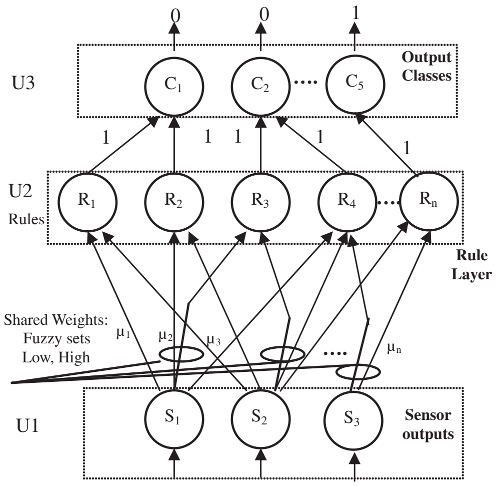

Fig. 4. The NEFCLASS structure with three inputs and five output classes.

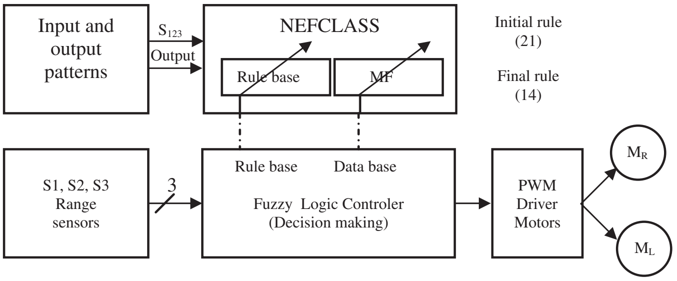

Fig. 5. Block diagram of NEFCLASS based SR control.

Regards to learning procedure, new rules will be added or some of the initially defended rules will be deleted. The initial number of rule can be defined, and NEFCLASS tries to determine the best rules per class. The maximum number of the rules to be created may be specified by the user or this task can be achieved by NEFCLASS through learning algorithm. Once the number of hidden units (rules) has been determined, the fuzzy sets between the input and the hidden layer are tuned to improve the classification accuracy of the network (Nauck & Kruse, 2000).

The second learning step is the fuzzy parameter modification by learning algorithm. This modification results in shifting MFs, enlarging or reducing its support. Modification of a MF is determined by classification error of each rule. This approach uses a learning algorithm like back-propagation, and the output error which is propagated from outputs to inputs to change the MF parameters (Nauck & Kruse, 1995). During this step, the number of misclassified or not classified samples will be reduced.

4. Applying Neuro-Fuzzy Controller to Sumo Robot

Based on SR competition rules, a control algorithm is developed. This algorithm takes advantage of the essential characteristics of a differential drive. The controller continuously reads detection sensor output signals and sends control pulses to the motor driver circuit. During the facing mode, a simple multi input-single output FIS controls the robot. As shown in main algorithm (Fig. 2), the role of FIS begins when one of the IR sensors receives the opponent detection signal. The input values are evaluated regarding fuzzy rules, and the result is applied to motors. The proposed FIS parameters are optimized with NEFCLASS, and the structure of the proposed system is shown in Fig. 5. This structure has three inputs and five output nodes. Each output node defines the opponent position regard to the SR position. The initial fuzzy rules have determined based on the opponent's possible locations in wrestling ring. Some of these aspects are shown in Fig. 6. Erdem (2007) defined the fuzzy rules based on this assumption and applied FIS to SR control. Dataset which was recorded during the last study is used for NEFCLASS based application.

The block diagram of proposed NEFCLASS based fuzzy controller for SR is shown in Fig. 5. As shown in figure, the FIS parameters were first tuned offline by NEFCLASS algorithm. Then, based on tuned parameters, the controller is implemented into a fuzzy microcontroller. For input variables triangle form MF and for output class singleton MF is selected. The training dataset contains the description of the input and output variables and some patterns are given in Table 4. As shown in table, the dataset contains three inputs and corresponding five output classes with 592 samples. The first three columns are sensor output signals and the other columns are output classes. In training set, initial values of both MFs and main rules have been obtained regarding SR motion behavior during competition.

The IR sensor generates analog voltage between 0.42 V and 2.66 V. This range is divided into three sets for each input sensor, and the selected fuzzy sets are as follows: Low (L), Medium (M) and High (H). For the output of FIS which controls the motion of motors, five MFs are considered. These MFs are Full Left (FL), Small Left (SL), Center (C), Small Right (SL) and Full Right (FR). Fig. 7 shows initial fuzzy sets for input–output variables. Based on FIS results, PWM signals are sent to the motor driver circuit. For example, generation of Center (*C*) value maintains the robot on a straight direction while sending other values to the controller, this makes the robot turn to the right or the left. Rule extraction training step is started with three main rules. These rules define three main motions of the robot (turn left, turn right and forward), and the other rules are achieved by NEFCLASS. The initial rules are shown in Table 5. Finding the maximum rules is provided by NEFCLASS. In this work, 16 rules are created by classifier procedure and with using pruning procedure the rules are reduced to fourteen. The optimized rules are shown in Table 6, and the modified fuzzy sets are given in Fig. 8. For example, Fig. 9 shows respond of FLC regard to instantaneous input values when the opponent is nearly in the center position.

In learning steps, classification scheme was obtained by running NEFCLASS with learning rate 0.2. The final advantage of NEFCLASS is nearly 30% rule reduction (reduced rules from 21 to 14) and best

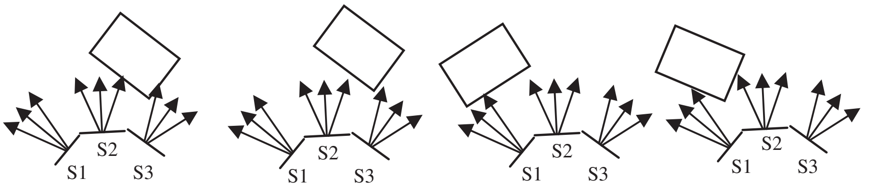

Fig. 6. Some of the possible opponent positions.

Table 4
Input–output patterns for NEFCLASS.

No of pattern	*S*1	*S*2	*S*3	Full Left	Small Left	Center	Small Right	Full Right
1	1.2	0.1	0.2	1	0	0	0	0
2	0.1	0.9	0.3	0	0	1	0	0
3	0.8	0.9	0.1	0	1	0	0	0
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
590	0.6	0.2	01.5	0	0	0	0	1
591	0.2	0.8	1.0	0	0	0	1	0
592	1.2	1.3	0.6	0	0	1	0	0

fitting fuzzy sets for input variables. Following offline fuzzy parameter learning, the controller was implemented into the fuzzy microcontroller. Microcontroller accepts sensor outputs voltage. After AD conversion, the result is fuzzified and evaluated in FIS algorithm. The result of rule evaluation and defuzzification provides real values for PWM module, which controls the speed of motors and the robot speed, respectively. The output of FIS is calculated as the average of the individual centroid, weighted by their heights as given in Eq. (6).

$$\Delta i = \frac{\sum_{i=1}^{n}\mu(\mu_i).\mu_i}{\sum_{i=1}^{n}\mu(\mu_i)} \qquad (6)$$

4.1. Hardware implementation of the controller

Fuzzy controllers have been used in many applications because of their rapid design by translating heuristic knowledge to the control action. However, they require special operators (such as fuzzification or defuzzification) which are not available in standard microcontrollers. ST52 × 420 as a fuzzy microcontroller supports a set of dedicated fuzzy logic instructions, and it can perform a fuzzy algorithm in an efficient manner. This controller is used as the main controller for the designed robot. In hardware application, the sensor outputs directly are connected to three analog inputs of the controller, and the ADC results are sent to the fuzzy block. In practical application, the five singleton MFs are selected based on 8 bit register values as follows; Full Left (FL = $40), Small Left (SL = $60), Center (C = $80), Small Right (SL = $A0) and Full Right (FR = $C0).

Table 5
Initial fuzzy rules.

Rule	S1	S2	S3	Motion
Rule1	High	Low	Low	Full Left
Rule2	Low	High	Low	Center
Rule3	Low	Low	High	Full Right

Table 6
The optimized fuzzy rules table.

Rule	*S*1	*S*2	*S*3	Motion
Rule1	Low	Low	Medium	Small Right
Rule2	Low	Low	High	Full Right
Rule3	Low	Medium	Low	Center
Rule4	Low	Medium	Medium	Small Right
Rule5	Low	Medium	High	Small Right
Rule6	Low	High	Low	Center
Rule7	Low	High	Medium	Small Right
Rule8	Low	High	High	Center
Rule9	Medium	Low	Low	Small Left
Rule10	Medium	Medium	Low	Small Left
Rule11	Medium	High	Low	Small Left
Rule12	High	Low	Low	Full Left
Rule13	High	Medium	Low	Small Left
Rule14	High	High	Low	Center

5. Experimental results

The developed NEFCLASS based FIS is implemented in designed mobile robot, and opponent seeking experiments was achieved. In

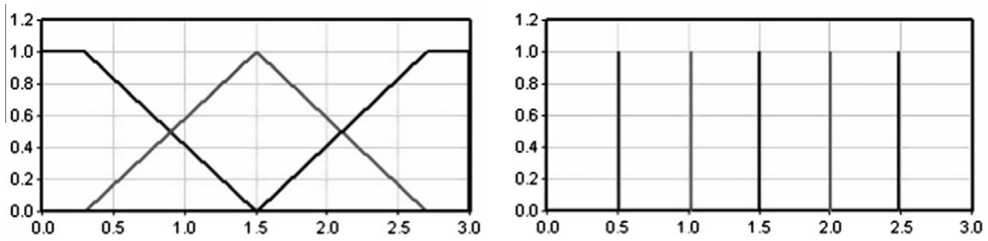

Fig. 7. Initial fuzzy sets for input and output variables.

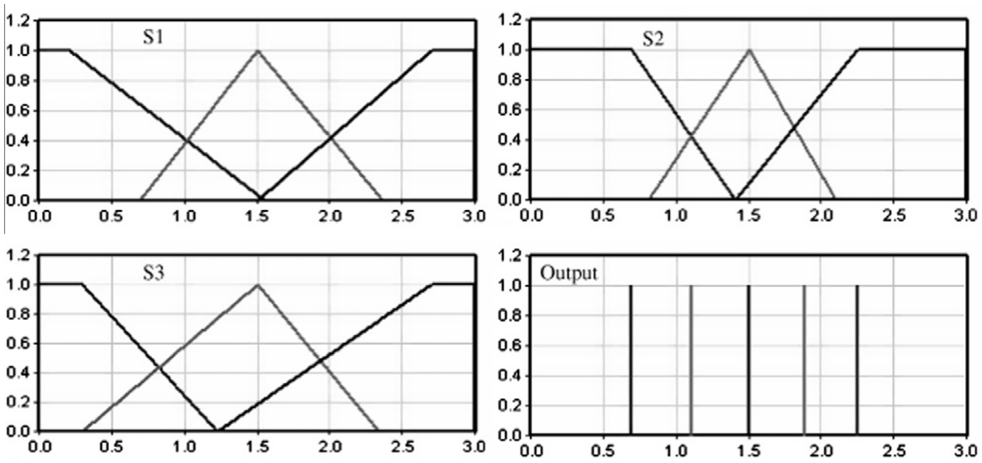

Fig. 8. Optimized MF's for input variables and classes labels.

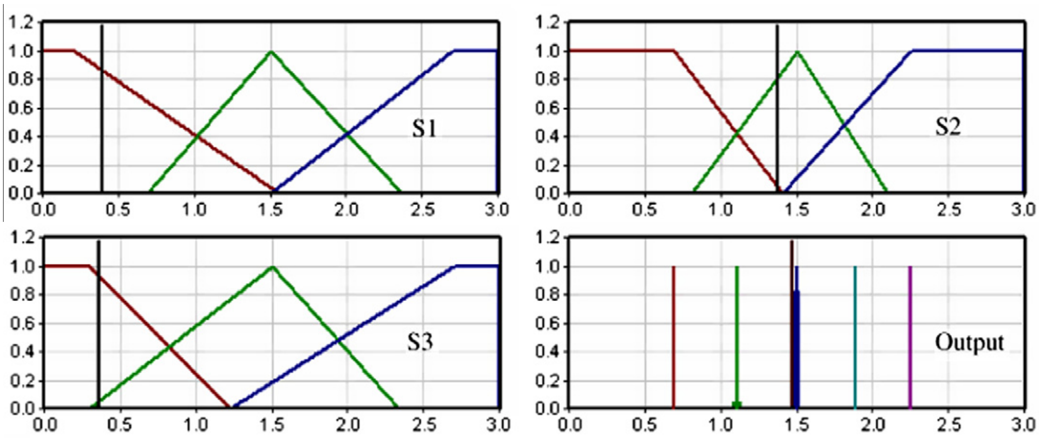

Fig. 9. Respond of controller when opponent is in center position.

the performed experiment, which is shown in Fig. 10, SR tries to find a motionless opponent, which is located in the left side of the robot. Robot begins motions in wander mode by rotating to the left. The sequences of opponent detection steps are as follows; during scanning mode, firstly S1 detects opponent and FL algorithm starts and the controller sends "turn to left" command to the motors. Following the first motion and turning, S2 detects the opponent. Based on the main control algorithm (Fig. 2), controller continuously reads sensors and sends control pulses until facing toward the opponent is achieved. Fig. 11 shows some of these sequential motions. During these motions, the sensors output voltage changes sequentially and the motors receive new commands. Figs. 12 and 13 show sensor output signals and the corresponding PWM pulses for motor control. As given in figures, during the first samples (opponent detecting) output of each sensor change rapidly because of the sequential facing maneuvers. During the next samples, S2's output (middle sensor) increases continuously, which implies direct motion of the robot toward the opponent (Attack mode). During same motions, the output of the other sensors is low and remains unchanged because these sensors do not sense the opponent. After 23 sampling time, the output of middle sensor (S2) remains in a constant value because the hunter has been touched the opponent and pushes it out of the ring. Based on the control algorithm, motor control pulses (8 bits PWM signal values) are changed continuously regarding sensor output signals. These control pulses (PWM signals) are unequal during search mode and during facing mode. During the last motions (attack mode) two control pulses are the same because of the direct motion toward the opponent. The sampling time of sensor output signal is 0.002 s, and PCI-DAS6023 data acquisition card is used for sampling analog values.

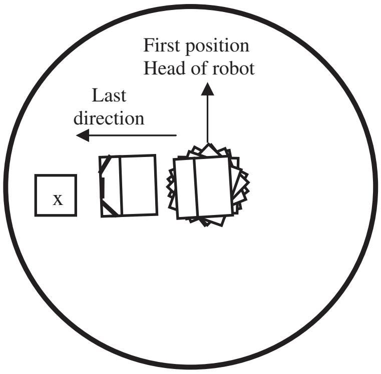

Fig. 10. Opponent detecting and tracking task.

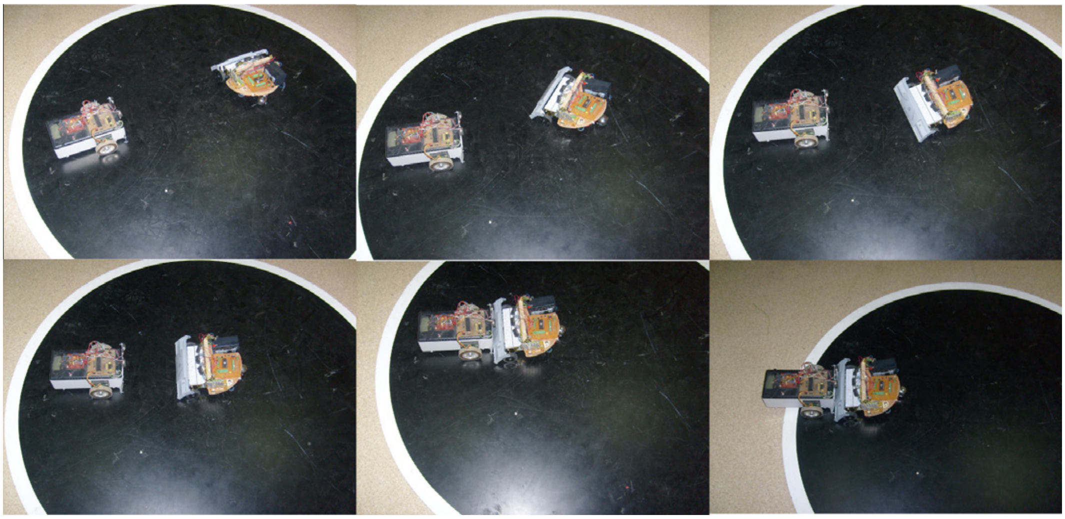

Fig. 11. SR sequential motions in the wrestling ring.

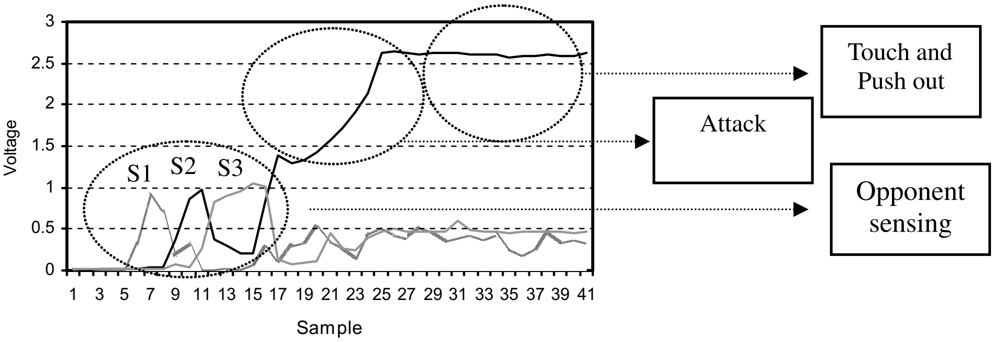

Fig. 12. Sensor responses during the opponent detection mode.

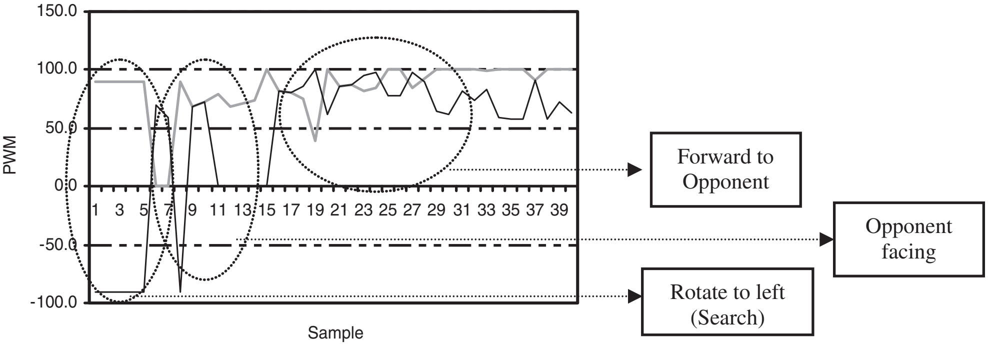

Fig. 13. Chang of 8 bit PWM signals during opponent detection task.

6. Conclusion

In this study, a new control algorithm was applied for SR control. The applied controller has used the combination advantages of fuzzy and ANN. NEFCLASS based controller was developed and applied by using a fuzzy microcontroller. Compared to previous study, application of learning algorithm has provided 30% rule reduction, which improves the robot response speed in the competition ring, and the robot can attack to the opponent with minimal maneuvers. The performance of applied control can be improved with adding extra sensors for sensing attack of the opponent during competition. The design and implementation results showed that intelligent control and soft computing techniques can be easily applied to various robotic competitions by engineering students.

References

Akay, D., Akcayol, M. A., & Kurt, M. (2008). NEFCLASS based extraction of fuzzy rules and classification of risks of low back disorders. *Expert Systems with Application, 35*(4), 2107–2112.

Akcayol, M. A., & Sagiroglu, S. (2007). Neuro-fuzzy controller implementation for an adaptive cathodic protection on Iraq–Turkey crude oil pipeline. *Applied Artificial Intelligence, 21*(3), 241–256.

Driankov, D., & Saffiotti, A. (2001). *Fuzzy logic for autonomous vehicle navigation*. New York: Springer.

Erdem, H. (2007). A Practical Fuzzy Logic Controller for Sumo Robot Competition. *Pattern Recognition and Machine Intelligence. LNCS* (Vol. 4815). Springer (pp. 217–225).

Fakhreddine, O., & Karray, C. de Silva (2004). *Soft computing and intelligent systems design, theory, tools and applications*. England: Pearson Education.

Fernandez, J., Marin, R., & Wirz, R. (2007). Online competitions: An open space to improve the learning process. *IEEE Transactions on Industrial Electronics, 54*(6), 3086–3093.

Liu, J. M., & Zhang, S. W. (2004). Multi-phase sumo maneuvers learning. *Robotica, 22*(Part 1), 61–75.

Maxwell, B. A., & Meeden, L. A. (2000). Integrating robotics research with undergraduate education. *IEEE Intelligent Systems, 15*(6), 22–27.

Mitra, S., & Hayashi, Y. (2000). Neuro-fuzzy rule generation: Survey in soft computing framework. *IEEE Transactions on Neural Networks, 11*(3), 748–768.

Montaner, MB., & Ramirez-Serrano, A. (1998). Fuzzy knowledge-based controller design for Autonomous Robot Navigation (robotics and computer vision). *Expert Systems with Applications, 14*(1–2), 179–186.

Murphy, R. (2001). Competing for a robotics education. *IEEE Robotics and Automation Magazine, 8*(2), 44–55.

Nauck, D. (1997). Neuro-fuzzy systems: Review and prospects. In *Fifth European congress on intelligent techniques and soft computing conference* (pp. 1044–1053).

Nauck, D., Kruse, R. (1995). NEFCLASS- A neuro- fuzzy approach for the classification of data. In *ACM symposium on applied computing* (pp. 461–465).

Nauck, D., Klawonn, F., & Kruse, R. (1997). *Foundations of neuro- fuzzy systems*. Wiley.

Nauck, D., & Kruse, R. (1999). Obtaining interpretable fuzzy classification rules from medical data. *Artificial Intelligence in Medicine, 16*, 149–169.

Nauck, D., & Kruse, R. (2000). NEFCLASS-J – A java-based soft computing tool. *Intelligent systems and soft computing: prospects, tools and applications. LNAI* (Vol. 1804). Berlin: Springer (pp. 143–164).

Nourbakhsh, I. R., Crowley, K., Bhave, A., Hamner, E., Hsiu, T., Perez-Bergquist, A., et al. (2005). The Robotic Autonomy mobile robotics course: Robot design, curriculum design and educational assessment. *Autonomous Robotics, 18*(1), 103–127.

Pratihar, D. K. (2003). Algorithmic and soft computing approaches to robot motion planning. *Machine Intelligence and Robotic Control, 5*, 1–16.

Rusu, P., Petriu, E. M., Whalen, T. E., Cornell, A., & Spoelder, H. J. W. (2003). Behavior-based neuro-fuzzy controller for mobile robot navigation. *IEEE Transactions on Instrumentation and Measurement, 52*(4), 1335–1340.

Sharp Corporation (1999), Distance measuring sensor (GP2D212).

Tang, A. M., Quek, C., & Ng, G. S. (2005). GA-TSKFNN: Parameters tuning of fuzzy neural network using genetic algorithms. *Expert Systems with Applications, 29*(4), 769–781.

Wang, L. X., & Mendel, J. M. (1992). Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man and Cybernetics, 22*(6), 1414–1427.