© Universiti Tun Hussein Onn Malaysia Publisher's Office

IJIE

http://penerbit.uthm.edu.my/ojs/index.php/ijie
ISSN : 2229-838X e-ISSN : 2600-7916

The International
Journal of
Integrated
Engineering

Mobile Robot Path Following Controller Based On the Sirms Dynamically Connected Fuzzy Inference Model

Abdeslam Benmakhlouf[1*], Abdellah Meraoumia[2], Mohammed Lakhder Louazene[1]

[1]Applied Sciences Faculty, Department of electrical engineering,
Kasdi Merbah university, Ouargla, ALGERIA

[2]Laboratory of Mathematics, Informatics and Systems (LAMIS),
Larbi Tebessi University, Tebessa, ALGERIA

*Corresponding Author

DOI : https://doi.org/10.30880/ijie.2023.15.01.021
Received 25 May 2022; Accepted 08 February 2023; Available online 28 February 2023

Abstract: In this paper, the single input rule modules (SIRMs) dynamically connected fuzzy inference model is employed to design a path following controller for a unicycle wheeled mobile robot (WMR). The SIRMs model is mainly used to reduce the number of fuzzy inference rules and consequently reduce the processing time of the conventional Fuzzy Logic Control (FLC) schemes. The adopted path following strategy uses two control unites working in parallel to drive the WMR to follow a path composed of a succession of discrete waypoints. The first unit is a heading controller charged of generating the angular velocity command and the second one is a linear velocity controller charged of the generation the speed control signal. For the heading controller, all input variables are assigned with two fuzzy inference modules: a SIRM and a dynamic importance degree (DID). The structure of the SIRMs based linear velocity control unit was modified to simplify the design and to fulfill the requirements of the path following strategy. It contains two SIRM modules and one common DID. To ensure the stability of the WMR's motion, the SIRMs and the DIDs in the two control units were designed in such a manner that the control of the robot's orientation gets higher priority over the control of the WMR speed. The use of the SIRM fuzzy inference model allows the online automatic adjustment of priority orders of the different control actions. No optimization of the controller parameters is required since that all base values are set to 0 and all the breadths are set to 1 for the DID modules. This means that all the input items play equal roles in the control of the WMR. The structure of the proposed path following controller is simple and the inference rules are intuitively understandable because they are inspired from the knowledge of a human expert. Numerical simulations conducted in the MATLAB environment show that the proposed SIRMs based controller outperforms a path following controller that uses the conventional FLC scheme.

Keywords: Fuzzy logic control, mobile robot, path following, Single Input Rule Modules (SIRM)

1. Introduction

Wheeled mobile robots (WMR) having a relatively simple and robust mechanical structure, high load capacity, and flexible motion characteristic are considered as an important category of mobile robots. WMRs are widely investigated and have gained popularity in many fields such as industrial and service robotics [1].

Actually, various types of WMRs exist, they differ by their mechanical structure, their locomotion principle which depends on the type and number of the wheels and actuators, etc.; however, the most widely studied WMR type is the unicycle mobile robot [2]. The differential drive WMR equipped with two independent coaxial wheels driven by two distinct actuators, has several advantages such as the simple structure, the ease of implementation, and the ability to

---

*Corresponding author: ab.benmakhlouf@gmail.com

change its heading by just changing the angular velocities of the actuated wheels, without the necessity for any additional steering hardware [3].

Motion control focuses on how to make a WMR autonomous and self-ruling without external action. Motion control problems can be subdivided into three types of control problems: stabilization, path following, and trajectory tracking. The goal of a WMR stabilization controller is to stabilize it at a given target pose in the configuration space. In the path following control problem, the WMR is intended to converge to a geometrically specified reference path with the higher possible precision, independently of any temporal constraints starting from a given initial state. However, a trajectory tracking control algorithm aims to make a mobile robot track a desired trajectory whose parameters vary with time [4].

A wide range of diverse control algorithms dealing with the path following problem were proposed in the literature. Some of the proposed methods are based on the classical control theories [5] - [11], while others are based on modern and intelligent control techniques [12] - [23]. Among intelligent control techniques, fuzzy control based on linguistic information has been effectively presented in many research works [19] - [23]. The fuzzy logic control permits converting expert knowledge expressed as a linguistic inference system into an automatic control approach. The aim of the fuzzy path following controller presented in [23] is to reduce the number of inference rules by adopting a simple structure; it uses two FLC modules with 36 fuzzy inference rules in the total.

It is notable that in conventional fuzzy inference systems, the fact of increasing the number of input variables in the antecedent parts of each inference rule will exponentially increase the whole number of rules in the rule base. Consequently, constructing each rule clearly becomes difficult. To solve these problems and make the design process of the conventional fuzzy inference systems simpler, the SIRMs dynamically connected fuzzy inference model was proposed for plural input fuzzy control, in which, a SIRM is built and a dynamic importance degree is set up for each input variable [24]. This approach has been applied successfully for modeling [25], prediction [26] and control applications [27] - [33]. Reference [26] extends the SIRMs fuzzy model to the SIRMs connected neural-fuzzy system. To further enhance the performance of SIRMs, a type 2 fuzzy logic based SIRMs was introduced in [34]. The authors of [35] present a novel scheme to measure the importance degrees of the SIRMs using multivariable functional weights. The properties of continuity, monotonicity, robustness and stability of type 1 and type 2 fuzzy SIRMs are studied in [36], [37], and [38]. The most important and the most time-consuming step in the design process of a SIRMs based fuzzy inference system is the tuning of its parameters; which is achieved through the random optimization search method [32] or genetic algorithms [34].

This article proposes the use of the SIRMs dynamically connected type 1 fuzzy inference model to implement a path following strategy that was proposed in a previous paper [23]. The structure of the controller is divided into two control units; a heading controller implemented as a usual SIRM control system and a velocity controller implemented using a modified SIRM controller. The total number of inference rules used in this controller is 21 fuzzy rules with one variable in the antecedent part of each rule.

The contributions of the paper are:

a. The application of the SIRMs fuzzy inference model to the path following problem of a WMR.
b. The modification of the structure of the SIRMs fuzzy model in the velocity controller.
c. The simplification of the controller parameters optimization step by setting all the base values to *0* and all the breadths to *1* for all used DIDs modules.

The remainder of this paper is organized as follows: the kinematic model of a differential drive mobile robot is presented in section 2. Section 3 gives a review of the fundamentals of Fuzzy Logic Control (FLC) and presents the details of the SIRMs dynamically connected fuzzy model. The path following method is explained in section 4 and the controller structure and design are detailed in section 5. Simulation results are shown and discussed in section 6. Finally, the paper ends with a brief conclusion.

2. Kinematic Model of a Unicycle WMR

In the present paper, a typical configuration of a WMR with unicycle-kinematics is used (Fig. 1). In this construction, the robot possesses two coaxial driving wheels mounted on the chassis, each wheel with radius r. In this design, each driving wheel is equipped with a DC motor for actuation and an incremental encoder for revolutions counting. The robot uses the measurements delivered by the encoders to calculate the spinning velocity of each wheel. The mobile robot navigates on a flat horizontal terrain. The following vector specifies the pose of the WMR:

$$q_I = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \tag{1}$$

x and ***y*** in (*m*) represent the coordinates of the point ***P*** in the global reference frame (X_I, Y_I). ***θ*** (in *rad*) represents the angle made by the moving reference frame (X_R, Y_R) attached to the robot's chassis, and the fixed reference frame (X_I, Y_I), it is the heading (orientation) of the mobile robot.
The linear velocities of the driving wheels are calculated using equation (2), and the mobile robot's angular velocity and linear velocity are calculated using equation (3),

$$v_R = r\omega_R,\, v_L = r\omega_L \tag{2}$$

$$\omega = \frac{v_R - v_L}{l},\, v = \frac{v_R + v_L}{2} \tag{3}$$

where $\boldsymbol{\omega}$ is the angular velocity of the robot centroid in (*rad/s*). $\boldsymbol{v}$ is the velocity of the robot centroid in (*m/s*), $\boldsymbol{v_L}$ and $\boldsymbol{v_R}$ are the linear velocities of the left and right wheel respectively. $\boldsymbol{\omega_R}$ and $\boldsymbol{\omega_L}$ are the rotational speeds of the right wheel and the left wheel, respectively. ***r*** is the left and right wheels' radius in (*m*). Combining (2) with (3), the expressions of the linear and angular velocities can be written as follows:

$$\omega = \frac{r}{l}\left(\omega_R - \omega_L\right),\, v = \frac{v_R + v_L}{2} \tag{4}$$

Moreover, the kinematic model of the mobile robot can be defined as:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{5}$$

From Eq. 5, it's clear that the mobile robot has two control inputs and generates the derivative of its pose vector as output.

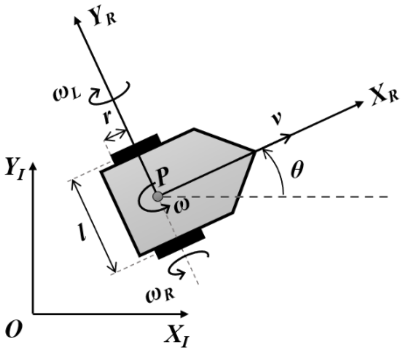


Fig. 1 - Differential drive mobile robot

3. SIRMs Dynamically Connected Fuzzy Inference Model

This section gives a brief review of the conventional Fuzzy Logic Control (FLC) and presents the SIRMs dynamically connected fuzzy inference model for Multi Input Single output (MISO) control systems of *n* input items and *1* output item.

FLC is a popular intelligent heuristic-based control technique. The FLC uses linguistic or qualitative information to embed the key elements of approximate reasoning and human expertise in the design of nonlinear control algorithms [13]. FLC does not require accurate mathematical models, can handle imprecise or uncertain inputs, nonlinearities, and present disturbance insensitivity greater than most classical nonlinear control algorithms. FLC outperforms other controllers in complex, nonlinear, or unmodeled systems for which adequate expert knowledge exists.

Fig. 2 shows the structure of a FLC. The FLC performs three basic steps to process the real crisp inputs: fuzzification, decision-making, or rule inference, and defuzzification as a means to generate a crisp value at the output [20].

In the fuzzification step, the crisp real values of the input variables are converted into linguistic variables with the help of membership functions (MFs), that is, the mapping of the input variables into suitable linguistic values. The fuzzified inputs are then manipulated by the inference engine to generate the fuzzified output according to the predefined rule base (knowledge base). The rule base represents an expert’s knowledge on how to control the system expressed in the form of *IF-THEN* inference rules. The inference engine (the interpreter) uses fuzzy operators on fuzzified inputs and fired rules to make a decision. The final step is defuzzification, in this stage, one of the defuzzification methods is used to transform the fuzzified output into the actual crisp output used for controlling the system.

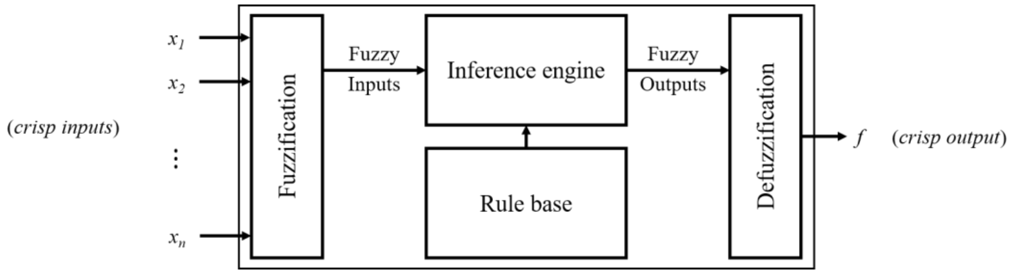


Fig. 2 - Structure of a Fuzzy Logic Controller (FLC)

In the conventional fuzzy inference model, the antecedent part of each fuzzy rule contains all the input items, which leads to the exponential increase of the number of fuzzy rules with the number of input items and causes difficulty in setting up each rule [32]. To overcome these problems, the SIRMs dynamically connected fuzzy inference model was proposed [24].

Fig. 3 shows the structure of a SIRMs based controller with *n* input variable and one output item *f*. For each input item x_i, two fuzzy modules are defined: a SIRM and a DID which use the *min-max-gravity* method or *product-sum-gravity* method, or *simplified inference* method to produce their outputs.

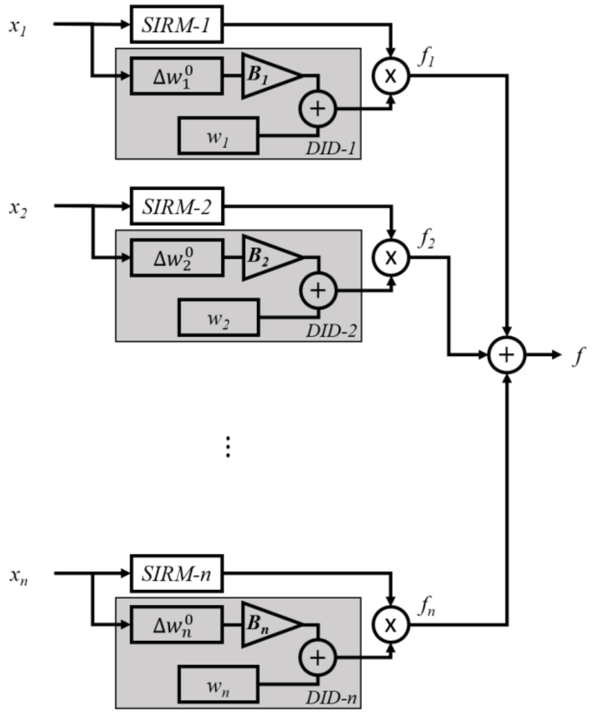


Fig. 3 - The structure of the SIRMs dynamically connected fuzzy inference model

For each input variable x_i, the corresponding SIRM defines the individual role of the i^{th} input in the control action. The SIRM uses a number of m_i membership functions (A_i) for the input fuzzification and C_i singletons (constant values) for the output calculation. The rule base (R_i) for this module contains m_i inference rule.

The output of the SIRM module (f_i) is computed using the following equation:

$$SIRM-i:\left\{R_i^j : if\ x_i = A_i^j\ then\ f_i = C_i^j\right\}_{j=\overline{1,m_i},\, i=\overline{1,n}} \tag{6}$$

Here, *i* and *j* are the indices of the input item and inference rule respectively.

It was noticed that this scheme does not perform well due to the inequalities of the roles played by each input variable in the generation of the control action. Some of the input items have significant contributions to the performance of the controller, so their contributions should be reinforced. While other input items have weak contributions, so their roles don't have to be reinforced [24]. Here comes the DID module, which are used to tune and indicate obviously the individual role of each input item on system performance according to the state of the system.

As shown in Fig. 3, each DID contains a fuzzy module that generates the dynamic importance degree of the input item ($\Delta \boldsymbol{w}_i^0$), multiplied by the breadth (B_i) to increase or decrease its effect, and a base value (w_i) which represents the minimum contribution of the input variable [32].

The output of an individual DID module is defined as:

$$w_i^D = w_i + B_i \Delta w_i^D \tag{7}$$

The output of the *SIRM-i* is multiplied by the output of the *DID-i* to produce the intermediate output $\boldsymbol{f}_i^0$. Summing up all the intermediate outputs results in the system output (f), as defined by:

$$f = \sum_{i=1}^{n} w_i^D f_i^0 \tag{8}$$

4. Path Following Strategy

The reference path is composed of a series of *n* discrete waypoints N_i ($i=1, 2, \ldots, n$), through which the WMR should pass to get to the final waypoint N_n. The coordinates of the desired waypoints are given as $\left[x_d(i), y_d(i)\right]^T$. The WMR is located at the first point of the path with an arbitrary orientation. The path following controller is charged of driving the WMR to get to its destination smoothly in the faster possible way by adjusting its linear and angular velocities (v, ω) as should a human driver do.

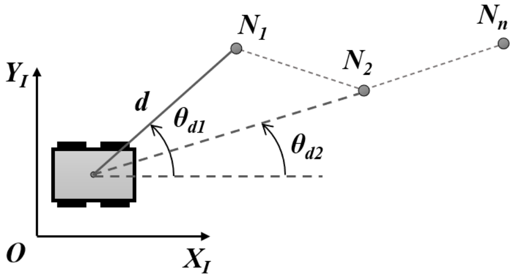


Fig. 4 - Parameters of the path following method

During its displacement, the mobile robot moves towards a current target waypoint until the distance to that waypoint gets lower than a certain threshold fixed depending on the robot geometry. When the distance to the current waypoint is within the fixed limit, the mobile robot switches to targeting the next waypoint.

The path can be seen as a set of straight lines and bends. On straight parts, the mobile robot can increase its speed (and decrease the angular velocity). While approaching a turn, the mobile robot must decrease its linear velocity and increase its angular velocity accordingly with the look-ahead curvature LAC of the turn so as not to overturn.

The estimation of the *LAC* is obtained by the absolute value of the difference of the values of the desired orientations to two consecutive waypoints (the current and the next one) θ_{d1} and θ_{d2} (Fig. 4). The LAC is calculated using the following expression:

$$LAC = \left|\theta_{d1} - \theta_{d2}\right| \tag{9}$$

θ_{d1} and θ_{d2} are defined as follows:

$$\theta_{d1} = \operatorname{atan2}\left(y_d(i) - y, x_d(i) - x\right) \tag{10}$$

$$\theta_{d2} = \operatorname{atan} 2\left(y_d\left(i+1\right) - y, x_d\left(i+1\right) - x\right) \tag{11}$$

here *x* and *y* represent the robot position, $x_d(i)$ and $y_d(i)$ the coordinates of the current target ***Nᵢ*** and $x_d(i+1)$ and $y_d(i+1)$ the coordinates of the next target N_{i+1}.

On the straight parts of the path, the *LAC* gets small values. However, its value will increase when the WMR approaches a bend or a corner (on the curved parts of the path).

5. Controller Design

The architecture of the path following controller is shown in Fig. 5. The calculation module has the role of providing the controller's normalized values of the input items by performing some computations on the coordinates of the current and the next waypoints and the current pose of the mobile robot. The normalized input items are obtained by multiplying the current input items values by scaling factors. The FLCs (heading and velocity controllers) generate the angular and linear velocities commands to drive the WMR toward its target waypoint.

To avoid the tedious task of the controller parameters optimization and to keep our design simple all base values are set to 0 and all the breadth are set to 1 for the DID modules. This means that all the input items play equal roles in the control of the WMR.

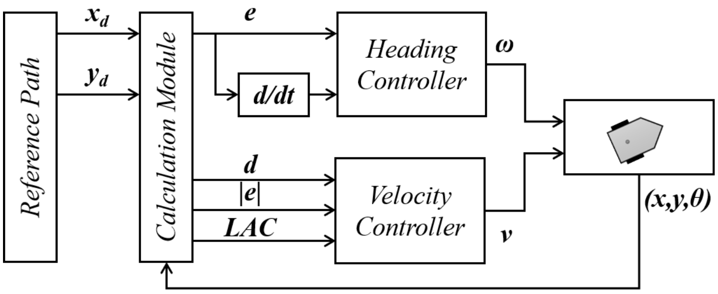


Fig. 5 -The path following controller

5.1 Heading controller

The heading controller acts like a conventional fuzzy Proportional Derivative (PD) controller. The heading controller takes in two input variables: the heading error *e*, and its derivative *ė*. The output of this controller is the angular velocity command (*ω*) that allows the WMR to correct its orientation to head for the actual target waypoint. The reference angular velocity varies from *-1* to *1 rad/s*. The heading error is given by:

$$e = \theta_{d1} - \theta \tag{12}$$

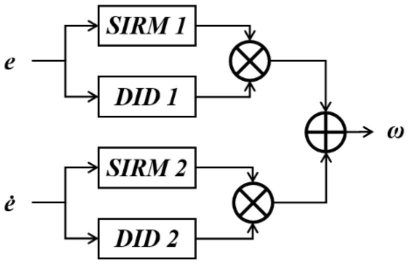


Fig. 6 - The structure of the heading controller

The structure of the heading controller module is shown on Fig. 6. The SIRMs of the input variables ***e*** and ***ė*** corresponding to the heading error and its derivative can be set up as in Table 1 which represents the inference rules for these modules. Here, the linguistic variables ***NE***, ***ZE***, ***PO*** represent the membership functions of each input variable which are shown in Fig. 7. Three triangular membership functions uniformly distributed in the normalized range [*-1.0, 1.0*] are used for the inputs. For the output item, three singletons (constants) are used as represented in Table 1.

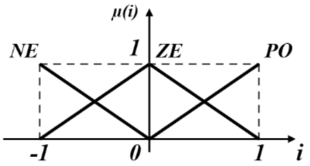


Fig. 7 - Membership functions for each SIRM of the heading controller

Table 1 - Fuzzy rules for the SIRMs of the input items of the heading controller

Input Variable **e and $\dot{e}$**	**Output Variable** **f_i**
NE	-1.0
ZE	0.0
PO	1.0

For the heading controller, the fuzzy rules for DID-1 and DID-2 of the input items e and $\dot{e}$ can be established as in Table 2 by selecting the absolute values of the input items as the antecedent variable. Here, the membership functions ***S***, ***M***, ***B*** are defined in the range [*0.0*, *1.0*] as shown in Fig. 8. For the output item of the DIDs, three singletons (constants) are used as represented in Table 2.

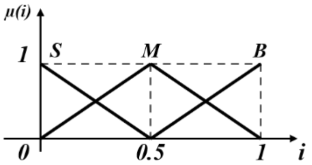


Fig. 8 - Membership functions for each dynamic variable of the heading controller

Table 2 - Fuzzy rules for the DID of the heading controller

Input Variable **$\lvert e\rvert$ and $\lvert\dot{e}\rvert$**	**Output Variable** **$\Delta\omega_i$**
S	0.0
M	0.5
B	1.0

5.2 Velocity controller

The velocity controller generates the robot's linear speed control signal (varying in the range [*0.0*, *1.0*] *m/s*). It takes in three input items: the distance to the current waypoint ***d***, the ***LAC***, and the absolute value of the orientation error ***e***.

Fig. 9 shows the structure of the velocity controller. The velocity controller uses two SIRM modules for the distance *d* and the *LAC* input items and one DID module taking in the absolute value of the orientation error. Note that only one DID module is used for both SIRMs modules.

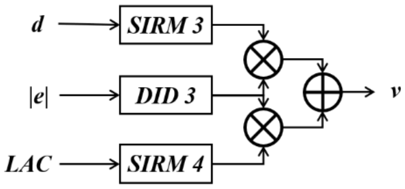


Fig. 9 - The structure of the velocity controller

The SIRMs of the two input items *d* and *LAC* can be set up as in Table 3 and Table 4 respectively. From these two tables, it can be noticed that the speed of the WMR is varied proportionally with the distance to the current waypoint and is varied inversely with the look-ahead curvature *LAC* at the current waypoint. From Table 5, it can be understood that when the absolute value of the orientation error is moderate or big, the dynamic degree of the input items is decreased so that the robot can correct its heading first. All the input items for this module use the membership functions of Fig. 8.

Table 3 - Fuzzy rules for the SIRM 3 module

Input Variable ***d***	Output Variable ***f**_i*
S	0.0
M	0.5
B	1.0

Table 4 - Fuzzy rules for the SIRM 4 module

Input Variable ***LAC***	Output Variable f_i
S	1.0
M	0.5
B	0.0

Table 5 - Fuzzy rules for the DID 3 modules

Input Variable $\|e\|$	Output Variable $\Delta\omega_i$
S	1.0
M	0.0
B	0.0

6. Simulation Results

In this section, the results of numerical simulations of the proposed path following controller are shown and compared with the results of the conventional fuzzy logic controller proposed in [23]. The simulations were performed using the robotics system toolbox of the MATLAB environment.

In the presented examples, both controllers; the SIRMs based controller and the FLC use the scaling factors shown in Table 6.

Table 6 - Control parameters

Input item	Scaling factor
e	10.0
$\dot{e}$	0.01
d	1.0
LAC	1.0

The initial pose of the mobile robot is: $\begin{bmatrix} x & y & \theta \end{bmatrix}^T = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$.

Fig. 10 shows the results of following an eight shaped path. The reference path is composed by discrete waypoints (red circles), where the reference path is described using the following equations:

$$\begin{cases} x_d = \dfrac{10\sin(t)}{\left(1+\cos^2(t)\right)} \\ y_d = \dfrac{10\sin(t)\cos(t)}{\left(1+\cos^2(t)\right)} \end{cases} \tag{13}$$

Here, ***t*** is a discrete number that varies in the interval [-*3.2, 3.2*] with a step of *0.2*.

The trajectories of the mobile robot controlled by the two control schemes show a big similitude in path following. The two controllers can drive the robot to follow smoothly the desired path.

Fig. 11 presents the mobile robot heading control; the desired orientation is computed online using Eq. (10). The current orientation of the robot is regulated by the heading controller which outputs the angular velocity command signal (Fig. 12). From these figures, it is noticed that the mobile robot controlled by the SIRMs based controller is faster in achieving its task than the one controlled by the conventional FLC. The SIRMs based controller achieves its goal in *108.9 s* while the FLC takes *180.4 s* to attain the goal.

Fig. 13 shows the linear velocity control signals generated by the FLC and the SIRMs based velocity controllers respectively. The differences in the shapes of the signals are due to inference methods of the FLC which uses a conventional fuzzy inference method and the proposed controller which is based on the SIRMs dynamically connected fuzzy inference model. These figures show also the advantage of the SIRMs based controller in executing the path faster than the conventional FLC.

Fig. 14 shows the result of following a discrete sinusoidal path. It is noticed the smoothness of the mobile robot path under the control of the two controllers.

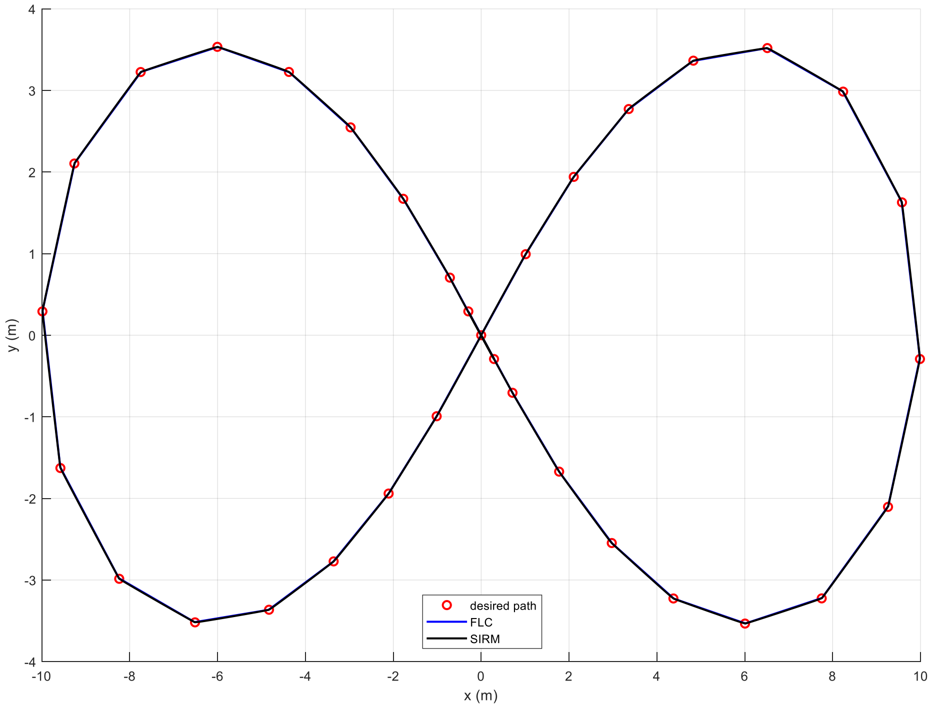
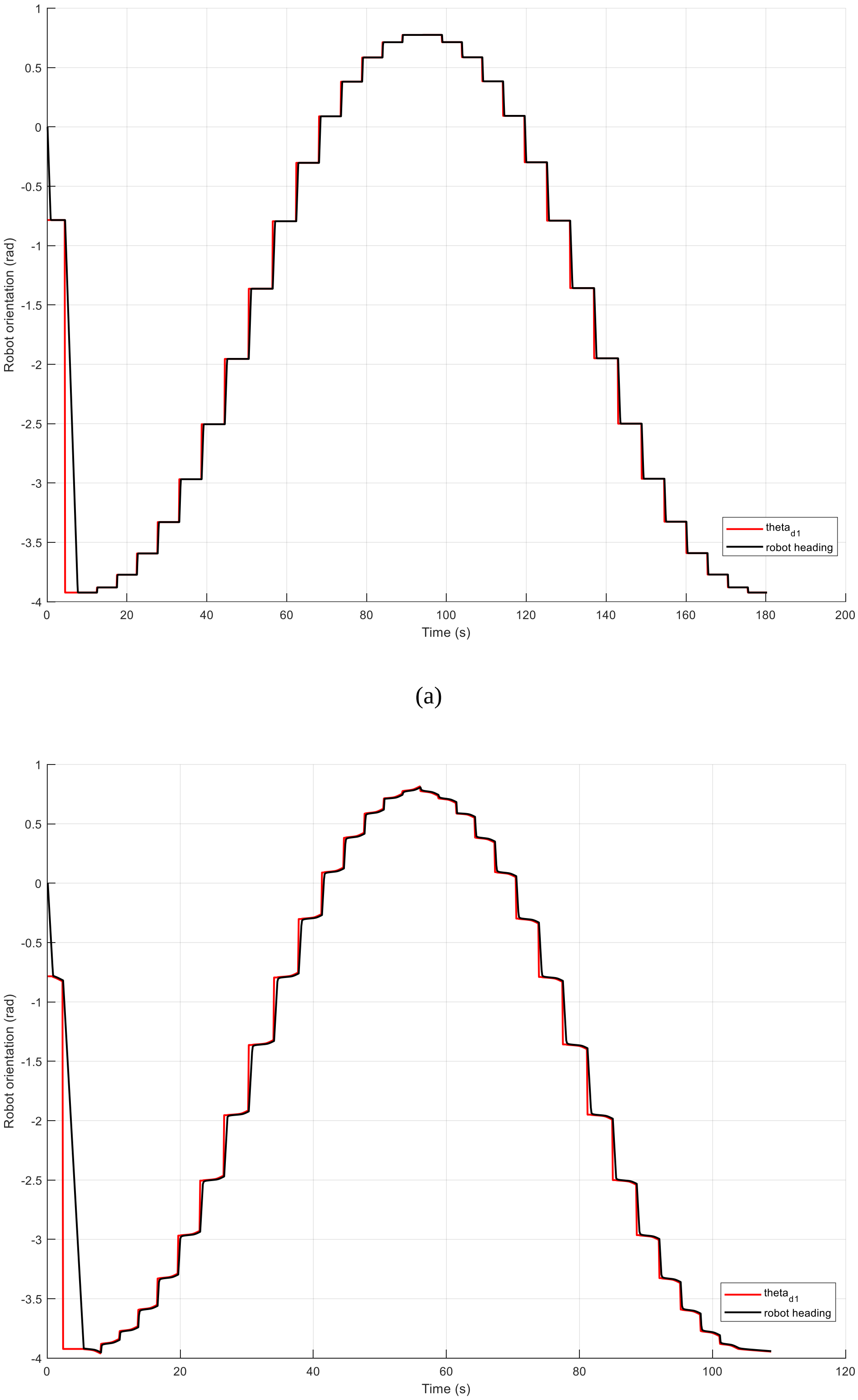


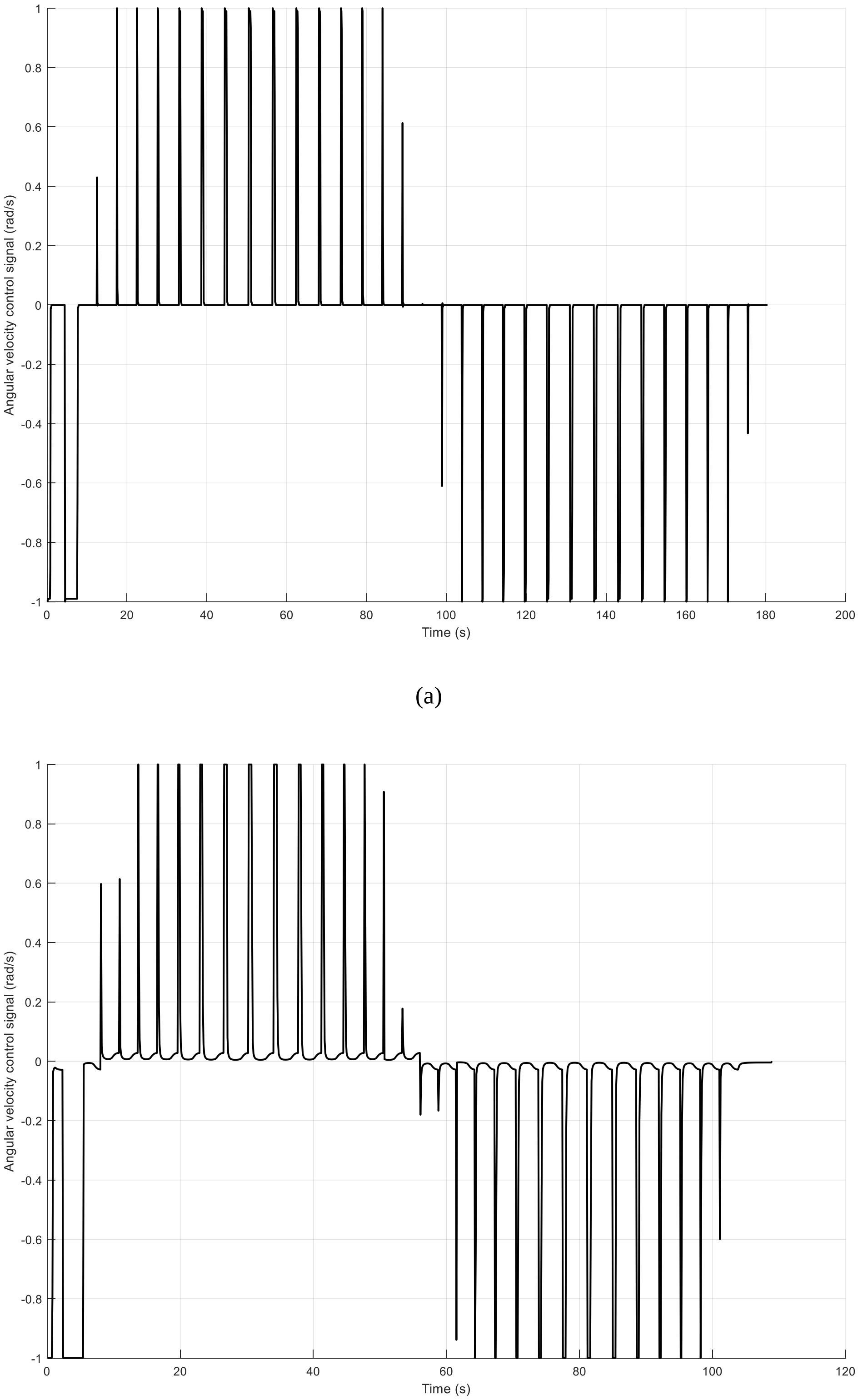
Fig. 10 - Following an 8 shaped path



(a)

(b)

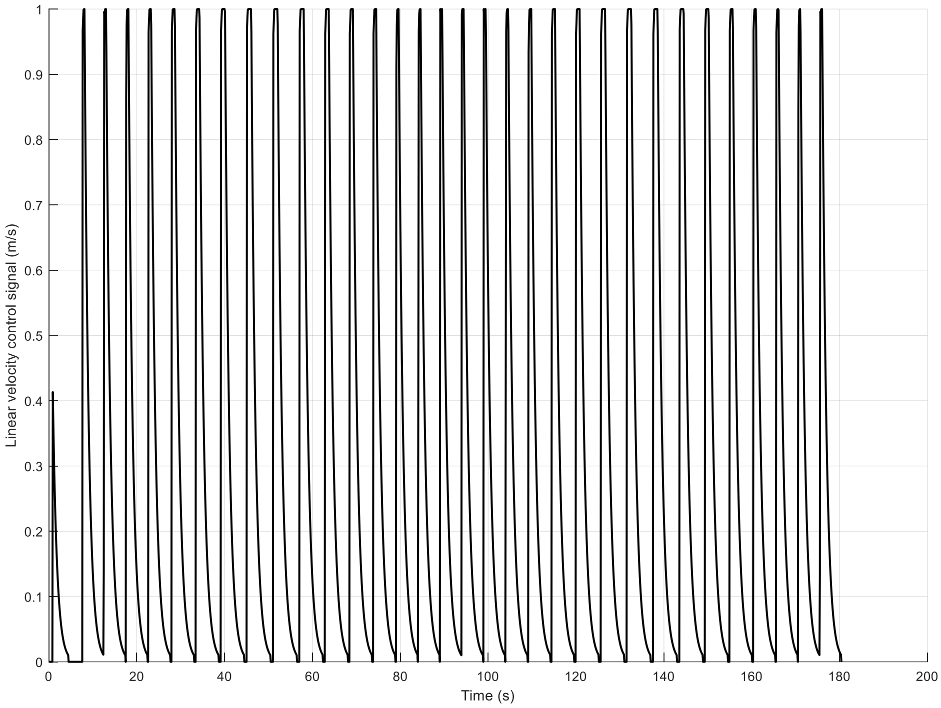
Fig. 11 - Robot heading control (a) FLC; (b) SIRM



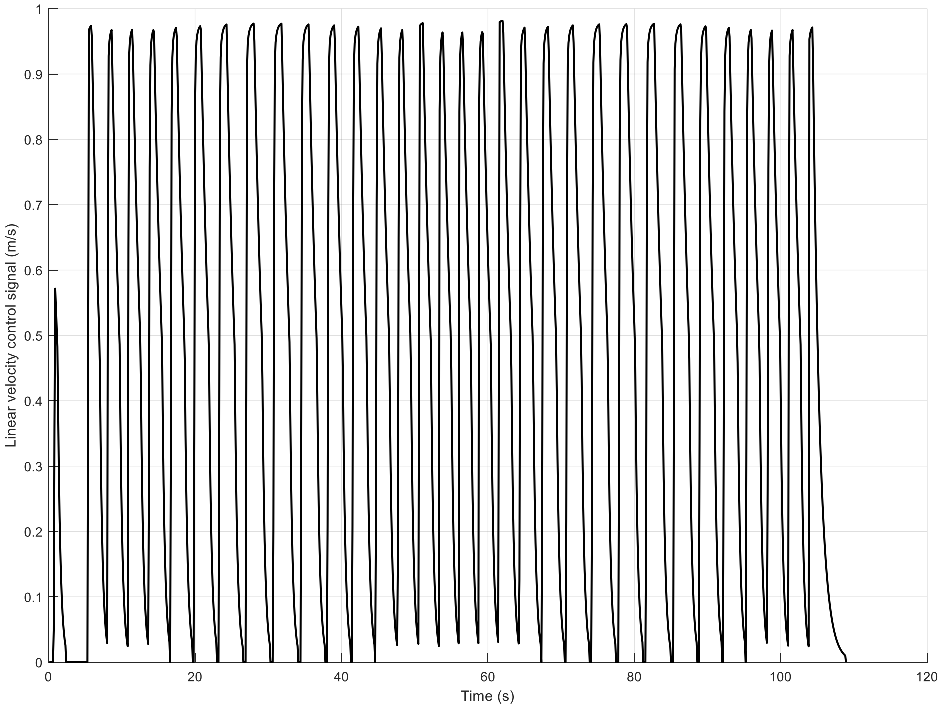
(a)

(b)

Fig. 12 - Angular velocity signals (a) FLC; (b) SIRM



(a)



(b)

Fig. 13 - Linear velocity command signals (a) FLC; (b) SIRM

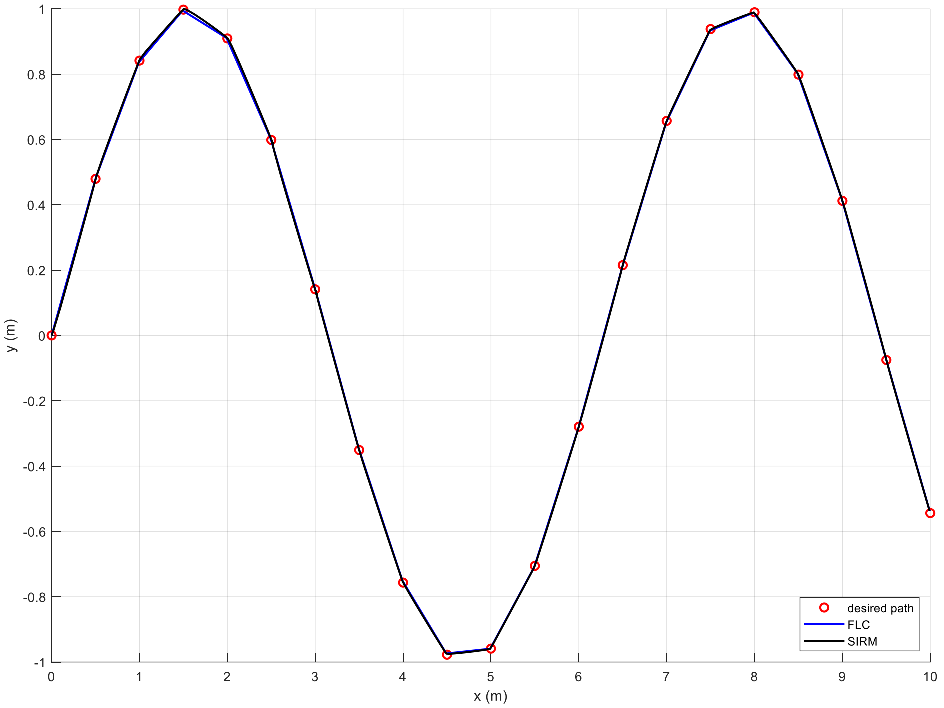


Fig. 14 - Following a sinusoidal path

7. Conclusion

This paper proposes a path following controller for differential drive wheeled mobile robots based on the SIRMs dynamically connected fuzzy inference model. This specific type of inference model is used to solve the problem of the increase of the number of inference rules in conventional fuzzy control for multi-input control systems. The structure of a SIRMs based control system allows also to reduce the processing time.

The proposed controller drives a differential drive WMR to follow a discrete reference path composed of an ordered series of distinct waypoints by adjusting the angular and the linear velocities of the robot. The controller is made up of two independent control unites working in parallel; the heading controller which generates the angular velocity command, and the velocity controller which is responsible of generating the linear velocity command. The controller uses a total number of 21 inference rules to imitate the driving way of a human driver.

The heading controller takes the orientation error and its derivative as input items and takes the angular velocity as the output item. Each input item is assigned with a *SIRM* and a *DID*.

The velocity controller has three input variables: the distance to the target waypoint, the estimate of the look ahead curvature (*LAC*), and the absolute value of the heading error. It generates as output the linear velocity control signal to apply to the WMR. The structure of the SIRMs inference model was modified for this controller such that the distance and *LAC* are assigned with *SIRMs* and a common *DID* which takes the absolute value of the orientation error as input item. The *SIRMs* and the *DIDs* of the velocity controller are set up in a such manner that the angular velocity control of the mobile robot gets the higher priority over the linear velocity control. The priority orders adjustment is automatically performed online according to the current navigation situation.

No optimization of the controller parameters is required since that all base values are set to 0 and all the breadths are set to 1 for the *DID* modules. This means that all the input items play equal roles in the control of the mobile robot.

The simulation results show that the proposed controller outperforms the conventional FLC path following controller in following precision and execution time of the path.

References

[1] Francisco Rubio, Francisco Valero and Carlos Llopis-Albert. (2019). A review of mobile robots: Concepts, methods, theoretical framework, and applications, International Journal of Advanced Robotic Systems, 16(2), 1-21. https://doi.org/10.1177/1729881419839596

[2] Alessandro De Luca, Giuseppe Oriolo, Marilena Vendittelli. (2001). RAMSETE: Articulated and Mobile Robotics for Services and Technology, in, Lecture Notes in Control and Information Sciences. Springer (Ed.), pp. 181-226.

[3] Ulrich Nehmzow. (2003). Mobile Robotics A Practical Introduction, Springer-Verlag London. https://doi.org/10.1007/978-1-4471-0025-6

[4] Joanna Płaskonka. (2015). Different Kinematic Path Following Controllers for a Wheeled Mobile Robot of (2,0) Type, Journal of Intelligent & Robotic Systems, 77(3-4), 481-498. https://doi.org/10.1007/s10846-013-9879-6

[5] Yuri A. Kapitanyuk, Anton V. Proskurnikov, Ming Cao. (2018). A Guiding Vector-Field Algorithm for Path-Following Control of Nonholonomic Mobile Robots, IEEE Transactions on Control Systems Technology, 26(4), 1372-1385. DOI: 10.1109/TCST.2017.2705059

[6] D. Soetanto, L. Lapierre, A. Pascoal. (2003). Adaptive, non-singular path-following control of dynamic wheeled robots, 42nd IEEE International Conference on Decision and Control, Maui, HI, 1765-1770.

[7] P. Encarnação, A.Pascoal. (2002). Combined Trajectory Tracking and Path Following for Underwater Vehicles, IFAC Proceedings Volumes, 35(2), 451-456.

[8] Morten Breivik, Thor I.Fossen. (2005). Guidance-based path following for wheeled mobile robots, IFAC Proceedings Volumes, 38(1), 13-18.

[9] Chang Boo, Danwei Wang. (2010). Maneuverability and Path Following Control of Wheeled Mobile Robot in the Presence of Wheel Skidding and Slipping, Journal of Field Robotics, 27(2), 127-144. https://doi.org/10.1002/rob.20329

[10] P. Coelho, U. Nunes. (2005). Path-Following Control of Mobile Robots in Presence of Uncertainties, IEEE TRANSACTIONS ON ROBOTICS, 21(2), 252-261. DOI: 10.1109/TRO.2004.837240

[11] Goran Huskić, Sebastian Buck, Matthieu Herrb, Simon Lacroix, Andreas Zell. (2019). High-speed path following control of skid-steered vehicles, The International Journal of Robotics Research, 38(9), 1124-1148. https://doi.org/10.1177/0278364919859634

[12] Wei Wang, Jiangshuai Huang, Changyun Wen. (2016). Prescribed performance bound-based adaptive path following control of uncertain nonholonomic mobile robots, INTERNATIONAL JOURNAL OF ADAPTIVE CONTROL AND SIGNAL PROCESSING, 31(5), 805-822. https://doi.org/10.1002/acs.2732

[13] Xing Wu, Peng Jin, Ting Zou, Zeyu Qi, Haining Xiao, Peihuang Lou. (2019). Backstepping Trajectory Tracking Based on Fuzzy Sliding Mode Control for Differential Mobile Robots, Journal of Intelligent & Robotic Systems, 96(4), 109-121. https://doi.org/10.1007/s10846-019-00980-9

[14] Shuyou Yu, Yang Guo, Lingyu Meng, Ting Qu, Hong Chen. (2018). MPC for Path Following Problems of Wheeled Mobile Robots, IFAC-PapersOnLine, 51(20), 247-252. https://doi.org/10.1016/j.ifacol.2018.11.021

[15] Fady Ibrahim, A. A. Abouelsoud, Ahmed M. R. Fath Elbab, Tetsuya Ogata. (2019). Path following algorithm for skid-steering mobile robot based on adaptive discontinuous posture control, Advanced Robotics, 33(9), 439-453. https://doi.org/10.1080/01691864.2019.1597764

[16] Wei Zhu, Xian Guo, Yongchun Fang, Xueyou Zhang. (2020). A Path-Integral-Based Reinforcement Learning Algorithm for Path Following of an Autoassembly Mobile Robot, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, 31(11), 4487-4499. DOI: 10.1109/TNNLS.2019.2955699

[17] Mahmut Dirik, Adnan Fatih Kocamaz, Oscar Castillo. (2020). Global Path Planning and Path-Following for Wheeled Mobile Robot Using a Novel Control Structure Based on a Vision Sensor, International Journal of Fuzzy Systems, 22(6), 1880-1891. https://doi.org/10.1007/s40815-020-00888-9

[18] Tiago T. Ribeiro, André G. S. Conceição. (2019). Nonlinear Model Predictive Visual Path Following Control to Autonomous Mobile Robots, Journal of Intelligent & Robotic Systems, 95(2), 731-743. https://doi.org/10.1007/s10846-018-0896-3

[19] Elie Maalouf, Maarouf Saad, Hamadou Saliah. (2006). A higher-level path tracking controller for a four-wheel differentially steered mobile robot, Robotics and Autonomous Systems, 54(1), 23-33. https://doi.org/10.1016/j.robot.2005.10.001

[20] Gianluca Antonelli, Stefano Chiaverini, Giuseppe Fusco. (2007). A Fuzzy-Logic-Based Approach for Mobile Robot Path Tracking, IEEE TRANSACTIONS ON FUZZY SYSTEMS, 15(2), 211-221. DOI: 10.1109/TFUZZ.2006.879998

[21] I. Susnea; A. Filipescu, G. Vasiliu, S. Filipescu. (2008). Path Following, Real-Time, Embedded Fuzzy Control of a Mobile Platform Wheeled Mobile Robot, 2008 IEEE International Conference on Automation and Logistics, Qingdao, 268-272.

[22] Hassan Talebi Abatari, Abdolreza Dehghani Tafti. (2013). Using a fuzzy PID controller for the path following of a car-like mobile robot, First RSI/ISM International Conference on Robotics and Mechatronics (ICRoM), Tehran, 189-193.

[23] A. Benmakhlouf, A. Louchene. (2016). Simple fuzzy logic based path tracking controller for a mobile robot, Journal of electrical engineering, 16(3), 210-217.

[24] Jianqiang Yi, Naoyoshi Yubazaki, Kaoru Hirota. (2002). A proposal of SIRMs dynamically connected fuzzy inference model for plural input fuzzy control, Fuzzy Sets and Systems, 125(1), 79-92. https://doi.org/10.1016/S0165-0114(00)00135-4

[25] Hirosato Seki. (2013). Nonlinear Identification Using Single Input Connected Fuzzy Inference Model, Procedia Computer Science, 22, 1121-1125. https://doi.org/10.1016/j.procs.2013.09.198

[26] Chengdong Li, Weina Ren, Jianqiang Yi, Guiqing Zhang, Fang Shang. (2015). Data-Driven Optimization of SIRMs Connected Neural-Fuzzy System with Application to Cooling and Heating Loads Prediction, In Advances in Neural Networks, Lecture Notes in Computer Science, Springer (Ed.), pp. 499-507.

[27] N. Yubazaki, Jianqiang Yi, M. Otani, N. Unemura, K. Hirota. (1997). Trajectory tracking control of unconstrained objects based on the SIRMs dynamically connected fuzzy inference model, Proceedings of 6th International Fuzzy Systems Conference, Barcelona, 609-614.

[28] N. Yubazaki, J. Yi, K. Hirota. (1998). SIRMs dynamically connected fuzzy inference model and PID controller, IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Anchorage, AK, USA, 325-330.

[29] Jianqiang Yi, Naoyoshi Yubazaki, Kaoru Hirota. (2001). Upswing and stabilization control of inverted pendulum system based on the SIRMs dynamically connected fuzzy inference model, Fuzzy Sets and Systems, 122(1), 139-152. https://doi.org/10.1016/S0165-0114(00)00049-X

[30] Jianqiang Yi, Naoyoshi Yubazaki, Kaoru Hirota. (2001). Stabilization control of series-type double inverted pendulum systems using the SIRMs dynamically connected fuzzy inference model, Artificial Intelligence in Engineering, 15(3), 297-308. https://doi.org/10.1016/S0954-1810(01)00021-8

[31] Jianqiang Yi, Naoyoshi Yubazaki, Kaoru Hirota. (2001). Stabilization control of ball and beam systems, Proceedings Joint 9th IFSA World Congress and 20th NAFIPS International Conference, Vancouver, BC, Canada, 2229-2234.

[32] Jianqiang Yi, Naoyoshi Yubazaki, Kaoru Hirota. (2002). A new fuzzy controller for stabilization of parallel-type double inverted pendulum system, Fuzzy Sets and Systems, 126(1), 105-119. https://doi.org/10.1016/S0165-0114(01)00028-8

[33] Jianqiang Yi, Naoyoshi Yubazaki, Kaoru Hirota. (2003). Anti-swing and positioning control of overhead traveling crane, Information Sciences, 155(1-2), 19-42. https://doi.org/10.1016/S0020-0255(03)00127-0

[34] Chengdong Li, Jianqiang Yi, Dongbin Zhao. (2009). Control of the TORA system using SIRMs based type-2 fuzzy logic, IEEE International Conference on Fuzzy Systems, Jeju Island, 694-699.

[35] Chengdong Li, Junlong Gao, Jianqiang Yi, Guiqing Zhang. (2018). Analysis and Design of Functionally Weighted Single-Input-Rule-Modules Connected Fuzzy Inference Systems, IEEE Transactions on Fuzzy Systems, 26(1), 56-71. DOI: 10.1109/TFUZZ.2016.2637369

[36] Chengdong Li, Guiqing Zhang, Jianqiang Yi, Tiechao Wang. (2011). On the Properties of SIRMs Connected Type-1 and Type-2 Fuzzy Inference Systems, IEEE International Conference on Fuzzy Systems, Taipei, Taiwan, 1982-1988.

[37] Jianqiang Yi, N. Yubazaki, K. Hirota. (2001). Stability analysis of SIRMs dynamically connected fuzzy inference model, 10th IEEE International Conference on Fuzzy Systems, Melbourne, Victoria, Australia, 143-146.

[38] Chengdong Li, Jianqiang Yi, Tiechao Wang. (2010). Stability analysis of SIRMs based type-2 fuzzy logic control systems, International Conference on Fuzzy Systems, Barcelona, 1-7.