VOLUME 19, N° 2 2025
Journal of Automation, Mobile Robotics and Intelligent Systems

Autonomous Goal Following for A Quadruped Robot Using Fuzzy Proportional Control

Submitted: 16th August 2024; accepted: 23rd November 2024

Jose Eduardo Lopez-Ramos, Oscar Castillo, Patricia Melin

DOI: 10.14313/jamris-2025-017

Abstract:
In this paper, a fuzzy Proportional Controller was designed and implemented for dynamically adapting the velocity and motion parameters in a quadruped robot for autonomously following a goal. The FNK0050 Freenove quadruped robot was utilized for the experiments, which has 12 degrees of freedom and thus higher complexity. Experimental results show that the proposed fuzzy controller surpasses the standard PID controller provided as default by the manufacturer of the robot.

Keywords: *fuzzy control, fuzzy proportional, quadruped robot*

1. Introduction

Quadruped robots are machines designed to imitate the movement of four-legged animals. These robots have four extremities, each of them with their own sensors and actuators, that enable a movement similar to quadruped animals, such as dogs, cats or horses [1–4]. Quadruped robots have been receiving increasing attention in the robotics area because they can have many real applications, such as search and rescue, exploration, agriculture and entertainment [5–7].

Quadruped robots are designed with the goal of imitating the real movement of quadruped animals and require appropriate control algorithms for achieving stability and equilibrium while moving [8–10]. There are several kinds of quadruped robots in the literature, but we selected one that has not been previously considered with fuzzy logic. This robot is called Robot Dog Kit FNK0050 from Freenove, for which only a proportional-integral-derivative (PID) control existed in the previous literature. This is an existing research gap in the state of the art that we decided to consider as our research work.

The contribution of this paper is the proposed design of a fuzzy proportional derivative (PD) controller for the goal following problem, which can be viewed as an enhancement to the existing PID controller. The main idea is that fuzzy logic enables having a nonlinear control model, which can provide better results for complex problems. In this way, fuzzy PD control surpasses the traditional linear PID control in the goal following task.

The rest of the document is structured as follows: Section 2 explains the quadruped robot utilized in this work. Section 3 outlines the existing PID controller for this robot and then the proposal for a fuzzy PD controller. Section 4 summarizes the experimental results, and Section 5 offers the conclusions and future work.

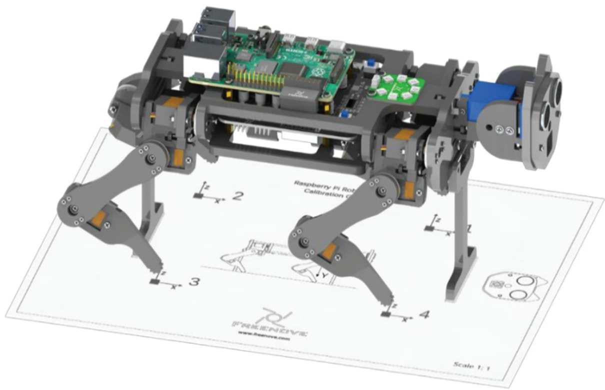

Figure 1. Robot Dog side view

2. Quadruped Robots and Problem Definition

In this section, we describe the particular quadruped robot utilized in this research work. The Robot Dog Kit FNK0050 is one of the models from Freenove, which is a quadruped design with open code that is compatible with Raspberry Pi. It has an acrylic light structure with several sensors, such as a camera, an ultrasound sensor, a gyroscope and an accelerometer. The robot has a total of 12 degrees of freedom, which makes it an ideal system for low-cost robotic applications. Figure 1 shows a side-view of the robot dog after assembly. Figure 2 depicts a front view of the robot.

Controlling the stability of quadruped robots is very important for several reasons: stability will contribute to reduced likelihood of falling, robustness against external perturbations, energy efficiency and improved performance. In addition, controlling the motion of the robot in following a trajectory to achieve a particular goal is very important. In this paper, we are addressing this last problem by providing the robot with a controller that will make the robot autonomously move to achieve a goal. Figure 3 illustrates the control problem, which basically consists of reaching the value of an input command starting from an initial point. In Figure 4, we illustrate how

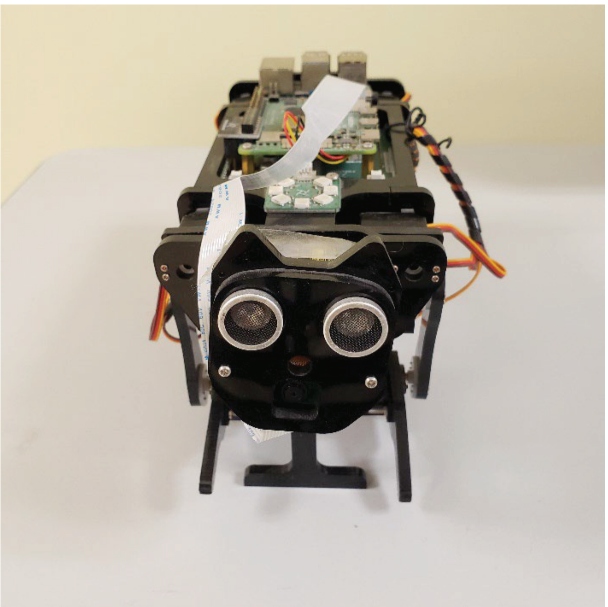

Figure 2. Robot Dog front view

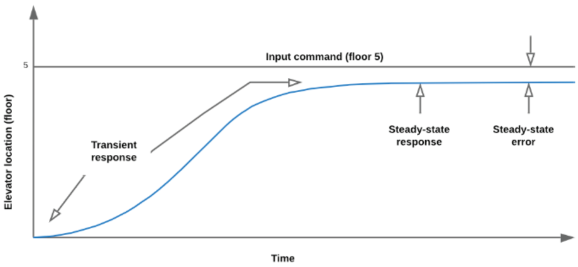

Figure 3. Control problem definition

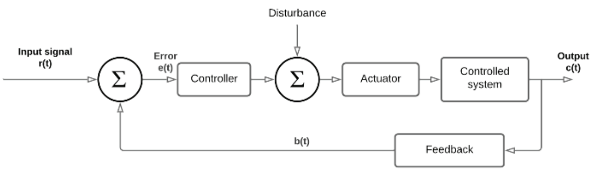

Figure 4. Effect of disturbances on the controller

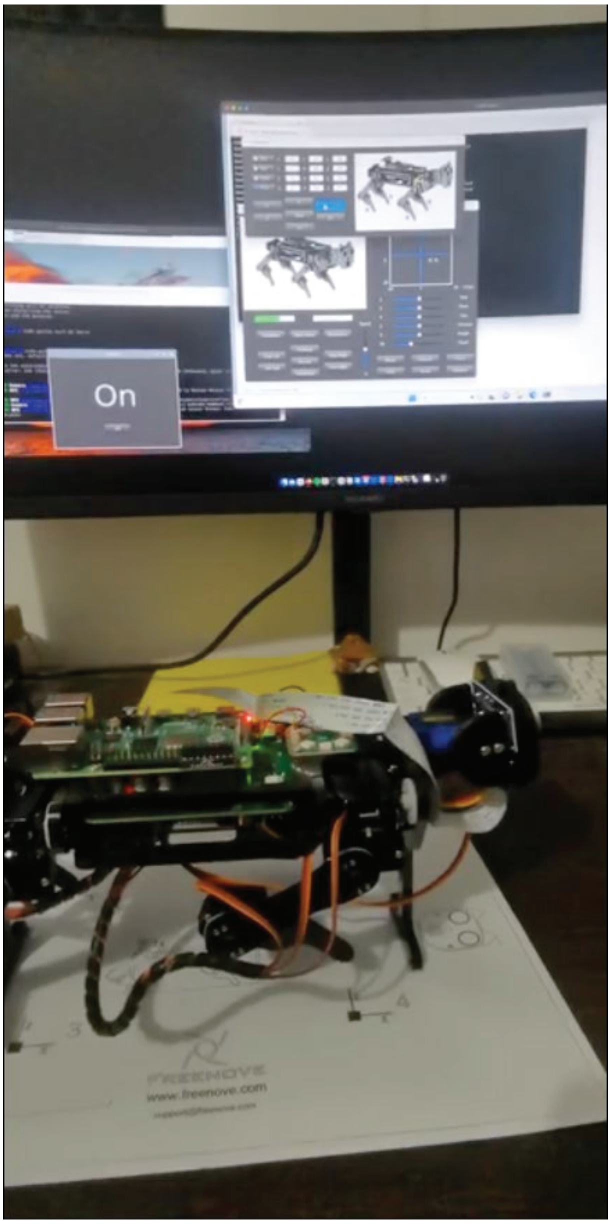

Figure 5. Controlling the robot with the server

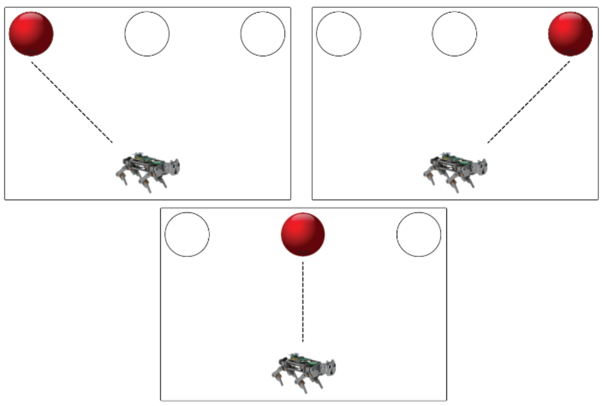

Figure 6. Possible robot trajectories

disturbances can affect the behavior of the controller in the closed feedback loop.

In Figure 5 we show the actual implementation of the controllers in the server to control the physical robot.

For the test scenario, we created two configurations: the first one with a distance of 1 meter with the objective on the side and an irregular terrain, and the second one with a distance of 80 centimeters with the goal in front and a plain terrain. These configurations could generate trajectories like the ones illustrated in Figure 6.

3. Proposed Fuzzy PD Controller

In this section, we describe the implementation of the two fuzzy PD controllers, one for controlling the direction and velocity on the x axis (to maintain the red ball centered in the robot view), and the other to control the direction and velocity on the z axis (to position the robot in the front of the objective at a desired distance.

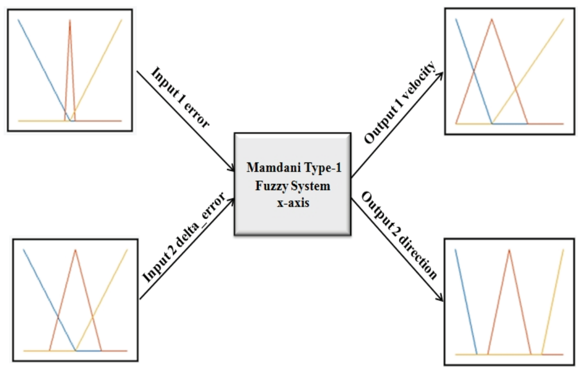

Figure 7. Structure of the fuzzy PD controller (x axis)

3.1. Fuzzy PD Controller on x Axis

The fuzzy PD controller for the *x* axis is structured as illustrated in Figure 7, containing two inputs and two outputs. The first input is the *error*, and the second input is the change of the error called *delta_error*. The first output is the velocity of turning, and the second output is used to control the direction. The fuzzy system is of type-1 Mamdani form.

The fuzzy rules that were used in the fuzzy PD controller to position the goal in the center of the *x* axis are listed below.

1) If error is left and *delta_error* is low then *velocity* is medium, *direction* is right
2) If *error* is left and *delta_error* is medium then *velocity* is low, *direction* is right
3) If *error* is left and *delta_error* is high then *velocity* is low, *direction* is right
4) If *error* is center and *delta_error* is low then *velocity* is medium, *direction* is center
5) If *error* is center and *delta_error* is medium then *velocity* is low, *direction* is center
6) If *error* is center and *delta_error* is high then *velocity* is low, *direction* is center
7) If *error* is right and *delta_error* is low then *velocity* is medium, *direction* is left
8) If *error* is right and delta_error is medium then *velocity* is low, *direction* is left
9) If *error* is right and *delta_error* is high then *velocity* is low, *direction* is left

In Figure 8 we illustrate the nonlinear surface of the fuzzy PD controller for the *x* axis.

3.2. Fuzzy PD Controller on x Axis

The fuzzy PD controller for the *z* axis is structured as illustrated in Figure 9, containing two inputs and two outputs. The first input is the *error*, and the second input is the change of the error called *delta_error*. The first output is the *velocity* of motion in the *z* axis, and the second output is used to control the direction. The fuzzy system is of type-1 Mamdani form.

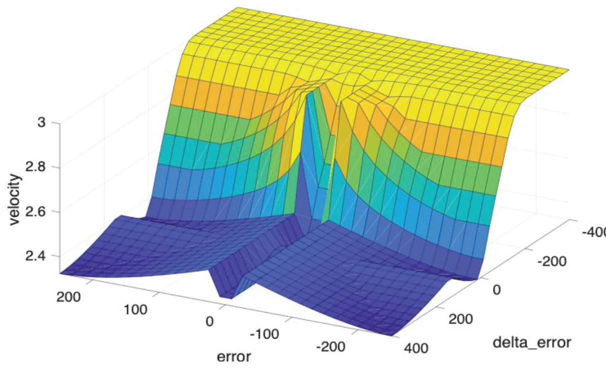

Figure 8. Surface of the controller for the velocity

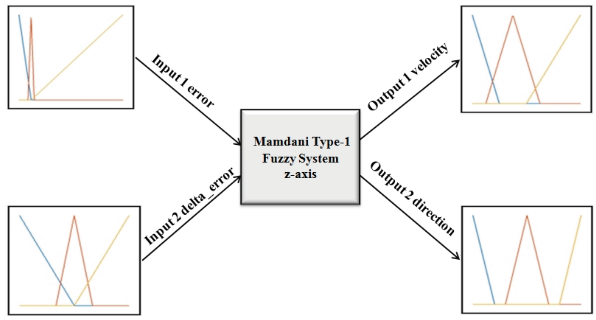

Figure 9. Input and outputs of the fuzzy PD controller (z axis)

The fuzzy rules that were used in the fuzzy PD controller to position the goal in the center of the *z* axis are listed below

1) If *error* is left and *delta_error* is low then *velocity* is medium, *direction* is backward
2) If *error* is left and *delta_error* is medium then *velocity* is low, *direction* is backward
3) If *error* is left and *delta_error* is high then *velocity* is low, *direction* is backward
4) If *error* is center and *delta_error* is low then *velocity* is medium, *direction* is stay
5) If *error* is center and *delta_error* is medium then *velocity* is low, *direction* is stay
6) If *error* is center and *delta_error* is high then *velocity* is low, *direction* is stay
7) If *error* is right and *delta_error* is low then *velocity* is medium, *direction* is forward
8) If *error* is right and *delta_error* is medium then *velocity* is low, *direction* is forward
9) If *error* is right and *delta_error* is high then *velocity* is low, *direction* is forward

In the same form as with the fuzzy PD controller for the *x* axis, the rules for the *z* axis were obtained by physically experimenting with the FNK0050 robot to avoid slipping and collision during the movement on the trajectory.

In Figure 10, we illustrate the nonlinear surface of the fuzzy PD controller for the *z* axis.

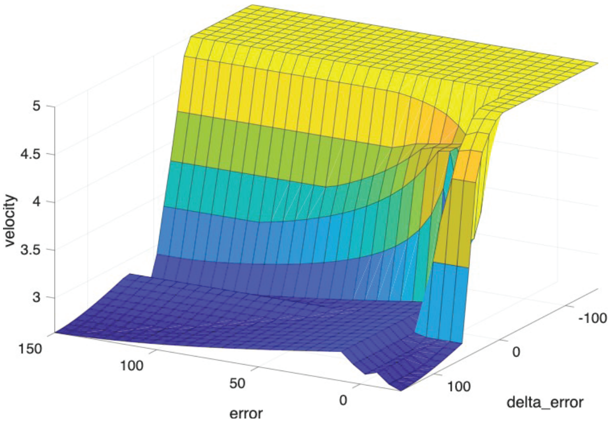

Figure 10. Control surface for the velocity for the z axis

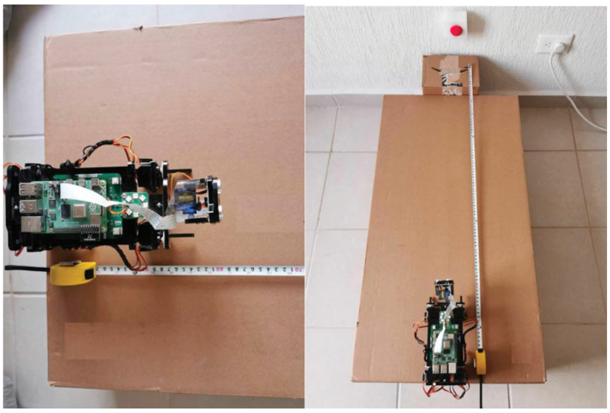

Figure 11. Test scenario for the experiments

4. Experimental Results

In this section, we summarize the results that were obtained with the controllers mentioned in Section 3. These results were achieved by performing physical tests with the FNK0050 robot platform at distances of 80 cm and 100 cm, respectively. In Figure 11, we illustrate the test scenario for experimenting with the controllers.

4.1. Test with PID Control

The PID controller was tested with two different scenarios, the first one is to find the objective in a straight line with a distance of 80 cm on a plain surface, and the second one is to find the objective that is positioned on the sides at a distance of 1 m on a rough surface.

In the first case, we performed 30 tests with a distance of 80 cm and the objective positioned in front of the robot, and in all cases, the robot reached the objective with an average time of 24.2056 seconds. It is worth mentioning that this controller (by default) gives priority to the distance to the objective before centering the objective on the *x* axis.

In the second case, we performed 40 tests with a distance of 1 m and the objective positioned on the left side of the robot. In this situation, the robot was not able to reach the objective in all the tests, which we believe is because the default controller gives priority to distance over centering the objective.

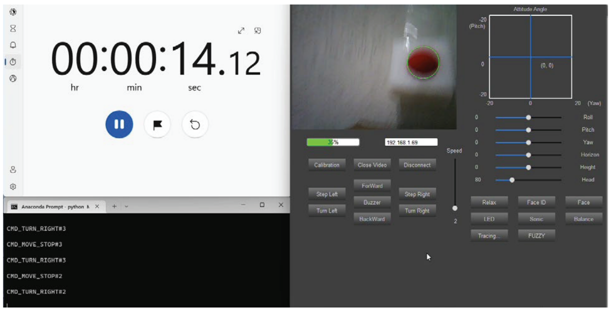

Figure 12. Fuzzy PD controller adjusting the position with respect to the ball in x axis

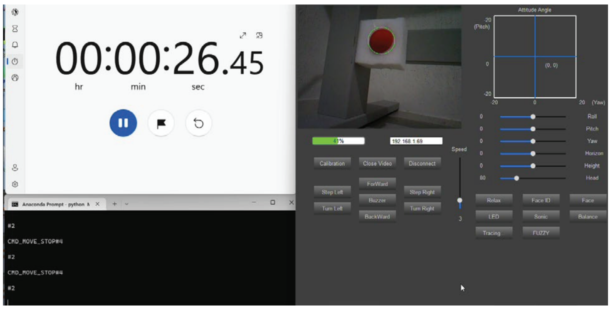

Figure 13. Fuzzy PD controller performance in the 1 m distance case

4.2. Test with Fuzzy PD Control

The fuzzy PD controllers were tested with the same scenarios as for the default controller of the robot: the first one with the straight-line trajectory of 80 cm on a plain surface, and the second one with a lateral trajectory on a rough surface.

4.2.1. Track of 80 cm

We performed 30 tests with a distance of 80 cm and the objective positioned in front of the robot, and in all cases, the robot reached the objective with an average time of 13.4796 seconds. The acceleration oscillated between 2 and 5, and the velocity between 2 and 3. The main advantages of the fuzzy PID controller were the dynamic adjustment of velocities and the adjustment to the position of the objective in the *x* axis, which enabled better movement of the robot, as shown in Figure 12.

4.2.2. Track of 100 cm

The fuzzy PD controller was used in 40 tests on a rough surface, which could randomly add noise to the motion of the robot by affecting the leg movement of the robot. Even with the noise, the controller was able to reach the objective in all cases with an average time of 29.5325 seconds. In Figure 13, we see the result of one of the tests.

4.3. Comparison of results

Tables 1 and 2 show the results of each test for the above-mentioned controllers, respectively. The results with underline and bold are the minimal times in reaching the objective, and in bold are the maximum times.

Table 1. Results for the experiments with 80 cm

Track of 80 cm		
Tests	PID	Fuzzy PD
1	26.1	12.45
2	28.39	12.26
3	24.61	11.54
4	22.43	11.51
5	26.71	**10.31**
6	26.43	12.72
7	22.73	10.63
8	21.04	11.16
9	20.05	13.87
10	**19.42**	15.84
11	22.71	13.52
12	22.91	12.70
13	20.76	14.55
14	21.36	12.66
15	**30.03**	13.39
16	25.47	14.07
17	23.00	**17.08**
18	21.43	16.62
19	24.01	12.71
20	26.51	15.32
21	27.09	13.42
22	20.82	15.71
23	23.97	14.47
24	26.47	13.63
25	24.78	14.84
26	24.56	12.77
27	23.11	14.16
28	27.17	13.89
29	29.47	14.69
30	22.63	11.90
$\overline{X}$	24.2056	13.4796
s	2.8009	1.6871

The results obtained in the experiments with a scenario of 80 cm show a clear advantage in the average times of the fuzzy PD controller with respect to the PID controller. This statement is true despite this scenario being ideal for the PID controller. Note that the PD controller was better in all cases as well as on average. The reason for the superiority of the fuzzy PD controller is that it dynamically changes the velocity values according to the situation. A statistical test comparing the averages produces a t value of 17.96 and p value of 3.82×10^{-23}, which are evidence of the superiority of the proposed fuzzy controller.

In Table 2, the acronym NAO (not able objective) is used to represent that the controller is not able to reach the objective. The results from Table 2 show that the PID controller is not able to reach the objective in all cases, which is due to the limited capability that it has to adapt to noisy situations (terrain with perturbations). On the other hand, the fuzzy PD controller is able to reach the objective in all cases, although it takes more time than in the previous table, and it is able to adapt to noisy situations.

Table 2. Results for the experiments with 1 m

Track of 1 m		
Tests	Fuzzy PD	PID
1	26.28	NAO
2	28.12	NAO
3	**18.62**	NAO
4	36.12	NAO
5	26.50	NAO
6	33.47	NAO
7	30.47	NAO
8	27.02	NAO
9	32.56	NAO
10	35.92	NAO
11	22.87	NAO
12	31.83	NAO
13	33.00	NAO
14	31.26	NAO
15	33.33	NAO
16	27.30	NAO
17	29.89	NAO
18	20.31	NAO
19	23.18	NAO
20	33.55	NAO
21	25.78	NAO
22	36.65	NAO
23	25.83	NAO
24	31.04	NAO
25	23.51	NAO
26	**38.31**	NAO
27	32.30	NAO
28	26.93	NAO
29	36.39	NAO
30	33.30	NAO
31	26.70	NAO
32	32.59	NAO
33	28.74	NAO
34	31.02	NAO
35	33.36	NAO
36	27.88	NAO
37	28.07	NAO
38	23.83	NAO
39	29.32	NAO
40	28.15	NAO
$\overline{X}$	29.5325	NAO
s	4.6036	NAO

5. Conclusion

In this paper, we presented fuzzy PD control for the problem of a quadruped robot. A fuzzy PD controller was designed and implemented with fuzzy logic to enhance its performance with respect to traditional linear controllers. The problem is very important to achieve efficient movement of the robot, as well as to minimize energy usage. A comparison of the designed fuzzy PD controller with respect to the PID controller was presented to verify the superiority of the proposal.

The experimentation was performed with a robot called FNK0050 for which only a PID controller was previously used, so the contribution is in the enhancement of the control using fuzzy logic. Future works include optimizing the design of the fuzzy PID controller, as well as elevating its design to type-2 [12–15] and possibly type-3 fuzzy logic [16–18] with the goal of handling higher levels of uncertainty in the control process . Finally, we envision optimizing the fuzzy PID design with metaheuristics, as in [19–21].

AUTHORS

Jose Eduardo Lopez-Ramos – Tijuana Institute of Technology, TecNM, Tijuana, 22379, Mexico, e-mail: m22210010@tectijuana.edu.mx.

Oscar Castillo* – Tijuana Institute of Technology, TecNM, Tijuana, 22379, Mexico, e-mail: ocastillo@tectijuana.mx.

Patricia Melin – Tijuana Institute of Technology, TecNM, Tijuana, 22379, Mexico, e-mail: pmelin@tectijuana.mx.

*Corresponding author

ACKNOWLEDGEMENTS

This work was supported by TecNM with a research grant and Conacyt with a scholarship for Jose Eduardo Lopez Ramos.

References

[1] J. T. Machado and M. F. Silva, "An Overview of Legged Robots," *Proc. Int. Symp. Math. Methods Eng.*, vol. 48, no. 2.3, Apr. 2006.

[2] P. G. De Santos, E. Garcia, and J. Estremera, *Quadrupedal Locomotion: An Introduction to the Control of Four-Legged Robots*, vol. 1. London: Springer, 2006.

[3] S. Hirose, Y. Fukuda, K. Yoneda, A. Nagakubo, H. Tsukagoshi, K. Arikawa, R. Hodoshima et al., "Quadruped Walking Robots at Tokyo Institute of Technology," *IEEE Robot. Autom. Mag.*, vol. 16, no. 2, 2009, pp. 104–114.

[4] J. Ingvast, C. Ridderström, and J. Wikander, "The Four-Legged Robot System WARP1 and Its Capabilities," *Proc. Second Swedish Workshop on Autonomous Systems*, Oct. 2002.

[5] M. P. Murphy, A. Saunders, C. Moreira, A. A. Rizzi, and M. Raibert, "The Littledog Robot," *Int. J. Robot. Res.*, vol. 30, no. 2, 2011, pp. 145–149.

[6] B. Katz, J. Di Carlo, and S. Kim, "Mini Cheetah: A Platform for Pushing the Limits of Dynamic Quadruped Control," *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2019, pp. 6295–6301.

[7] P. Biswal and P. K. Mohanty, "Development of Quadruped Walking Robots: A Review," *Ain Shams Eng. J.*, vol. 12, no. 2, 2021, pp. 2017–2031.

[8] K. Arphakorn, P. Amornphun, and J. Wisanu, "Gait Control of a Four-Legged Robot with Fuzzy-PID Controller," *Proc. Int. Conf. Artificial Life Robot.*, vol. 25, 2020, pp. 514–518.

[9] K. Chang, X. J. Han, and Y. Yang, "Self-adaptive PID Control Of Hydraulic Quadruped Robot," *Appl. Mech. Mater.*, vol. 496, 2014, pp. 1407–1412.

[10] K. Y. Chen and C. Y. Tsui, "The Fuzzy Control Approach for a Quadruped Robot Guide Dog," *Int. J. Fuzzy Syst.*, vol. 23, 2021, pp. 1789–1796.

[11] O. Montiel, R. Sepulveda, P. Melin, O. Castillo, M. A. Porta García, and I. M. Meza-Sánchez, "Performance of a Simple Tuned Fuzzy Controller and a PID Controller on a DC Motor," *Proc. IEEE Symp. Found. Comput. Intell. (FOCI)*, 2007, pp. 531–537.

[12] T. Kumbasar and H. Hagras, "Interval Type-2 Fuzzy PID Controllers," *Springer Handbook of Computational Intelligence*, J. Kacprzyk and W. Pedrycz, eds. Berlin, Germany: Springer, 2015.

[13] P. Melin and O. Castillo, "A New Method for Adaptive Control of Non-Linear Plants Using Type-2 Fuzzy Logic and Neural Networks," *Int. J. Gen. Syst.*, vol. 33, no. 2–3, 2004, pp. 289–304.

[14] L. Astudillo, O. Castillo, P. Melin, A. Alanis, J. Soria, and L. T. Aguilar, "Intelligent Control of an Autonomous Mobile Robot Using Type-2 Fuzzy Logic," *Eng. Lett.*, vol. 13, no. 3, 2006.

[15] R. Sepúlveda, O. Montiel, O. Castillo, and P. Melin, "Embedding a High Speed Interval Type-2 Fuzzy Controller for a Real Plant into an FPGA," *Appl. Soft Comput.*, vol. 12, no. 3, pp. 988–995, 2012.

[16] A. Mohammadzadeh, M. H. Sabzalian, and W. Zhang, "An Interval Type-3 Fuzzy System and a New Online Fractional-Order Learning Algorithm: Theory And Practice," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 9, pp. 1940–1950, Sep. 2020.

[17] O. Castillo, J. R. Castro, and P. Melin, *Interval Type-3 Fuzzy Systems: Theory and Design*. Cham, Switzerland: Springer, 2022.

[18] O. Castillo, J. R. Castro, and P. Melin, "Interval Type-3 Fuzzy Control for Automated Tuning of Image Quality in Televisions," *Axioms*, vol. 11, 2022, p. 276.

[19] F. Valdez, P. Melin, and O. Castillo, "Evolutionary Method Combining Particle Swarm Optimization and Genetic Algorithms Using Fuzzy Logic for Decision Making," *Proc. IEEE Int. Conf. Fuzzy Syst.*, 2009, pp. 2114–2119.

[20] F. Valdez, J. C. Vazquez, P. Melin, and O. Castillo, "Comparative Study of the Use of Fuzzy Logic in Improving Particle Swarm Optimization Variants for Mathematical Functions Using Co-Evolution," *Appl. Soft Comput.*, vol. 52, 2017, pp. 1070–1083.

[21] D. Sanchez, P. Melin, and O. Castillo, "A Grey Wolf Optimizer for Modular Granular Neural Networks for Human Recognition," *Comput. Intell. Neurosci.*, 2017, https://doi.org/10.1155/2017/4180510.