Intelligent Control of Nonlinear Dynamic Plants Using a Hierarchical Modular Approach and Type-2 Fuzzy Logic

Leticia Cervantes, Oscar Castillo, and Patricia Melin

Tijuana Institute of Technology
ocastillo@tectijuana.mx

Abstract. In this paper we present simulation results that we have at this moment with a new approach for intelligent control of non-linear dynamical plants. First we present the proposed approach for intelligent control using a hierarchical modular architecture with type-2 fuzzy logic used for combining the outputs of the modules. Then, the approach is illustrated with two cases: aircraft control and shower control and in each problem we explain its behavior. Simulation results of the two case show that proposed approach has potential in solving complex control problems.

Keywords: Granular computing, Type-2 fuzzy logic, Fuzzy control, Genetic Algorithm.

1 Introduction

This paper focuses on the field of fuzzy logic, granular computing and also considering the control area. These areas can work together to solve various control problems, the idea is that this combination of areas would enable even more complex problem solving and better results. We explain and illustrate the proposed approach with some control problems, one is the automatic design of fuzzy systems for the longitudinal control of an airplane using genetic algorithms. This control is carried out by controlling only the elevators of the airplane. To carry out such control it is necessary to use the stick, the rate of elevation and the angle of attack. These 3 variables are the inputs to the fuzzy inference system, which is of Mamdani type, and we obtain as output the values of the elevators. For optimizing the fuzzy logic control design we use a genetic algorithm. We also illustrate the approach of fuzzy control with the benchmark case of shower control. Simulation results show the feasibility of the proposed approach of using hierarchical genetic algorithms for designing type-2 fuzzy systems.

The rest of the paper is organized as follows: In section 2 we present some basic concepts to understand this work, in section 3 we define the proposed method, section 4 describes automatic design of a fuzzy system for control of aircraft dynamic system genetic optimization, Section 5 presents a hierarchical genetic algorithm for optimal type-2 fuzzy system design, and finally conclusions are presented in section 6.

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 1–12, 2011.

2 Background and Basic Concepts

We provide in this section some basic concepts needed for this work.

2.1 Granular Computing

Granular computing is based on fuzzy logic. There are many misconceptions about fuzzy logic. To begin with, fuzzy logic is not fuzzy. Basically, fuzzy logic is a precise logic of imprecision. Fuzzy logic is inspired by two remarkable human capabilities. First, the capability to reason and make decisions in an environment of imprecision, uncertainty, incompleteness of information, and partiality of truth. And second, the capability to perform a wide variety of physical and mental tasks based on perceptions, without any measurements and any computations. The basic concepts of graduation and granulation form the core of fuzzy logic, and are the main distinguishing features of fuzzy logic. More specifically, in fuzzy logic everything is or is allowed to be graduated, i.e., be a matter of degree or, equivalently, fuzzy. Furthermore, in fuzzy logic everything is or is allowed to be granulated, with a granule being a clump of attribute values drawn together by in-distinguishability, similarity, proximity, or functionality. The concept of a generalized constraint serves to treat a granule as an object of computation. Graduated granulation, or equivalently fuzzy granulation, is a unique feature of fuzzy logic. Graduated granulation is inspired by the way in which humans deal with complexity and imprecision. The concepts of graduation, granulation, and graduated granulation play key roles in granular computing. Graduated granulation underlies the concept of a linguistic variable, i.e., a variable whose values are words rather than numbers. In retrospect, this concept, in combination with the associated concept of a fuzzy if–then rule, may be viewed as a first step toward granular computing[2][6][30][39][40].Granular Computing (GrC) is a general computation theory for effectively using granules such as subsets, neighborhoods, ordered subsets, relations (subsets of products), fuzzy sets(membership functions), variables (measurable functions), Turing machines (algorithms), and intervals to build an efficient computational model for complex with huge amounts of data, information and knowledge[3][4][6].

2.2 Type-2 Fuzzy Logic

A fuzzy system is a system that uses a collection of membership functions and rules, instead of Boolean logic, to reason about data. The rules in a fuzzy system are usually of a form similar to the following: if x is low and y is high then z = medium, where x and y are input variables (names for known data values), z is an output variable (a name for a data value to be computed), low is a membership function (fuzzy subset) defined on x, high is a membership function defined on y, and medium is a membership function defined on z. The antecedent (the rule's premise) describes to what degree the rule applies, while the conclusion (the rule's consequent) assigns a membership function to each of one or more output variables. A type-2 fuzzy system is similar to its type-1 counterpart, the major difference being that at least one of the fuzzy sets in the rule base is a Type-2 Fuzzy Set. Hence, the outputs of the inference engine are Type-2 Fuzzy Sets, and a type-reducer is needed to convert them into a Type-1 Fuzzy Set before defuzzification can be carried out. An example of a Type-2 Fuzzy Set $\tilde{X}_{mn}$ is shown in Fig. 1.

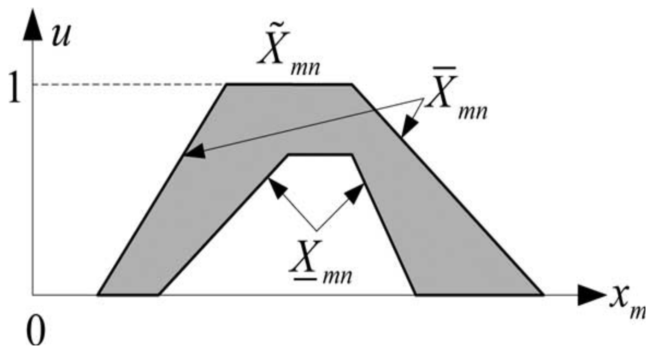

Fig. 1. Type-2 fuzzy set

Its upper membership function (UMF) is denoted $\bar{X}_{mn}$ and its lower membership function (LMF) is denoted $\underline{X}_{mn}$. A Type-2 fuzzy logic system has M inputs $\{x_m\}$ m=1,2,...,M and one output y. Assume the mth input has N_m MFs in its universe of discourse $\mathbb{X}_m$. Denote the nth MF in the mth input domain as $\tilde{X}_{mn}$. A complete rulebase with all possible combinations of the input fuzzy system consists of $K = \prod_{m=1}^{M} N_m$ rules in the form of:

$$\begin{aligned} &\tilde{R}^k\text{: IF } x_1 \text{ is } \tilde{X}_{1,n_{1k}} \text{ and ...and } x_M \text{ is } \tilde{X}_{M,n_{Mk}} \\ &[\underline{y}_k, \overline{y}_k], n_{ik} = 1, 2, \ldots, N_i, k = 1, 2, \ldots, K \end{aligned} \tag{1}$$

where $[\underline{y}_k, \overline{y}_k]$ is a constant interval, and generally, it is different for different rules.

$[\underline{y}_k, \overline{y}_k]$ represents the centroid of the consequent Type-2 Fuzzy Set of the kth rule. When $\underline{y}_k = \overline{y}_k$, this rulebase represents the simplest TSK model, where each rule consequent is represented by a crisp number. Again, this rulebase represents the most commonly used Type-2 Fuzzy Logic System in practice. When KM type-reduction and center-of-sets defuzzification are used, the output of a Type-2 Fuzzy Logic System with the aforementioned structure for an input $\boldsymbol{x}$ = (x1, x2, . . . , xM) is computed as:

$$y(\mathrm{x}) = \frac{y_l(\mathrm{x}) + y_r(\mathrm{x})}{2} \tag{2}$$

Where

$$y_l(\mathrm{x}) = \min_{\forall f_k \in [\underline{f}_k, \overline{f}_k]} \frac{\sum_{k=1}^{K} f_k \underline{y}_k}{\sum_{k=1}^{K} f_k} = \frac{\sum_{k=1}^{k_l} \overline{f}_k \underline{y}_k + \sum_{k=k_l+1}^{K} \underline{f}_k \underline{y}_k}{\sum_{k=1}^{k_l} \overline{f}_k + \sum_{k=k_l+1}^{K} \underline{f}_k} \tag{3}$$

$$y_r(\mathrm{x}) = \max_{\forall f_k \in [\underline{f}_k, \overline{f}_k]} \frac{\sum_{k=1}^{K} f_k \underline{y}_k}{\sum_{k=1}^{K} f_k} = \frac{\sum_{k=1}^{k_r} \underline{f}_k \overline{y}_k + \sum_{k=k_r+1}^{K} \overline{f}_k \overline{y}_k}{\sum_{k=1}^{k_r} \underline{f}_k + \sum_{k=k_r+1}^{K} \overline{f}_k} \tag{4}$$

in which $[\underline{y}_k, \overline{y}_k]$ is the firing interval of the kth rule, i.e.

$$\begin{aligned} \underline{f}_k &= \mu_{\underline{X}_{1,n_{1k}}}(x_1) \star \mu_{\underline{X}_{2,n_{2k}}}(x_2) \star \cdots \star \mu_{\underline{X}_{M,n_{Mk}}}(x_M) \\ \overline{f}_k &= \mu_{\overline{X}_{1,n_{1k}}}(x_1) \star \mu_{\overline{X}_{2,n_{2k}}}(x_2) \star \cdots \star \mu_{\overline{X}_{M,n_{Mk}}}(x_M). \end{aligned} \tag{5}$$

Observe that both $\underline{f}_k$ k and $\overline{f}_k$ are continuous functions when all Type-2 Membership Functions are continuous. A Type-2 Fuzzy System $\tilde{X}$ is continuous if and only if both its UMF and its LMF are continuous Type-1 Fuzzy Systems [38].

2.3 GAs

Genetic algorithms (GAs) are numerical optimization algorithms inspired by both natural selection and genetics. We can also say that the genetic algorithm is an optimization and search technique based on the principles of genetics and natural selection. A GA allows a population composed of many individuals to evolve under specified selection rules to a state that maximizes the "fitness"[15]. The method is a general one, capable of being applied to an extremely wide range of problems. The algorithms are simple to understand and the required computer code easy to write. GAs were in essence proposed by John Holland in the 1960's. His reasons for developing such algorithms went far beyond the type of problem solving with which this work is concerned. His 1975 book, Adaptation in Natural and Artificial Systems is particularly worth reading for its visionary approach. More recently others, for example De Jong, in a paper entitled Genetic Algorithms are NOT Function Optimizers , have been keen to remind us that GAs are potentially far more than just a robust method for estimating a series of unknown parameters within a model of a physical system[5]. A typical algorithm might consist of the following:

1. Start with a randomly generated population of n l–bit chromosomes (candidate solutions to a problem).
2. Calculate the fitness f(x) of each chromosome x in the population.
3. Repeat the following steps until n offspring have been created:

- Select a pair of parent chromosomes from the current population, the probability of selection being an increasing function of fitness. Selection is done "with replacement," meaning that the same chromosome can be selected more than once to be-come a parent.

- With probability P_c (the "crossover probability" or "crossover rate"), cross over the pair at a randomly chosen point (chosen with uniform probability) to form two offspring. If no crossover takes place, form two offspring that are exact copies of their respective parents. (Note that here the crossover rate is defined to be the probability that two parents will cross over in a single point. There are also "multipoint crossover" versions of the GA in which the crossover rate for a pair of parents is the number of points at which a crossover takes place.)
- Mutate the two offspring at each locus with probability P_m (the mutation probability or mutation rate), and place the resulting chromosomes in the new population. If n is odd, one new population member can be discarded at random.
- Replace the current population with the new population.
- Go to step 2 [27].

Some of the advantages of a GA include:

- Optimizes with continuous or discrete variables,
- Doesn't require derivative information,
- Simultaneously searches from a wide sampling of the cost surface,
- Deals with a large number of variables,
- Is well suited for parallel computers,
- Optimizes variables with extremely complex cost surfaces (they can jump out of a local minimum),
- Provides a list of optimal values for the variables, not just a single solution,
- Codification of the variables so that the optimization is done with the encoded variables, and
- Works with numerically generated data, experimental data, or analytical functions [13].

3 Intelligent Control of Nonlinear Dynamic Plants Using a Hierarchical Modular Approach and Type-2 Fuzzy Logic

The main goal of this work is to develop type-2 fuzzy systems for automatic control of nonlinear dynamic plants using a fuzzy granular approach and bio-inspired optimization; our work scheme is shown in Fig.2.

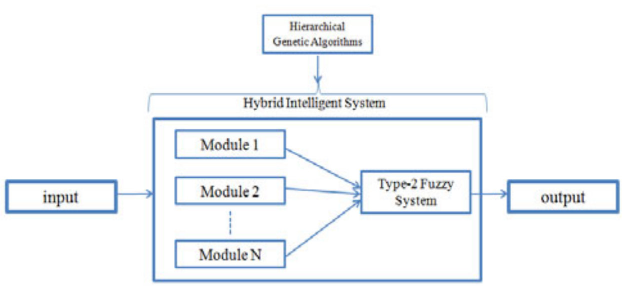

Fig. 2. Proposed modular approach for control

The use of Type-2 granular models is a contribution of this paper to improve the solution of the control problem that is going to be considered, since it divides the problem in modules for the different types of control and this model will receive the signal for further processing and perform adequate control. We can use this architecture in many cases to develop each controller separately. We can see in Fig.3 an example that how we can use this architecture in the area of control. In this example the fuzzy logic control has inputs 1 to n and outputs are also 1 to n. When we have more than one thing to control we can use type-1 fuzzy logic in each controller and then when we will have the outputs, we can then use the outputs and implement a type-2 fuzzy system to combine these outputs, and finally optimize the fuzzy system with the genetic algorithm.

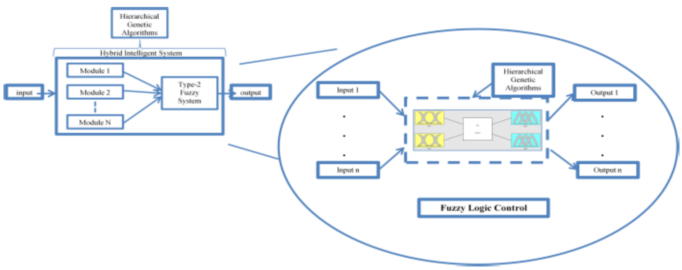

Fig. 3. Proposed granular fuzzy system

4 Automatic Design of Fuzzy Systems for Control of Aircraft Dynamic Systems with Genetic Optimization

We consider the problem of aircraft control as one case to illustrate the proposed approach. Over time the airplanes have evolved and at the same time there has been work on improving their techniques for controlling their flight and avoid accidents as much as possible. For this reason, we are considering in this paper the implementation of a system that controls the horizontal position of the aircraft. We created the fuzzy system to perform longitudinal control of the aircraft and then we used a simulation tool to test the fuzzy controller under noisy conditions. We designed the fuzzy controller with the purpose of maintaining the stability in horizontal flight of the aircraft by controlling only the movement of the elevators. We also use a genetic algorithm to optimize the fuzzy logic control design.

4.1 Problem Description

The purpose of this work was to develop an optimal fuzzy system for automatic control to maintain the aircraft in horizontal flight. The goal was to create the fuzzy system to perform longitudinal control of the aircraft and also to use a simulation tool to test the fuzzy controller with noise. The main goal was to achieve the stability in horizontal flight of the aircraft by controlling only the movement of the elevators.

4.2 PID and Fuzzy System for Longitudinal Control

If we want to use the longitudinal control we need to use 3 elements, which are: Stick: The lever of the pilot. Moving the control stick back-wards (toward the pilot) will rise the nose of the plane, and if push forward there is a lowering of the nose of the plane. Angle of attack (α).Rate of elevation (q): The speed at which an aircraft climbs. We need the above mentioned elements to perform elevator control. The comparison of the control system was carried out by first using the PID controller for longitudinal control and then we compared the results obtained with the same plant but using a fuzzy controller that was created and eventually carried out the simulation of the 2 controllers and the comparison of the results of fuzzy control with respect to PID control. The fuzzy system has 3 inputs (stick, angle of attack and rate of elevation) and 1 output (elevators). The fuzzy system that we used as a controller has 3 membership functions for each of the inputs and 3 membership functions of the output. We worked with different types of membership functions, such as the Gaussian, Bell, Trapezoidal and Triangular.

4.3 Simulation Results

In this section we present the results obtained when performing the tests using the simulation plant with the PID and Fuzzy controllers. It also presents the results obtained by optimizing the fuzzy system with a genetic algorithm. The first simulation was performed with the PID controller and we obtained the elevators behavior. We obtained an average elevator angle of 0.2967. Once the simulation results with the PID Controller were obtained, we proceeded with our Fuzzy Controller using the fuzzy system that was created previously. The simulations were carried out with different types of membership functions and the results that were obtained are shown in Table 1.

Table 1. Results for simulation plant with a type-1 fuzzy controller

Membership functions	Trapezoidal	Triangular	Gauss	Bell
Errors comparing with PID	0 .1094	0.1131	0.1425	0.1222
Comments	Fast Simulation	Less Fast Simulation	Slow simulation in comparison with previous	Slow simulation in comparison with previous

Having obtained the above results, we used a genetic algorithm to optimize the membership functions of the fuzzy system and after implementing the genetic algorithm we obtained the optimized results shown in Table 2.

Table 2. Results for the simulation plant with the fuzzy controller optimized by a Genetic Algorithm

Genetic Algorithm	**Error with respect to PID**
Using **Trapezoidal** membership functions	0.0531
Using **Gauss** membership functions	0.084236395
Using **Bell** membership functions	0.0554
Using **Triangular** membership functions	0.0261

Given the above results we can see that better results were obtained using genetic algorithms and in particular the best result was using Membership functions of triangular type. When we used the genetic algorithm the best result that we obtained was when we worked using triangular membership functions because we obtained an error of 0.0261.When we apply the Genetic Algorithm using a sine wave as a reference in our simulation plant (see Table 3) we could observe differences between the simulations. As we mentioned before we used 4 types of membership functions, such as bell, Gauss, trapezoidal and triangular. At the time of carrying out the simulation, the error was 0.004 using bell membership functions, as we can appreciate this is the better result. The decrease of error is because when we work with sine wave at the time of carrying out the simulation, our plant does not have many problems for this type of waveform and that is because the sine wave is easier to follow (higher degree of continuity).When we work using square wave we have more complex behavior because this kind of wave is more difficult. To consider a more challenging problem we decided to continue working with square wave and in this form improve our controller. We were also interested in improving the controller by adding noise to the plant. We decided to use Gaussian noise to simulate uncertainty in the control process. The Gaussian Noise Generator block generates discrete-time white Gaussian noise. Results with more noise are shown in Table 4.

Table 3. Results for simulation plant with fuzzy controller and Genetic Algorithm

Genetic Algorithm	**Error with respect to PID**
Using **Trapezoidal** membership functions	0.0491
Using **Gauss** membership functions	0.0237
Using **Triangular** membership functions	0.0426
Using **Bell** membership functions	0.004

Table 4. Results for the simulation plant with a fuzzy controller and Gaussian noise (Type-2 and Type-1)

Membership functions	**Noise Level**					
	84	123	580	1200	2500	5000
Triangular	0.1218	0.1191	0.1228	0.1201	0.1261	0.1511
Trapezoidal	0.1182	0.1171	0.1156	0.1196	0.1268	0.1415
Gauss	0.1374	0.1332	0.1356	0.1373	0.1365	0.1563
Bell	0.119	0.1172	0.1171	0.1203	0.1195	0.1498
Type-2 Triangular	0.1623	0.1614	0.1716	0.163	0.1561	0.1115

In this case a type-2 fuzzy system (last row) produces a better result when the noise level is high. In Table 4 we can observe that in many cases the type-1 provided better results than type-2. But when we raise the noise level the type-2 fuzzy system obtained better results as it supports higher levels of noise.

5 Hierarchical Genetic Algorithm for Optimal Type-2 Fuzzy System Design in the Shower Control

In this case we propose an algorithm to optimize a fuzzy system to control the Temperature in the Shower benchmark problem, in this application the fuzzy controller has two inputs: the water temperature and the flow rate. The controller uses these inputs to set the position of the hot and cold valves. In this part the genetic algorithm optimized the fuzzy system for control.

5.1 Problem Description

The problem was of developing a genetic algorithm to optimize the parameters of a fuzzy system that can be applied in the fuzzy logic areas. The main goal was to achieve the best result in each application, in our case fuzzy control of the shower. We started to work with different membership functions in these cases and after performing the tests finally we took the best result. The genetic algorithm can change the number of inputs and outputs depending on that we need it. The Chromosome for this case is shown in fig.4.

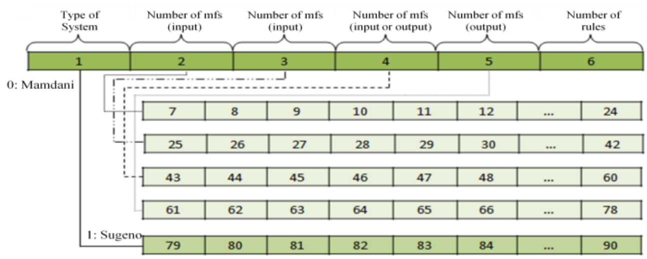

Fig. 4. Chromosome of the Genetic Algorithm

5.2 Fuzzy Control

In this case we realized the simulation with the Simulink plant in the Matlab programming language. The problem was to improve temperature control in a shower example the original fuzzy system has two inputs to the fuzzy controller: the water temperature and the flow rate. The controller uses these inputs to set the position of the hot and cold valves. When we simulated the type-2 fuzzy system the best result that we obtained was 0.000096, and in the same problem but using type-1 we obtained 0.05055. This shows that type-2 fuzzy control can outperform type-1 for this problem. The best fuzzy system that we obtained in fuzzy control is shown in Figure 5.

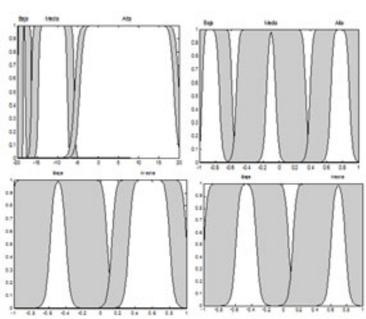

Fig. 5. Type-2 Fuzzy system for control

6 Conclusions

We use two benchmark problems and based on the obtained results we can say that to achieve control of the present problems, type-2 fuzzy logic is a good alternative to achieve good control. When we worked with a type-1 fuzzy system we obtained good results, but if we want to work with noise the previous good results will not be so good, in this case we need to work with type-2 and with this we obtained better results and also using a genetic algorithm to optimize the fuzzy system. When we have a problem for example to control the flight of an airplane we need to control 3 different controllers. In this case the fuzzy granular method is of great importance, because we want to control the flight of an airplane completely. We want to use a type-1 fuzzy system in each controller and then use a type-2 fuzzy system to combine the outputs of the type-1 fuzzy systems and implement the concept of granularity and with this method we hope to obtain a better result in this problem.

References

1. Abusleme, H., Angel, C.: Fuzzy control of an unmanned flying vehicle, Ph.d. Thesis, Ponti-ficia University of Chile (2000)
2. Bargiela, A., Wu, P.: Granular Computing: An Introduction. Kluwer Academic Publish, Dordercht (2003)
3. Bargiela, A., Wu, P.: The roots of Granular Computing. In: GrC 2006, pp. 806–809. IEEE (2006)
4. Blakelock, J.: Automatic Control of Aircraft and Missiles. Prentice-Hall (1965)
5. Coley, A.: An Introduction to Genetic Algorithms for Scientists and Engineers. World Scientific (1999)
6. The 2011 IEEE Internaional Confenrece on Granular Computing Sapporo, GrC 2011, Japan, August 11-13. IEEE Computer Society (2011)
7. Dorf, R.: Modern Control Systems. Addison-Wesley Pub. Co. (1997)
8. Dwinnell, J.: Principles of Aerodynamics. McGraw-Hill Company (1929)
9. Engelen, H., Babuska, R.: Fuzzy logic based full-envelope autonomous flight control for an atmospheric re-entry spacecraft. Control Engineering Practice Journal 11(1), 11–25 (2003)
10. Federal Aviation Administration, Airplane Flying Handbook, U.S. Department of Transportation Federal Aviation Administration (2007)
11. Federal Aviation Administration. Pilot’s Handbook of Aeronautical Knowledge, U.S. Department of Transportation Federal Aviation Administration (2008)
12. Gardner A.: U.S Warplanes The F-14 Tomcat, The Rosen Publishing Group (2003)
13. Gibbens, P., Boyle, D.: Introductory Flight Mechanics and Performance. University of Sydney, Australia. Paper (1999)
14. Goedel, K.: The Consistency of the Axiom of Choice and of the Generalized Continuum Hypothesis with the Axioms of Set Theory. Princeton University Press, Princeton (1940)
15. Haupt R., Haupt S.: Practical Genetic Algorithm. Wiley-Interscience (2004)
16. Holmes, T.: US Navy F-14 Tomcat Units of Operation Iraqi Freedom, Osprey Publishing Limited (2005)
17. Jamshidi, M., Vadiee, N., Ross, T.: Fuzzy Logic and Control: Software and Hardware Applications, vol. 2. Prentice-Hall, University of New Mexico (1993)

18. Kadmiry, B., Driankov, D.: A fuzzy flight controller combining linguistic and model based fuzzy control. Fuzzy Sets and Systems Journal 146(3), 313–347 (2004)
19. Karnik, N., Mendel, J.: Centroid of a type-2 fuzzy set. Information Sciences 132, 195–220 (2001)
20. Keviczky, T., Balas, G.: Receding horizon control of an F-16 aircraft: A comparative study. Control Engineering Practice Journal 14(9), 1023–1033 (2006)
21. Liu, M., Naadimuthu, G., Lee, E.S.: Trayectory tracking in aircraft landing operations management using the adaptive neural fuzzy inference system. Computers & Mathematics with Applications Journal 56(5), 1322–1327 (2008)
22. McLean D.: Automatic Flight Control System. Prentice Hall (1990)
23. McRuer, D., Ashkenas, I., Graham, D.: Aircraft Dynamics and Automatic Control. Princeton University Press (1973)
24. Melin, P., Castillo, O.: Intelligent control of aircraft dynamic systems with a new hybrid neuro- fuzzy–fractal Approach. Journal Information Sciences 142(1) (May 2002)
25. Melin, P., Castillo, O.: Adaptive intelligent control of aircraft systems with a hybrid approach combining neural networks, fuzzy logic and fractal theory. Journal of Applied Soft computing 3(4) (December 2003)
26. Mendel, J.: Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Prentice-Hall, Upper Saddle River (2001)
27. Mitchell, M.: An Introduction to Genetic Algorithms. Massachusetts Institute of Technology (1999)
28. Morelli, E.A.: Global Nonlinear Parametric Modeling with Application to F-16 Aerodynamics, NASA Langley Research Center, Hampton, Virginia (1995)
29. Nelson, R.: Flight Stability and automatic control, 2nd edn. Department of Aerospace and Mechanical Engineering, University of Notre Dame., McGraw Hill (1998)
30. Pedrycz, W., Skowron, A., et al.: Handbook granular computing. Wiley Interscience, New York (2008)
31. Rachman, E., Jaam, J., Hasnah, A.: Non-linear simulation of controller for longitudinal control augmentation system of F-16 using numerical approach. Information Sciences Journal 164(1-4), 47–60 (2004)
32. Reiner, J., Balas, G., Garrard, W.: Flight control design using robust dynamic inversion and time- scale separation. Automatic Journal 32(11), 1493–1504 (1996)
33. Sanchez, E., Becerra, H., Velez, C.: Combining fuzzy, PID and regulation control for an autonomous mini-helicopter. Journal of Information Sciences 177(10), 1999–2022 (2007)
34. Sefer, K., Omer, C., Okyay, K.: Adaptive neuro-fuzzy inference system based autonomous flight control of unmanned air vehicles. Expert Systems with Applications Journal 37(2), 1229–1234 (2010)
35. Song, Y., Wang, H.: Design of Flight Control System for a Small Unmanned Tilt Rotor Air-craft. Chinese Journal of Aeronautics 22(3), 250–256 (2009)
36. Walker, D.J.: Multivariable control of the longitudinal and lateral dynamics of a fly by wire helicopter. Control Engineering Practice 11(7), 781–795 (2003)
37. Wu, D.: A Brief Tutorial on Interval Type-2 Fuzzy Sets and Systems (July 22, 2010)
38. Wu, D., Jerry, M.: On the Continuity of Type-1 and Interval Type-2 Fuzzy Logic Systems. IEEE T. Fuzzy Systems 19(1), 179–192 (2011)
39. Zadeh, L.A.: Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems. Soft Comput. 2, 23–25 (1998)
40. Zadeh, L.A.: Outline of a new approach to the analysis of complex systems and decision processes. IEEE Trans. Syst. Man Cybern. SMC-3, 28–44 (1973)