

A PLC-based Fuzzy Logic Control with Metaheuristic Tuning

Jeydson SILVA*, Davidson MARQUES, Ronaldo AQUINO, Otoni NÓBREGA

DEE, Federal University of Pernambuco, Recife, Pernambuco, Brazil, 50740-530

jeydsonl@gmail.com (*Corresponding author); davidson_cm@hotmail.com;

rrbaquino@gmail.com; otoninobrega@gmail.com

Abstract: In recent times, artificial intelligence applications in the field of dynamic systems control have proved to be an efficient tool for the improvement of processes. Programmable Logic Controller-based control systems are found in the vast majority of industrial automation systems. In this paper, a fuzzy controller approach is proposed to enhance the design of industrial control systems using commercial PLCs. The fuzzy controller is tuned by artificial intelligent techniques, and the performances of different fuzzy controller-tuning methods are compared, including the metaheuristic ones. The results shown in this paper validate the proposal that metaheuristics tuning methods of fuzzy controller have better performance in intelligent automation.

Keywords: Artificial intelligence, Control systems, Industrial automation, PLC, Fuzzy, Metaheuristics.

1. Introduction

In modern automation systems, Programmable Logic Controller (PLC) is an essential component. The industrial application of PLC occurs virtually in all the types of industrial systems. With the continuous improvement of the technology in the industrial area, the PLC function has been greatly improved; therefore, PLC applications continue to grow (Dong & Yang-Ming, 2017; Mahato, Maity & Antony, 2015). Some models of PLCs have implemented Fuzzy Logic (FL) tools that facilitate the deployment of fuzzy controllers, being possible to configure the fuzzy system, defining its Member Functions (MFs) for the input and output variables, as well as their respective operational ranges and also its rule base. Such tools generally use the centroid method for their output (Altas & Sharaf, 2007).

The fuzzy control is based on the fuzzy logic, which unlike logical systems is much closer to natural language as well as to human thinking. Such an approach allows one to approximate the inexact settings of the real world nature (Takagi & Sugeno, 1985). Founded on this fuzzy approach, there are so-called fuzzy logic controllers (FLCs) that based on expert knowledge into an automatic control strategy can provide algorithms which in their turn, can convert the linguistic control strategy. Previous research work has shown that the application of FLC often exceeds the efficiency of conventional control algorithms. In dealing mainly with more complex processes for conventional quantitative techniques, or when the process information is uncertain, like most situations of nature, such as renewable energy applications, the FLC proved to be quite useful (Zhou et al., 2014).

Although diffuse rules are somewhat easy to develop by human specialists, diffuse membership functions (MFs) are difficult to obtain. In this context, the adjustment of MFs is a time-consuming exercise and often does not reach a good result. In order to overcome such difficulties, some techniques have been reported to automate this process of tuning MFs (Wang, Xia, & Zhou, 2009).

Metaheuristics such as Genetic Algorithm (GA), have been widely used in automatic designs of fuzzy controllers, especially in the areas of mobile robotics. In various circumstances, the GA was used for the purpose of adjusting both the fuzzy MF and the fuzzy rule bases (Zhang, Zhang, & Wang, 2009). Another technique used is the particle swarm optimization (PSO) whose inspiration comes from the collective natural behaviour of the animals. Because of its simplicity in implementation, the PSO is gaining popularity over time in engineering applications such as system modelling and image processing (Fang, Kwok, & Ha, 2008).

In this work, the GA and PSO metaheuristics are used to automatically adjust the MFs of a Mamadani-type fuzzy controller when performing the control of PLC on a real industrial fan system. A real experiment was accomplished in a didactic industrial plant of a fan system, which is controlled by a programmable logic controller (PLC), located at the Energy Efficiency and Power Quality Laboratory of the Federal University of Pernambuco (Aquino et. al, 2016). The results of these tunings were then compared with a manual

tuning. PLC-based on FLC tuned by PSO and GA. The main contribution of this work refers to the use of PSO and Ag metaheuristics in the adjustment of a fuzzy controller implemented directly in PLC, written in Ladder programming language. This system is being used to control some industrial plant models, obtaining satisfactory practical results compared to the PI control.

This paper is organized as follows: Section 2 details the LAMOTRIZ industrial fan system used in the experiments. In section 3, the process of surveying the dynamic information of the plant and the acquisition of its discrete mathematical model, used in the work simulations, are briefly discussed. In section 4, the characteristics of the fuzzy control system are discussed. This section also deals with the PSO and GA methods and how these can be used in the MF tuning process distorted configuration. The results and simulation control show the effectiveness of the proposed MF tuning method. In addition, the design of the FLC control in the PLC and the actual used supervision system are illustrated. The results are discussed in section 5, which provides the comparisons between the proposed FLC controller and a manual adjustment.

2. Industrial Fan System

All experiments described in this paper were performed in the Energy Efficiency and Power Quality Laboratory of the Federal University of Pernambuco. In this laboratory, there are several industrial systems, including an industrial fan system. This system uses Siemens's WinCC Flexible® supervision software as the main user interface. Through this interface, it's possible to perform a complete system operation. Figure 1 displays a picture of this industrial fan system.

A description of the fan system used in this study can be observed below:

- Motor-Fan group
- Inflow regulator (Damper)
- Inflow transmitter
- Pressure transmitter
- Air admission
- Air output
- Temperature sensor of the air input
- Temperature sensor of the air output
- Command and control panel of the fan system (PLC, frequency inverter, measurer, command key, etc.)

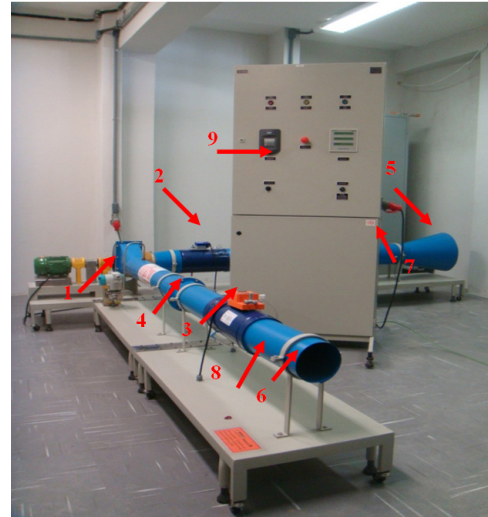


Figure 1. Prototype of an industrial fan system

3. System Modelling

3.1 System Identification

The used model approximates the real process, which cannot fully incorporate all the characteristics of the real physical system. There must be a compromise between the cost of having the model and the level of detail of it regarding the benefits expected by its application (Araydah, Tutunji & Al-Naimi, 2017).

The identification of the plant model and its transfer function, were carried out experimentally. From a set of experiments on the physical fan system, it was possible to obtain the input-output pairs of the open-loop system. The analysis performed in this work focused on the airflow control through the frequency inverter, from which the process input signal is a frequency value in the motor, regulated by the frequency inverter. The supervisory system is responsible for providing this frequency value. The system output signal is represented by the airflow in the fan system outlet duct. In this way, according to this dynamic response characteristic of the system, it's possible to approximate the physical system by using a mathematical dynamic system of first order with delay. In this system, the mathematical model has the following characteristics (Dorf & Bishop, 2010; Ogata, 2009).

$$G(s) = \frac{K_P e^{-\theta s}}{\tau s + 1} \quad (1)$$

3.2 Modelling of the Fan System

Through the step response of the plant, a method known to be useful to approximate the models of dynamic systems, it was possible to obtain the dynamic fan system's model present in LAMOTRIZ. The experiment for identifying the system was possible by, configuring the plant in the open-loop mode through the supervisory system. In this work, a PLC is used to send the control signals to the fan system. The sensors of the plant, which are in communication with the supervisory system, collect the data from the experiments. From the collection of these output values of the experiment, it's possible to create output airflow graphs with respect the time of the airflow control process.

By observing the response of the open-loop system, the industrial fan system's transfer function is given according to Equation (2) below:

$$G(s) = \frac{22,46e^{-4s}}{5,5s+1} \quad (2)$$

3.3 Digital Modelling

An important aspect that must be considered while tuning the controllers is how the techniques of controllers are designed. Most of these techniques are designed in continuous time and it could affect significantly the behaviour of the controller's tuning (Mekki et al., 2017). In real application of control systems, the digital format technique is more appropriated to be applied because it encompasses the characteristics of discrete signals, such as quantization and discretization, which are present in the digital control techniques of the systems.

The Zero-Order Hold (ZOH) technique was used to obtain an approximate correspondence between continuous and discrete systems. This could transform the transfer function from continuous time into discrete time with a good precision and with no information lost. This methodology is crucial because the sample rates influence the actual process dynamics (Du, Ying & Lin, 2009; Leondes, 1996). By using the ZOH technique, the discrete-time transfer function was obtained with a time sampling of 1 second. It has been given the following equation:

$$G(z) = \frac{3,773}{z-0,338} \quad (3)$$

4. Methodology

In this work, the entire methodology will be used in a prototype of a fan system; however, the application of this concept can be extended for most industrial control processes.

4.1 Fuzzy System

The fuzzy system is an important artificial intelligence technique. This system has been used widely in various control tasks, as its application in the control processes is necessary because of the nonlinearities, inefficient measurements and a large noise level that characterize the industrial processes (Zadeh, 1994). Fuzzy logic is a decision-making system and it helps to find the output for the given set of input variables. The strength of fuzzy logic stems from its ability to infer conclusions and generate responses based on vague, ambiguous, incomplete, or imprecise information (Dong & Yang-Ming, 2017).

In that respect, fuzzy-based systems have the ability to reason in a similar way to humans.

In addition, its direct architecture doesn't need precise mathematical models. This system is based on the empirical knowledge of the operator with which it simulates human reasoning to solve several types of problems. Therefore, the concepts of fuzzy logic can be used; this strategy can be represented by a set of rules. In some cases, the operator develops a good control strategy only with the empirical knowledge.

The control scheme was based on TSK (Takagi-Sugeno-Kang) controller, because it allows the concatenation of different controllers designed through different strategies and gathered in the same set of rules (Lam et al., 2016).

4.2 Fuzzy Controller Design

The Fuzzy logic controllers (FLC) are based on the concepts of fuzzy implication and the compositional rules of inference by which the control action can be obtained from a knowledge set described linguistically and given by experts or experience operators. The design of a fuzzy controller requires more decisions than that of other controllers; this type of control can be described simply as control with sentences rather than equations (Liu, Chen & Tsao, 2001).

Usually the expert knowledge is in the form of fuzzy control rules. One of the most complicated processes in fuzzy controller design is the organization of a fuzzy rule base (Pitalúa-Díaz, Lagunas-Jiménez & González, 2013). Recent works demonstrate the versatility of fuzzy in several applications, such as a robust model-free adaptive FLC control free of robust model based on the fractional order and extended observer for half-car suspension system (Mustafa, Wang & Tian, 2019).

This paper demonstrates the use of algorithms based on metaheuristics to promote simple tunings of these diffuse rules of the FLC. A fuzzy controller includes empirical rules, and that is especially useful for the operator controlled plants. In the process of building the rule base, the airflow error and error's change are partitioned into five fuzzy sets labelled as {NB (Negative Big), NS (Negative Small), Z (Zero), PS (Positive Small), PB (Positive Big)}, totaling 25 rules.

Take for instance a typical fuzzy controller:

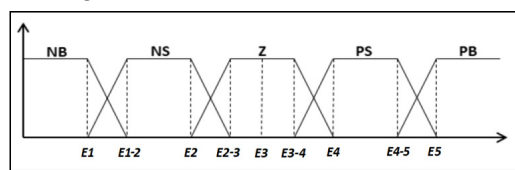
IF (x is **A1**) AND (y is **B1**) THEN (z is **C1**)

where, A1 is one of the fuzzy set X (i.e. airflow error), B1 is one of the fuzzy set Y (i.e. error's change of airflow) and C1 is one of the fuzzy set Z (i.e. output gain of controller).

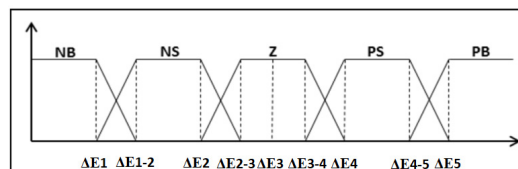
In this work, the FLC was modelled with trapezoidal membership functions (MFs) applied to the inputs, as showing in Figure 2. The degree of membership is calculated in PLC and is based on the ladder diagram, using a basic instruction set. The metaheuristics that will be used will find values for these MFs, which will be better adapted to the dynamics of the FLC operation, based on an optimized evaluation equation.

For the error and error's change MFs, their positioning parameters were established, by defining the position of the fuzzy sets (NB, NS, Z, PS, PB), which represent the optimization variables of the FLC tuning problem. A total of 20 variables were established, 8 for the error, 8 for the error's change and 4 for the output gain. In this methodology, the trapezoidal functions used the information of the neighbouring functions to complete their positioning, thus reducing considerably the number of problem variables and allowing to use less computational memory to perform the optimization. In addition, this type of member function allowed greater stability of

the present PLC. The limits of MFs obey the following characteristic rules:



(a) Error membership function



(b) Error's change membership function

Figure 2. Trapezoidal membership functions

For the input's values, error (E) and error's change (ΔE), the antecedents of a diffuse rule are known as the degrees of association, obtained during fuzzification: $\mu_{A1}(E)$ and $\mu_{B1}(\Delta E)$. From these values, the consequence of this diffuse rule is calculated according to the following equation:

$$\mu_1(z_1) = \min [\mu_{A1}(x); \mu_{B1}(y)] \tag{4}$$

The singleton form in the output membership function was chosen due mainly to its simplicity in solving the crisp output during the defuzzification. For the defuzzification process in this work, the Centroid method was used. This method could provide superior control performance, and some previous researchers reported this aspect (Aisbett, & Rickard, 2013).

$$CG = \frac{\sum_i \mu_i(z_i)z_i}{\sum_i \mu_i(z_i)} \tag{5}$$

As in the case of the input MFs, these MFs output values will also be used as variables in the metaheuristic optimization for the tuning of FLC. However, the outputs of the MFs have only 4 variables for the optimization of the problem (see Figure 3), unlike the inputs, due to the fact that the MFs are singleton. A variable is thus defined for each position of the fuzzy set.

The limits of MFs obey the following characteristic rules:



Figure 3. Output Membership function

In fact, there are about 20 variables in this fuzzy tuning problem. This is not a trivial problem which could have been solved by using the traditional optimization methods, so the use of metaheuristics was proposed in order to find an optimal solution for the fuzzy sets of the FLC.

In this work the fuzzy rules were predefined by means of the operator tests and did not undergo any kind of changes in the course of the optimization. There are some cases which use the change of the positions of the rules in order to obtain a better performance of the control method, but this situation does not apply for the present approach.

The fuzzy rules are shown in Table 1:

Table 1. Fuzzy set rules

$E \setminus \Delta E$	NB	NS	Z	PS	PB
NB	NB	NB	NS	NS	PB
NS	NB	NS	NS	PS	PS
Z	NB	NS	Z	PS	PB
PS	NS	NS	PS	PS	PB
PB	NB	PS	PS	PB	PB

The final objective of that problem is to provide stability for the signal of control in case that unexpected changes might happen in the reference airflow. To perform the appropriate control frequency adjustment for the motor to return to the reference flow, error signals and input error change are used for FLC.

4.3 Evaluation Function

To perform the optimization, it's necessary to define a fitness function which will allow measuring the quality of the dynamic behaviour of the controller. The performance requirements function as constraints on the controller setting, imposing limits on the behaviour of the control system (Ogata, 2009).

The fitness function of this work is composed of the following performance indices: IAE (integral of absolute value error), usually a performance parameter for PID controller tuning, Overshoot (M_p), Rise time (t_r) and Settling Time (t_s).

In this work, the fitness function is defined as follows:

$$J = \int_0^{\infty} |e(t)| dt + 80000 * M_p + 1000 * t_r + 1000 * t_s \quad (6)$$

The weights of Equation 6 were obtained empirically through the various test results in simulations of the system plant, which is represented as follows (see Figure 4):

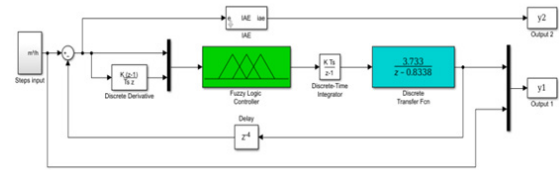


Figure 4. LAMOTRIZ fan system control in Simulink

4.4 Genetic Algorithm

Genetic algorithms (GA) are population-based optimization algorithms. They are specific for the evolution of the animal species. For a long time, the Genetic algorithms have been one of the most important evolutionary computing techniques; this metaheuristic has a conceptual basis in the evolution of individual structures (Goldberg, 1989).

This technique searches for the best possible solutions to the problem, using a strategy of individual survival by privileging those who have a greater aptitude for survival. In general, all individuals have a certain degree of aptitude for survival, but the fittest of the population tend to reproduce and survive the next generation, thereby improving future generations. In this method, a relatively good initial estimate of the problem variables is not necessary for the optimization process to have a good evolutionary behaviour (Goldberg, 1989). The accuracy of the individual of each population for survival is measured by means of a criterion chosen according to the problem of the evaluation optimization function.

In genetic science there is a term called *genotype*, which represents the set of genes that define the genetic makeup of the individual; such genes will go through the processes of the genetic operators of the algorithm. In most problems, this genotypic representation is given by a binary vector, where each parameter of this vector represents the presence of an important element of the individual. These characteristics combine to form the individual's phenotype, its detectable characteristics.

Because of the incomplete nature of GA theory, most of the knowledge about the successful implementation of GAs comes from experiment

and experience. The genetic algorithm cannot be considered a random algorithm to search for solutions, even using metaheuristic and probabilistic methods in its iterative chain. The exploration of the available information occurs in an intelligent way, seeking solutions capable of further improving the performance criterion (Kumar, Sharma, & Gautam, 2016; Pelikan, & Goldberg, 2001).

The GAs commonly use algorithms to solve various types of problems, such as linear and nonlinear systems. These algorithms exploring the regions of the search space in depth, discovering better areas that can contain the solution of the problem, all this mainly through their genetic operators' mutation, crossover and selection. (Pelikan, & Goldberg, 2001; Tsutsui & Ghosh, 1997).

As for the most of the animals, the selection process in GAs aims to represent the process of reproduction and natural selection. At this point, individuals are chosen for reproduction. This selection allows prioritizing those individuals who have a better fitness value; they, in turn, will contribute with their genetic material to the next generation. Different types of selection can be used to choose the fittest individuals, some methods may be more effective than others depending on the type of problem involved. Please start a new sentence: "The tournament method, in which individuals compete with each other to remain in the population, is among the most commonly used. After the selection of individuals of the population, the remaining population undergoes the crossing mechanism (Goldberg, 1989).

The main purpose of the crossover mechanism is to perform the mixing of the genetic material among the individuals chosen through the selection stage. This makes it possible to explore regions close to previous individuals, reducing the possibility of local minimum points. A chromosome position is chosen randomly from the crossover factors previously defined by the operator, these values will then be exchanged between some other individuals in order to provide a different genetic combination of both individuals involved in the crossover (Goldberg, 1989).

In order to provide diversity between individuals, the mutation process in the population is applied after the crossing mechanism, which is equivalent

to a random search of the viable solution region. The procedure consists in altering a part of the individual's chromosome by making it different; such a process is based on a mutation rate previously defined by the operator. This process provides population diversity because it randomly changes some genes in the chromosome, providing the means for introducing new elements into the population (Goldberg, 1989). This paper uses the uniform mutation, the single crossing and three different selection methods (the Stochastic sampling, the Roulette selection and the Tournament selection) as main parameters.

4.5 Particle Swarm Optimization

Kennedy and Eberhart initially proposed a particle swarm optimization (PSO) which provided the initial proposal provided a model to solve nonlinear functions. This method is considered a population algorithm with metaheuristic characteristics as in the case of GA. The method is based on the collective behaviour of the animals, in their search for places that provide a quantity of food for the group. One of the advantages of this method is the high capacity of this algorithm for fast convergence and simplicity of execution, when compared to the great majority of heuristic optimization methods (Dorigo, Maniezzo & Colomi, 1996; Kennedy, Eberhart & Shi, 2001; Olivas, Valdez, Castillo & Melin, 2016).

The PSO works based on population of individuals as in GA. The individuals are called "particles" and the set of these particles is called "swarm". Each particle that makes up the swarm has been associated with velocity called v and a positions called x . These particles move within the feasible search space looking for the possibly optimal solution of the problem. In this way, the search of the swarm individuals for a place with better food condition for the group is stimulated. (Dorigo, Maniezzo & Colomi, 1996; Kennedy, Eberhart & Shi, 2001; Olivas, Valdez, Castillo & Melin, 2016).

At the time of the movement of the particles in the search space, they will have their best positions stored in a vector, known as $pbest$ (position best). The best overall position of the particles in the examination is known as $gbest$ (global best). This position assessment is based on an evaluation function, which is modelled by the operator, founded on the prior knowledge of the problem and the interest of the desired solution.

$$v_{id}^{k+1} = w * v_{id}^k + c_1 * r_1 * (pbest_{id} - x_{id}^k) + c_2 * r_2 * (gbest_{id} - x_{id}^k) \quad (7)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \quad (8)$$

where i represents the i th particle in the system and d represents d th position of the swarm. The velocity of particles is influenced by its own experience, known as cognitive factor (c_1), and by its swarm companions, known as social factors (c_2). The constants c_1 and c_2 are positive, controlling the velocity and the displacement of the particle in the search space, providing an individual and a social search characteristic for each particle, respectively. The tendency should be to attract the particles towards the optimum or near the optimal solution of the problem by weighting the best values of the individual ($pbest$) and group ($gbest$) positions (Liang & Kang, 2016; Meng et al., 2016).

The terms x_{id}^k and v_{id}^k represent the position and velocity of the particle in the i -th position of the d -th place of the swarm dimension. The constants terms r_1 and r_2 must be random numbers of the interval $[0,1]$, contributing to a diversified search of possible solutions.

The term is considered as an inertial velocity weight and represents the influence of the actual speed on the speed of the next iteration of the particle. Higher values of w improve global search in new regions; otherwise, low values contribute to local search in a more promising region. A correct choice of that parameter improves or worsens the balance of the algorithm between the global and the local exploration skills. The velocity limits are given by $-v_{id}^{max} \leq v_{id}^k \leq v_{id}^{max}$, so that the velocity of the particle is not very disproportionate with respect to its convergence.

In order to maintain certain equilibrium in the cognitive and social learning of the particle, the constants c_1 and c_2 were set at 2, as shown in previous studies (Kennedy, Eberhart & Shi, 2001; Clerc & Kennedy, 2002).

For a good convergence of this type of problem the values of w must be around 0.6. Such value was obtained empirically through studies and simulations of this problem. Although a relatively recent finding, when compared to other types of metaheuristics, the PSO based on the social and cognitive behavior of the flocks of animals has shown to be quite promising, because its great

simplicity allows it to be applied in order to solve diverse types of problems (Tanweer, Suresh & Sundararajan, 2015; Wang, Wang & Wu, 2013).

4.6 PLC and Supervisory Control System

For the design of the FLC in the PLC, it is necessary to write the mathematical rules of fuzzy logic in the ladder programming language. From the fuzzy rules of Table 1 and from the centroid method of the trapezoidal MFs, it is possible to write the Functions Blocks (FB) in the ladder language of the PLC. The example of rule 21, its code in Matlab® and its FB written in the ladder language are shown in Figure 5:

```

% FR21: If E = PB AND dE is NB THEN output = yNB;

if (E > E4~5)

    if (E <= E miEPB = (E - E4~5)/(E5 - E4~5);

    else
        miEPB = 1;
    end
    if (dE < dE12)
        if (dE <= dE1)
            midENB = 1;
        else
            midENB = (dE12 - dE)/(dE12 - dE1);
        end
        rule21 = min(miEPB, midENB);
        mis = [mis mi21];
        yout=yNB;
    end
    
```

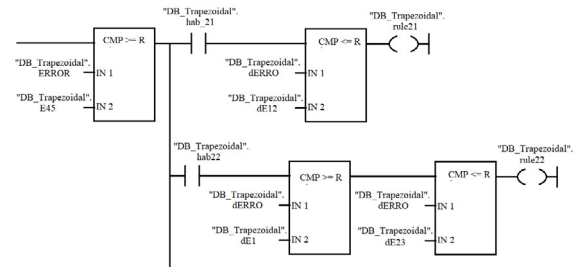


Figure 5. Ladder diagrams of the rule 21 on PLC

The fan system in this work is operated through a supervisory system, which is the main key to controlling the plant’s variables. The software responsible for this task is WinCC® Flexible that allows the complete operation of the system, by using a valve opening in order to adjust the motor speed, taking into account the system variables such as airflow, pressure, air temperature, rotation and torque of the motor, electrical variables, etc.

All the system’s components are modelled in the WinCC® computer program including valves, sensors, motor-fan group and others. The

supervisory software communicates with the control system via INDUSTRIAL-ETHERNET network and the interface between the PLC and the measuring and control devices communicates via PROFIBUS-DP network.

Figure 6 shows the main screen illustrated through the supervisory software WinCC®, which represents the industrial fan system, used in this work.

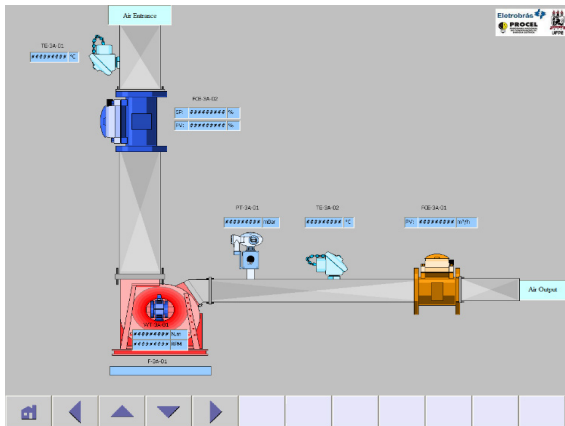


Figure 6. Main screen of industrial fan system on supervisory system

As can be seen in Figure 6 all the items of equipment are represented in the supervisory system by the exhibited tags. The tag F-3A-01 represents the fan, TE-3A-01 and TE-3A-02 represent the thermostats which show the air temperature on the air entrance and air output, FCE-3A-02 represents the damper which allow the airflow control, FT-3A-01 represents the speed transducer, WT-3A-01 represents the torque and rotation sensor and PT-3A-01 represents the pressure transducer.

The interface of the supervisory system with the trapezoidal inference functions of the FLC logic structure is presented in Figure 7.

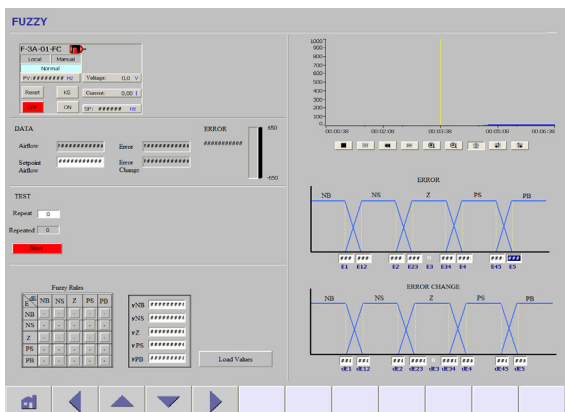


Figure 7. Screen of the Membership functions of FLC on the supervisory system

The PLC used in this work is the S7-300 produced by Siemens and is integrated with the CPU 313C-2 DP. The PLC was programmed through the Step-7 programming software, by using the ladder programming language.

5. Results

Due to the difficulty of this problem, metaheuristic optimization algorithms (PSO/GA) have been used to determine the variables that allow better operating conditions than the FLC manual adjustment for real systems. The total is represented by 20 variables that must be found in this problem (inputs error, change of inputs error and outputs gain). As there are no methods defined to find the solution to this type of problem, metaheuristics are quite appropriate.

The global structure of the FLC, implemented on Simulink/MATLAB® and on PLC, is shown in Figure 8.

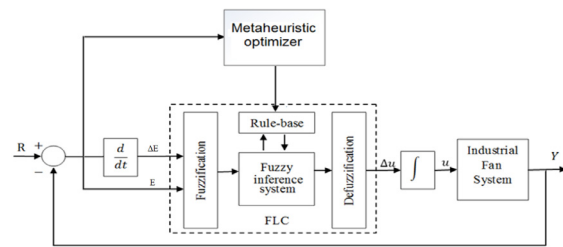
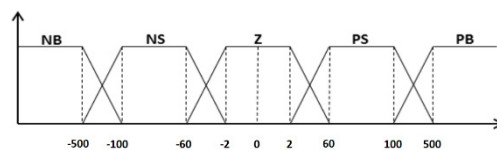


Figure 8. FLC control associated with metaheuristic

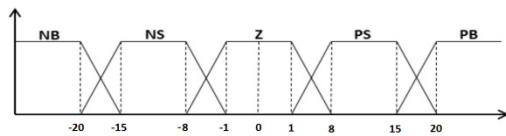
where R is the reference signal value, Y is the output value of the process, E the error, ΔE is the error's change and u is the control signal provided by the FLC.

The manual adjustment of the parameters of the input MFs was obtained empirically through the knowledge of the operator. The trial and error method was used and, therefore, an initial control point was obtained for the FLC. The positions are out of scale for better visualization of MFs.

The manual setting of the parameters of MFs of the inputs is shown below (see Figure 9).



(a) Error membership function



(b) Error's change membership function

Figure 9. Input membership functions

In addition to the input values, the initial values of the gains in the outputs were also obtained by trial and error according to the previous knowledge of the operator. The output MFs is shown in Figure 10.



Figure 10. Output membership function

In order to minimize the evaluation function J for the tuning of the FLC parameters, both PSO and GA metaheuristics were used; the results of these metaheuristic methods were compared in order to corroborate the outcomes of the optimization process. In addition, the FLC adjustments were analysed both quantitatively and qualitatively.

In the previous experiments, the search space of each variable of the problem was delimited, reducing the optimization time of the algorithm and avoiding local minimums points. Next, the PSO's behaviour is presented with a swarm containing 100 particles for each of the 20 variables.

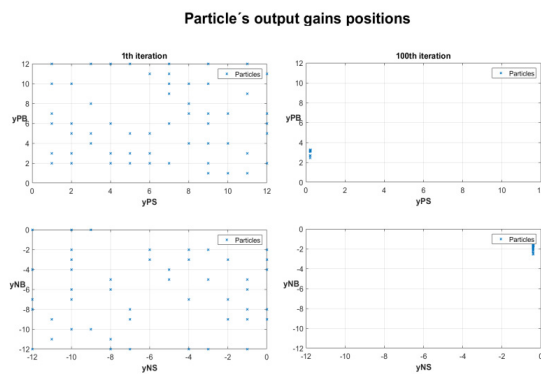


Figure 11. Particle's position of the gain in the PSO

During the first iteration, it can be noticed that the particles are dispersed within the boundary of the research, at the end of the iterations; most of the particles are concentrated around the minimum solution point of the optimization problem, demonstrating the convergence characteristic of the particles to the same point at the end. Figure

12 and Figure 13 present the values of the FLC parameters found by the PSO with 50 particles in 100 iterations. These values (particles) were plotted in pairs in order to give an idea of their linear decoupling.

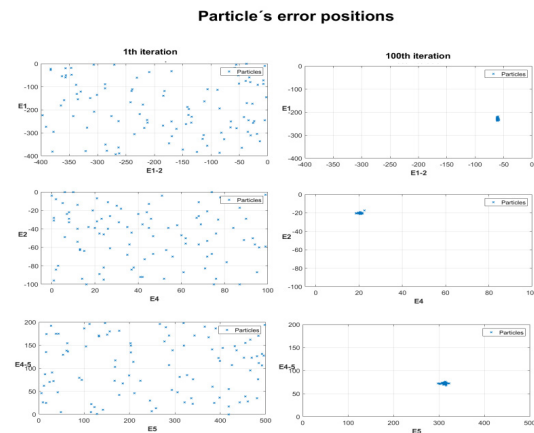


Figure 12. Particle's position of the error in the PSO

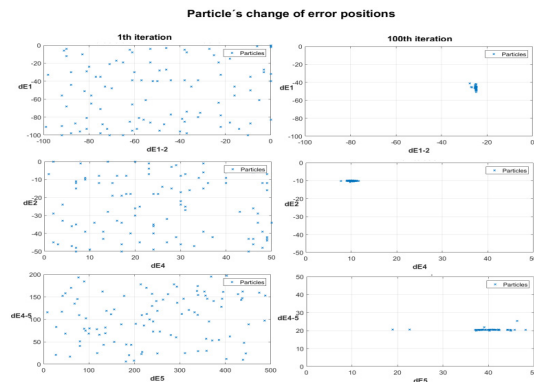


Figure 13. Particle's position of the change of error in the PSO

The graphs were presented in pairs, in some cases the scale of the graph was altered in order to visualize more clearly the positioning of the particle in its search space, since some particles differ from the search range of others.

From the moment that some particles find a more promising region within the search space, there are both individual and collective learning characteristics on their part, at the same time.; all particles receive a portion of influence from the one that found the most promising region as from their own search region.

The movements of the particles in the swarm are also presented in relation to the variable errors (E) and to the error's change (ΔE), as the particles converge to a point which represents the minimum value of the objective function.

Despite the difficulty of finding the 20 variables of the problem, which can allow the good dynamic behaviour of the FLC, the metaheuristic PSO obtained a considerable performance in this search, its characteristic of exploration of the space of search of several directions at the same time allowed the precision in the determination of the solution.

As in PSO, GA can also be considered a population-based metaheuristic method, but there is a wider range of parameters to be defined that makes it more difficult to work properly for the problem. However, in general, the adjusted GA parameters obtained a good final result. The two algorithms were chosen because they presented much similarity in their dynamics. The following are the results obtained by GA, as showing in Figure 14.

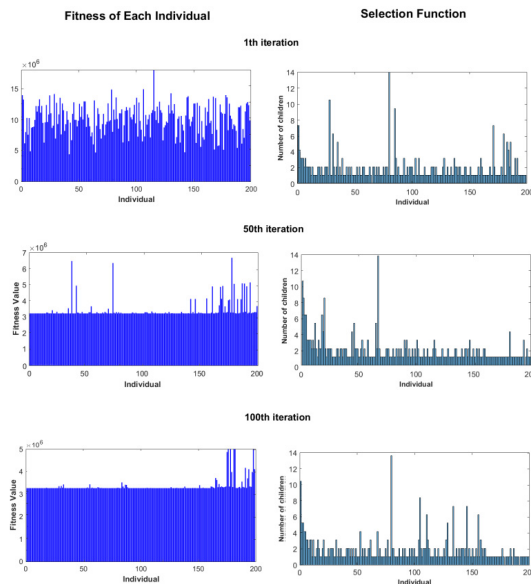


Figure 14. GA results of the selected individuals

According to the evolutionary characteristics of the algorithm, each individual (solution) has an initial fitness value, restricted to the search space. In the course of the generations, the best individuals are selected to generate children, which, in turn, are also evaluated according to the function of the evaluation within the optimization process.

Each individual is composed of the FLC parameters tuning of 20 variables of the FLC tuning. In this work, 200 individuals were used in the process; in addition, the used stop criterion was of maximum 200 generations.

The results below show the behaviour of GA in relation to the individual fitness value, in

addition to the number of children generated by the individuals.

Although the PSO algorithm has a swarm population smaller than the population of individuals in GA, it results in a lower J value than that found by GA, which has 200 individuals in its population (see Figure 15). This can be explained by a greater difficulty of defining the various GA parameters when compared to defining those required by the PSO. Overall, both algorithms had very satisfactory results.

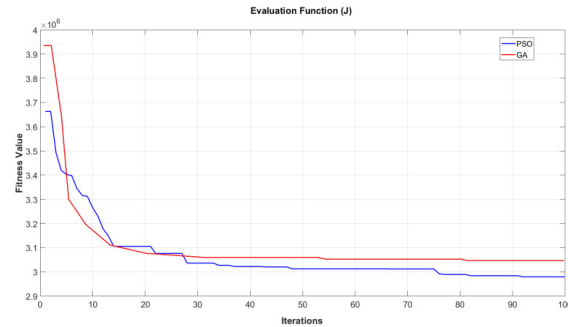


Figure 15. Behavior of the evaluation function (J) for each metaheuristic

The parameters of the manual, PSO and GA settings are shown below in Table 2.

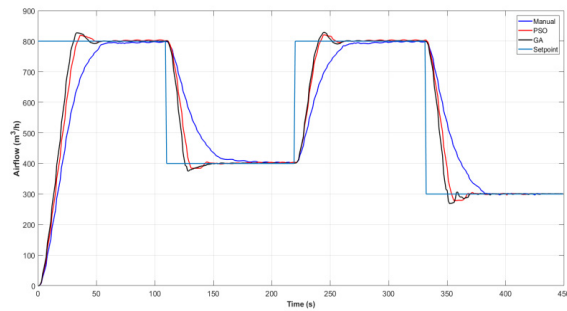
TABLE 2. FLC parameters tuning

FLC parameters	Manual	PSO	GA
$E1$	-500,0	-226,0	-195,0
$E1-2$	-100,0	-61,0	-80,0
$E2$	-60,0	-20,0	-18,0
$E2-3$	-2,0	-2,0	-2,0
$E3$	0	0	0
$E3-4$	2,0	2,0	2,0
$E4$	60,0	20,0	22,0
$E4-5$	100,0	80,0	80,0
$E5$	500,0	310,0	294,0
$\Delta E1$	-20,0	-48,0	-42,0
$\Delta E1-2$	-15,0	-25,0	-20,0
$\Delta E2$	-8,0	-10,0	-13,0
$\Delta E2-3$	-1,0	-2,0	-2,0
$\Delta E3$	0	0	0
$\Delta E3-4$	1,0	2,0	2,0
$\Delta E4$	8,0	10,0	12,0
$\Delta E4-5$	15,0	20,0	18,0
$\Delta E5$	20,0	40,0	34,0
yNP	-0,6	-1,0	-1,0
yNG	-7,0	-6,0	-6,0
yPP	0,6	0,5	0,650
yPG	7,0	8,0	7,0

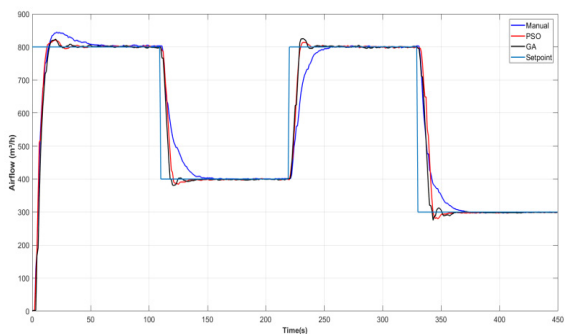
It was verified experimentally that the central variables of the FLC (E2-3, E3 and E3-4) are very sensitive to small variations of the values; a small perturbation in these variables could make the system unstable. In order to overcome this situation, these were eliminated from the optimization variables and kept constant in the tunings.

Regarding the variables E and ΔE , it was necessary to approximate the values of the FLC tuning, resulting from the computational optimization, to an integer, necessary for the programming of the PLC.

Figure 16 presents the the FLC dynamics tuning of this work, both in the computational simulation environment, as well as in the real fan system present in LAMOTRIZ.



(a) Computational environment



(b) Real environment

Figure 16. Dynamical behavior of the fan system in different environments

TABLE 3. FLC tuning performance comparison for different tunings

FLC Tuner	Evaluation function	Overshoot (%)		Rise Time (Sec.)		Settling Time (Sec.)	
		1st step	3rd step	1st step	3rd step	1st step	3rd step
Man.	8.5612e+06	5.6	0.4	7.8	16.3	41.2	29.3
PSO	2.9798 e+06	2.5	1.7	7.2	6.4	22.5	16.1
GA	3.12594e+06	3.0	3.1	7.7	6.1	26.7	15.2

Table 3 presents a qualitative comparison of the three tunings applied to the FLC, displaying some of the main parameters in the control evaluation of a dynamic system.

It can be observed that the metaheuristics were able to find better conditions for the FLC, comparing to the manual adjustment, evidenced by the results of the main criteria of dynamic control of the system. The overshoot, rise time and settling time were smaller in both metaheuristics, i.e., faster transient response and more stability in the setpoint.

Despite the good result obtained manually for FLC, this method, based on the attempt error, requires a lot of effort from the part of the operator, since in this FLC are at least 20 variables to find at the same time. The manual adjustment of the FLC presented a slow dynamic behaviour in the transient response of the setpoint change. The time period that the control signal needed to stabilize the reference value was considered long in relation to the time of other applied tunings.

Another important analysis is a comparison between the FLC, adjusted by the metaheuristic, and a control widely used in the industry, as in the case of Proportional Integral (PI), standard controller adopted by the CLP. Figure 17 shows the behavior of the system for this comparison. Note that the results were very similar, with a slight improvement of the FLC in the initial overshoot. This demonstrates that FLC can be applied in the control of plants with these characteristics in order to have a behavior similar to that obtained with the PI, and may even be better depending on the adjustment of the FLC parameters.

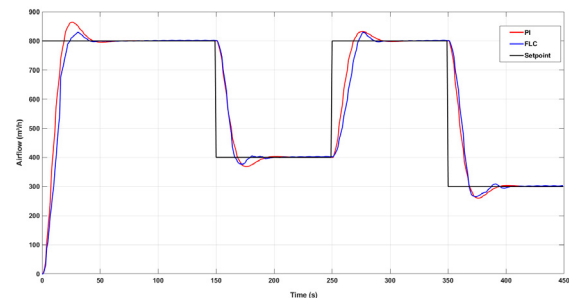


Figure 17. Dynamical behavior of the fan system with PI and FLC controllers

In addition, using the same methodology previously applied to the industrial fan system present in LAMOTRIZ, an FLC was made for two other systems, in order to expand and demonstrate its applicability in different situations. Two different systems, a Foucault brake system (see 18 (a)) and a pumping system (see Figure 18 (b)). The PI and

the FLC with metaheuristic adjustment are used again in both plants of the systems, it is possible to observe the controllers in the different plants of the systems have very similar characteristics.

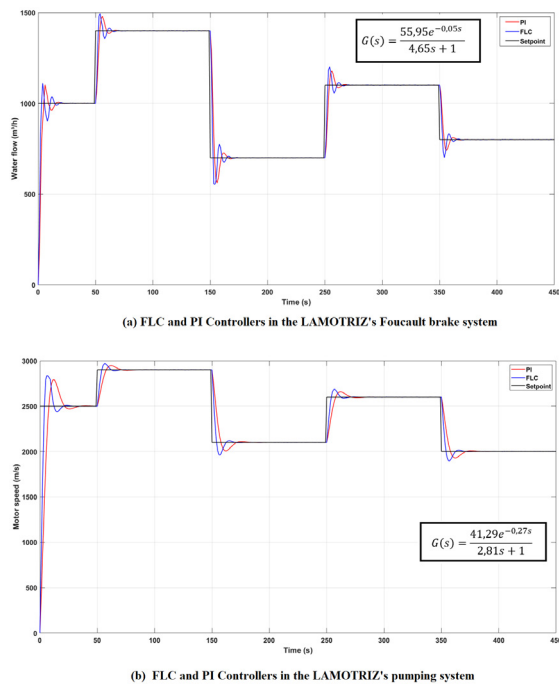


Figure 18. Dynamical behavior of the different systems with PI and FLC controllers

The parameters of the FLC tuning by means of the optimization techniques have the ability to adapt to the required parameter setting requirements, virtually eliminating the requirement additional adjustments when compared to the conventional tuning methods (Lam et al., 2016).

In this work, the application of metaheuristics PSO/GA to adjust the FLC was used only for the positions of parameters of the inferences

of the error and error's change, plus the output functions, not acting on the fuzzy rules system, which was predefined by the operator; despite the difficulty of finding the 20 variables for the FLC, the dynamic behaviour resulting from the optimization showed significant improvements in the dynamic performance of the FLC.

6. Conclusion

The FLC developed in ladder programming codes for PLC, using trapezoidal functions for MFs, was able to obtain good quality practical results. Similar methods of metaheuristic optimization based on population characteristics, particle swarm optimization and genetic algorithms were used with the main objective of improving the dynamic performance of the FLC, obtaining a higher performance in controlling the airflow of an industrial fan present in LAMOTRIZ.

The total was 20 variables submitted to the optimization process, error, error's change and output gains. The results obtained from the optimization were compared with the manually adjusted FLC performance.

Those results demonstrated the effectiveness of the metaheuristics techniques in the optimization of the FLC's MFs parameters applying to PLC system, resulting in a control with a faster and stable response, without the necessity of any additional adjustment when implemented in the real control system. In general, it is possible to expand the application of this type of tuning to other control models, different types of plants, among other applications in the area of control and automation of industrial plants.

REFERENCES

1. Aisbett, J. & Rickard, J. T. (2013). Centroids of fuzzy sets when membership functions have spikes. In *2013 Joint IFSA World Congress and NAFIPS Annual Meeting*, Edmonton, Alberta, Canada (pp. 97-101).
2. Altas, I. H. & Sharaf, A. M. (2007). A Generalized Direct Approach for Designing Fuzzy Logic Controllers in Matlab/Simulink GUI Environment, *International Journal of Information Technology and Intelligent Computing*, 1(4).
3. Aquino, R. R. B. et al. (2016). An emotional controller PLC implementation for an industrial fan system. In *2016 International Joint Conference on Neural Networks (IJCNN)* (in Portuguese), Vancouver, BC, Canada (pp. 3889-3895).
4. Araydah, W., Tutunji, T. A. & Al-Naimi, L. (2017). System identification for a liquid flow process. In *2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, Aqaba, Jordan (pp. 1-6).

5. Clerc, M. & Kennedy, J. (2002). The particle swarm explosion, stability, and convergence in a multidimensional complex space, *IEEE Transactions on Evolutionary Computation*, 6(1), 58-73.
6. Dash, P. K., Pradhan, A. K. & Panda, G. (2000). A novel fuzzy neural network based distance relaying scheme, *IEEE Transactions Power Delivery*, 15(3), 902–907.
7. Dong, Y. & Yang-Ming, X. (2017). Application of PLC, *Science & Technology Information*, 19, 310.
8. Dorf, R. C. & Bishop, R. H (2010). *Modern Control Systems Pearson*, 12th ed. Prentice Hall.
9. Dorigo, M., Maniezzo, V. & Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents, *IEEE Transactions on Systems, Man, and Cybernetics - Part B.*, 26(1), 29-41.
10. Du, X., Ying, H. & Lin, F. (2009). Theory of extended fuzzy discrete-event systems for handling ranges of knowledge uncertainties and subjectivity, *IEEE Transactions on Fuzzy Systems*, 17(2), 316-328.
11. Fang, G., Kwok, N. M. & Ha, Q. (2008). Automatic Fuzzy Membership Function Tuning Using the Particle Swarm Optimization. In *2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, Wuhan, China (pp. 324-328).
12. Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Addison-Wesley Longman Publishing Co., Boston.
13. Kennedy, J., Eberhart, R. & Shi, Y. (2001). *Swarm Intelligence*. Morgan Kaufmann Publishers, San Mateo, California.
14. Kumar, A., Sharma, M. & Gautam, P. K. (2016). GA Based Parameter Optimization of Fuzzy-PID Controller, *International Journal of Innovative Research in Technology & Science*, 4(1), 18-21.
15. Lam, H. K. et al. (2016). Membership-Function-Dependent Stability Analysis and Control Synthesis of Guaranteed Cost Fuzzy-Model-Based Control Systems, *International Journal of Fuzzy Systems*, 537-549. Springer Berlin Heidelberg, Print ISSN1562-2479.
16. Leondes, C. T. (1996). *Digital Control Systems Implementation and Computational Techniques, Volume 79*, 1st edition. Academic Press School of Engineering and Applied Science, University of California, Los Angeles.
17. Liang, H. T. & Kang, F. H. (2016). Adaptive mutation particle swarm algorithm with dynamic nonlinear changed inertia weight, *International Journal for Light and Electron Optics*, 127(19), 8036-8042.
18. Liu, B., Chen, C. & Tsao, J. (2001). Design of adaptive fuzzy logic controller based on linguistic-hedge concepts and genetic algorithms, *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, 31(1), 32-53.
19. Mahato, B., Maity, T. & Antony, J. (2015). Embedded Web PLC: A New Advances in Industrial Control and Automation. In *2015 Second International Conference on Advances in Computing and Communication Engineering*, Dehradun, India (pp. 156-160).
20. Mekki, A. O., Lin, F., Ying, H. & M. J. Simoff. (2017). Fuzzy detectabilities for fuzzy discrete event systems. In *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Naples, Italy (pp. 1-6).
21. Meng, A., Li, Z., Yin, H., Chen, S. & Guo, Z. (2016). Accelerating particle swarm optimization using crisscross search, *Information Sciences*, 329, 52-72.
22. Mustafa, G. I. Y., Wang, H. & Tian, Y. (2019). Model-free Adaptive Fuzzy Logic Control for a Half-car Active Suspension System, *Studies in Informatics and Control*, 28(1), 13-24, DOI: 10.24846/v28i1y201902
23. Ogata, K. (2009). *Modern Control Engineering*, 5th ed. Prentice Hall.
24. Olivas, F., Valdez, F., Castillo, O. & Melin, P. (2016). Dynamic parameter adaptation in particle swarm optimization using interval type-2 fuzzy logic, *Soft Computing*, 20(3), 1057-1070.
25. Pelikan, M. & Goldberg, D. E. (2001). Escaping hierarchical traps with competent genetic algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)* (pp. 511-518).

26. Pitalúa Díaz, N., Lagunas Jiménez, R. & González, A. (2013). Tuning Fuzzy Control Rules via Genetic Algorithms: An Experimental Evaluation, *Research Journal of Recent Sciences*, 2(10), 81-87.
27. Takagi, T. & Sugeno, M. (1985). Fuzzy identification of systems and its application to modeling and control, *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-15(1), 116-132.
28. Tanweer, M. R., Suresh, S. & Sundararajan, N. (2015). Self regulating particle swarm optimization algorithm, *Information Sciences*, 294, 182-202.
29. Tsutsui, S. & Ghosh, A. (1997). Genetic Algorithms with a Robust Solution Searching Scheme, *IEEE Transactions on Evolutionary Computation*, 1(3), 201-208.
30. Wang, H., Wang, W. & Wu, Z. (2013). Particle swarm optimization with adaptive mutation for multimodal optimization, *Applied Mathematics and Computation*, 221, 296-305.
31. Wang, Y., Xia, Y. & Zhou, P. (2009). Fuzzy-Model-Based Sampled-Data Control of Chaotic Systems: A Fuzzy Time-Dependent Lyapunov–Krasovskii Functional Approach, *IEEE Transactions on Fuzzy Systems*, 25(6), 1672-1684.
32. Zadeh, L. A. (1994). The role of fuzzy logic in modeling, identification and control, *Modeling, Identification and Control*, 15(3), 191-203.
33. Zhang H., Zhang, B. & Wang, F. (2009). Automatic Fuzzy Rules Generation Using Fuzzy Genetic Algorithm. In *2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery*, Tianjin, China (pp. 107-112).
34. Zhou, W., Li, M., Yin, H. & Ma, C. (2014). An adaptive fuzzy logic based energy management strategy for electric vehicles. In *Proceedings of the 2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE)*, Istanbul, Turkey (pp. 1778-1783).