Fuzzy Identification Based on a Chaotic Particle Swarm Optimization Approach Applied to a Nonlinear Yo-yo Motion System

Leandro dos Santos Coelho and Bruno Meirelles Herrera

Abstract—The identification of uncertain and nonlinear systems is an important and challenging problem. Fuzzy models, particularly Takagi–Sugeno (TS), have received particular attention in the area of nonlinear identification due to their potentialities to approximate any nonlinear behavior. A method of nonlinear identification based on the TS fuzzy model and optimization procedure is proposed in this paper. Chaotic particle swarm optimization (CPSO) algorithms, based on chaotic Zaslavskii map sequences, combined with efficient Gustafson–Kessel (GK) clustering algorithm are proposed here for the design of the premise part of production rules, while the least-mean-square technique is utilized for the subsequent part of the production rules of the TS fuzzy model. An experimental case study using a nonlinear yo-yo motion control system is analyzed by the proposed algorithms. The numerical results presented here indicate that the traditional particle swarm optimization algorithm and, particularly, the CPSO combined with GK algorithms are effective in building a good TS fuzzy model for nonlinear identification.

Index Terms—Chaotic map, clustering algorithm, fuzzy identification, nonlinear systems, optimization, particle swarm optimization, system identification.

I. INTRODUCTION

Knowledge of the behavior of real systems based on dynamic models is important in many fields of science and engineering. System identification can be described as the science of building mathematical models of dynamic systems based on observed input and/or output signals. The practical importance of process-model identification has been recognized for many years [1].

Recently, nonlinear techniques using fuzzy identification and control have received a great deal of attention [2]–[6]. A central feature of fuzzy systems is that they are based on the concept of fuzzy coding of information and operating with fuzzy sets instead of numbers [7]. Takagi–Sugeno’s (TS) fuzzy model [8], [9] exhibits both high nonlinearity and simple structure. As reported in literature, it is capable of approximating a complex system using fewer fuzzy rules than conventional Mamdani-type fuzzy models [10].

The identification problem in TS modeling consists of two major parts: structure identification and parameter identification. Furthermore, the TS model comprises premise-part identification and consequent-part identification. The identification of the premise part consists of determining the premise-space partition and extracting the number of rules. The consequent-part identification consists of determining the structure of the rules’ output parts. Lastly, the parameter learning task consists of determining the system parameters, so that a performance measurement based on the output errors is minimized.

TS fuzzy models can be designed by means of clustering, classical nonlinear optimization methods, evolutionary algorithms, and others. An alternative is the investigation of particle swarm optimization (PSO) for the optimization of membership functions parameters in the TS model design.

Social-insect societies are distributed systems which, despite the simplicity of their individuals, present a highly structured social organization. As a result of this organization, insect societies can accomplish complex tasks that, in some cases, far exceed the individual capabilities of a single ant [11]. The field of swarm intelligence is an emerging research area that presents features of self-organization and cooperation principles among group members bioinspired by social-insect societies [12]. Swarm intelligence is inspired by nature, based on the fact that the live animals of a group contribute with their individual experiences to the group, rendering it stronger to face other groups. The most familiar representatives of swarm intelligence in optimization problems are the food-searching behavior of ant colonies [11], PSO [13], artificial immune systems [14], and bacterial foraging [15].

Most heuristic optimization methods combined with fuzzy systems for nonlinear identification applications are used for rule extraction, tuning of membership functions, and training of fuzzy models using genetic algorithms. In this context, several hybrid methodologies of fuzzy models and genetic algorithms have been intensively developed, with a variety of works in nonlinear identification and control published in the literature in recent years [16]–[21]. The use of metaheuristics of PSO is an emerging approach in the TS fuzzy-model design [22]–[26].

In this paper, the structural identifications of the premise and consequent parts of production rules of the TS fuzzy model are performed separately. This paper presents the design of the TS...
fuzzy model based on the chaotic particle swarm optimization (CPSO) approaches combined with the Gustafson–Kessel (GK) clustering algorithm [27] for the premise-part design, and it also shows the least mean squares for calculation of the consequent part of production rules of the TS fuzzy model for nonlinear identification. An experimental case study using a yo-yo motion system is analyzed by the proposed design approach for nonlinear identification. The numerical results presented here indicate that the PSO and, particularly, the CPSO combined with the GK algorithm are effective in building a good TS fuzzy model for nonlinear identification.

The remaining sections of this paper are organized as follows. Section II describes the fundamentals of TS fuzzy models. Section III then describes the optimization procedure based on the concept of PSO and CPSO approaches. Section IV discusses the yo-yo motion system prototype, while Section V analyzes the nonlinear identification results applied to the TS fuzzy model. Lastly, Section VI presents our conclusion.

II. TS Fuzzy Model

Developing mathematical models of real systems is a central topic in many fields of engineering and science. Models can be used for computational simulations, analysis of complex systems, design of new industrial processes, and control of systems [28].

For nonlinear dynamic systems, traditional techniques of modeling and identification are difficult to implement and sometimes impracticable. However, other techniques based on fuzzy systems, such as the TS fuzzy model, are increasingly used for the identification of this kind of process. The TS fuzzy model is based on rules in which the consequent is not a linguistic variable, as in the Mamdani-type fuzzy model, but a function of the input variables. A relevant aspect of the TS fuzzy model is its power of representation, particularly in describing complex processes. This fuzzy system allows complex systems to be decomposed into simple subsystems.

The identification of the TS fuzzy model involves two primary tasks: parameter tuning and structure optimization. The parameter tuning procedure deals with the estimation of a feasible set of parameters for a given structure. The structure optimization procedure aims to find the optimal structure of the local models, the relevant premise variables, and the suitable partition of the premise space. The TS models consist of linguistic IF-THEN rules that can be represented by the following general form:

\[
R^{(j)} : \text{IF } z_1 \text{ IS } A_1^j \text{ AND, \ldots, AND } z_m \text{ IS } A_m^j \text{ THEN } g_j = w_0^j + w_1^j u_1^j + \cdots + w_q_j^j u_q_j^j. \tag{1}
\]

The IF preconditioned statements define the premise part, while the THEN rule functions constitute the consequent part of the fuzzy model; \( z = [z_1, \ldots, z_m]^T \), \( i = 1, \ldots, m \), is the input vector of the premise \( p \), and \( A_i^j \) are the labels of fuzzy sets. The parameter \( \bar{u} = [u_1^j, \ldots, u_q_j^j]^T \) represents the input vector to the consequent part of \( R^{(j)} \) that comprises \( g_j \) terms; \( g_j = g_j(u^j) \) denotes the \( j \)th rule output, which is a linear polynomial of the consequent input term \( u_i^j \), and \( w_j^j = [w_0^j, w_1^j, \ldots, w_q_j^j]^T \) are the polynomial coefficients that form the consequent parameter set. Each linguistic label \( A_i^j \) is associated with a membership function, \( \mu_{A_i^j}(z_i) \), which is described by

\[
\mu_{A_i^j}(z_i) = \exp \left[ -\frac{1}{2} \frac{(z_i - m_{ij})^2}{\sigma_{ij}} \right] \tag{2}
\]

where \( m_{ij} \) and \( \sigma_{ij} \) are the center and the spread of the Gaussian-type membership function, respectively. The union of all these parameters formulates the set of premise parameters. The firing strength of rule \( R^{(j)} \) represents its excitation level and is given by

\[
\nu_j(z) = \prod_{i=1}^{M} \mu_{A_i^j}(z_i). \tag{3}
\]

The fuzzy sets pertaining to a rule form a fuzzy region (cluster) within the premise space, \( A_1^j x A_2^j x \cdots x A_m^j \), with a membership distribution described by (3). Given the input vectors \( z \) and \( u_j^j \), \( j = 1, \ldots, M \), the final output of the fuzzy model is inferred by taking the weighted average of the local outputs \( g_j(u_j^j) \), which is given by

\[
y = \sum_{j=1}^{M} \nu_j(z) \cdot g_j(u_j^j). \tag{4}
\]

where \( M \) denotes the number of rules, and \( \nu_j(z) \) is the normalized firing strength of \( R^{(j)} \), which is defined as

\[
\nu_j(z) = \frac{\mu_j(z)}{\sum_{j=1}^{M} \mu_j(z)}. \tag{5}
\]

The TS model’s structure is identified based on the PSO or the CPSO combined with a GK algorithm for the premise-part optimization, while the consequent part is optimized by the least-mean-square method [1]. The PSO and the new CPSO approaches for fuzzy-model optimization are presented in the section below.

III. PSO AND CPSO APPROACHES FOR OPTIMIZATION OF THE TS Fuzzy MODEL

A. Fundamentals of the Classical PSO Approach

The proposal of PSO algorithm was put forward by several scientists who developed computational simulations of the movement of organisms such as flocks of birds and schools of fish. Such simulations were heavily based on manipulating the distances between individuals, i.e., the synchrony of the behavior of the swarm was seen as an effort to keep an optimal distance between them [29].

In theory, at least, individuals of a swarm may benefit from the prior discoveries and experiences of all the members of a swarm when foraging. The fundamental point of developing PSO is a hypothesis in which the exchange of information among creatures of the same species offers some sort of evolutionary advantage.
The PSO originally developed by Kennedy and Eberhart in 1995 [30], [31] is a population-based swarm algorithm. Similarly to a genetic algorithm [32], the PSO is an optimization tool based on a population, where each member is seen as a particle, and each particle is a potential solution to the problem under analysis. Each particle in the PSO has a randomized velocity associated with it, which moves through the space of the problem. However, unlike genetic algorithms, the PSO does not have operators, such as crossover and mutation. The PSO does not implement the survival of the fittest individuals; rather, it implements the simulation of social behavior.

Each particle in the PSO keeps track of its coordinates in the problem space, which are associated with the best solution (fitness) it has achieved so far. This value is called personal best (pbest). Another “best” value that is tracked by the global version of the PSO is the best value considering that particle has achieved so far in the search, which are associated with the best solution (fitness) it has achieved so far. This value is called global best (gbest).

The PSO concept consists of, in each time step, changing (accelerating) the velocity of each particle flying toward its pbest and gbest locations (global version of the PSO). Acceleration is weighted by random terms, with separate random numbers being generated for acceleration toward pbest and gbest locations, respectively. The procedure for implementing the global version of PSO is given by the following steps [33]–[35].

1) Initialize a population (array) of particles with random positions and velocities in the -dimensional problem space using a uniform probability distribution function.

2) Evaluate the fitness value of each particle.

3) Compare each particle’s fitness with the particle’s pbest. If the current value is better than the pbest, then set the pbest value equal to the current value and the pbest location equal to the current location in -dimensional space.

4) Compare the fitness with the population’s overall previous best. If the current value is better than gbest, then reset gbest to the current particle’s array index and value.

5) Change the velocity and position of the particle according to (6) and (7), respectively [36], [37]

\[
\nu_i(k + 1) = w \cdot \nu_i(k) + c_1 \cdot u_{d_{i,j}}(k) \cdot [p_i(k) - x_i(k)] + c_2 \cdot U_{d_{i,j}}(k) \cdot [g_i(k) - x_i(k)] \tag{6}
\]

\[
x_i(k + 1) = x_i(k) + \Delta t \cdot \nu_i(k + 1). \tag{7}
\]

Equation (7) represents the position updating, according to its previous position and its velocity [see (6)], considering that \(\Delta t = 1\), where \(k\) is the current iteration number. \(x_i = [x_{i1}, x_{i2}, \ldots, x_{in}]^T\) stands for the position of the \(i\)th particle and \(n\)th decision variable, \(\nu_i = [\nu_{i1}, \nu_{i2}, \ldots, \nu_{in}]^T\) stands for the velocity of the \(i\)th particle, and \(p_i = [p_{i1}, p_{i2}, \ldots, p_{in}]^T\) represents the best previous position (the position giving the best fitness value) of the \(i\)th particle. Index \(g\) represents the index of the best particle among all the particles in the group. Variables \(ud_{i,j}(t)\) and \(Ud_{i,j}(t)\) are two random functions based on uniform probability distribution functions in the range \([0,1]\) of the \(j\)th design variable of \(i\)th particle.

6) Loop to step 2) until a stopping criterion is met, usually a sufficiently good fitness and a maximum number of iterations (generations), \(k_{max}\).

The use of variable \(w\), called the inertia weight, was proposed by Shi and Eberhart [38]. This parameter is responsible for dynamically adjusting the velocity of the particles; therefore, it is responsible for balancing between local and global search, hence requiring fewer iterations for the algorithm to converge. A low value of inertia weight implies a local search, while a high value leads to a global search.

Applying a high inertia weight at the start of the algorithm and making it decay to a low value through the PSO execution make the algorithm search globally, at the beginning of the search, and search locally at the end of the execution. The following weighting function \(w\) is used in (6):

\[
w = w_{max} - \frac{w_{max} - w_{min}}{k_{max}} k. \tag{8}
\]

Equation (8) shows how the inertia weight is updated, considering that \(w_{max}\) and \(w_{min}\) are the initial and final weights, respectively [39].

Positive constants \(c_1\) and \(c_2\) are called cognitive and social components, respectively. These are the acceleration constants responsible for varying the particle velocity toward the pbest and the gbest. Particle velocities in each dimension are clamped to a maximum velocity \(V_{max}\). If the velocity in that dimension exceeds \(V_{max}\), which is a parameter specified by the user, then the velocity in that dimension is limited to \(V_{max}\).

\(V_{max}\) is a parameter serving to determine the resolution with which the regions around the current solutions are searched. If \(V_{max}\) is too high, the PSO facilitates a global search, and particles might fly past good solutions. Conversely, if \(V_{max}\) is too small, the PSO facilitates a local search, and particles may not explore sufficiently beyond locally good regions. Previous experience with PSO (trial and error, mostly) led us to set the \(V_{max}\) to 20% of the dynamic range of the particle in each dimension.

B. Chaotic PSO Approach

Chaos theory is recognized as very useful in many engineering applications. An essential feature of chaotic systems is that small changes in the parameters or the starting values for the data lead to different future behaviors, such as stable fixed points, periodic oscillations, bifurcations, and ergodicity. These behaviors can be analyzed based on the meaning of Lyapunov exponents and the attractor theory [40], [41].

Based on these features, much of the chaos as a science is connected with the notion of “sensitive dependence on initial conditions” [60]. An apparent paradox is that chaos is deterministic, which is generated by fixed rules that do not themselves involve any elements of change. In principle, the future is completely determined by the past, but, in practice, small uncertainties are unpredictable in the long term [42], [43].
Optimization algorithms based on the chaos theory are stochastic search methodologies that differ from any of the existing evolutionary algorithms. Due to the nonrepetition of chaos, it can carry out overall searches at higher velocities than stochastic ergodic searches that depend on probabilities.

In the PSO design, the main advantage of chaotic optimization approaches is the maintenance of population diversity in the problem of interest. Based on Clerc and Kennedy [44] and Trelea’s [45] studies, the parameters \( w, c_1, \) and \( c_2 \) are generally the key factors that affect the classical PSO convergence. In fact, however, parameters \( c_1 \) and \( c_2 \) cannot entirely ensure the ergodicity of the optimization in phase search because they are constant factors in the classical PSO. The variable \( w \) is also limited for procedures of linear or exponential reduction with the evolution of generations.

Therefore, this paper provides approaches introducing chaotic mapping with ergodicity, irregularity, and stochastic property in the PSO to improve the global convergence. The use of chaotic sequences in the PSO can be helpful to escape more easily from local minima than can be done through the traditional PSO.

The literature is rich in chaotic time-series sequences, such as logistic map, tent map, Hénon map, Ikeda map, Chua’s system, Lorenz systems, Lozi map, and others [40], [41], [46]. The interesting dynamic system evidencing chaotic behavior is the Zaslavskii map [47], whose equation is given by

\[
y_1(t) = \text{mod} \left[ y_1(t-1) + \nu + ay_2(t) \right]  \\
y_2(t) = \cos \left( 2\pi y_1(t-1) \right) + e^{-\epsilon} y_2(t-1)
\]

where mod is the modulus (signed remainder after division). The Zaslavskii map shows a strange attractor with large Lyapunov exponent for \( \nu = 400, r = 3, \) and \( a = 12.6695, \) as shown in Fig. 1. In this case, the values of \( y_2(t) e^{-1.0512, 1.0512} \).

The design of methods to improve the convergence of PSO is a challenging issue in the design of optimization methods. New PSO approaches are proposed here based on the Zaslavskii map.

The convergence properties of the PSO methods are strongly connected to the random sequence applied on variation operators of (6) during a run. In particular, it can be shown that when different random sequences are used during the evolution, the final results may effectively be very close but not equal [48]. However, there are no analytical results that guarantee an improvement of the performance indices of the PSO methods depending on the choice of particular generator in (6) instead of classical technique based on variables \( u_{d_{i,j}}(t) \) and \( U_{d_{i,j}}(t) \). Recently, chaotic sequences have been adopted instead of random ones, and interesting results have been shown in optimization applications [49]–[55].

The performance of the simple PSO greatly depends on its design parameters, and it often suffers the problem of being trapped in local optima so as to be premature convergence. The choice of the chaotic Zaslavskii map in the PSO design is justified theoretically by its unpredictability, i.e., by its spread-spectrum characteristic and large Lyapunov exponent (a quantitative measure of chaos) [56], [57]. Details of the Lyapunov exponent are presented in [40]–[42], and [46]. In this context, due to ergodic and dynamic properties of the Zaslavskii-map variables, the PSO using chaotic search based on the chaotic Zaslavskii map is more capable of escaping from local optima than random search.

These new PSO approaches combined with chaotic sequences (CPSO) based on the Zaslavskii map are described as follows:

Approach 1— CPSO1: Parameter \( c_1 \) of (6) is modified by the following equation:

\[
u_i(k+1) = w \cdot \nu_i(k) + z_1 \cdot u_{d_{i,j}}(k) \cdot [p_i(k) - x_i(k)] \\
+ c_2 \cdot U_{d_{i,j}}(k) \cdot [p_j(k) - x_i(k)]
\]

where \( z_1 \) is a function based on the results of \( y_2(t) \) of the Zaslavskii map with scaled values between 0.5 and 2.5.

Approach 2— CPSO2: Parameter \( c_2 \) of (6) is modified by the following equation:

\[
u_i(k+1) = w \cdot \nu_i(k) + c_1 \cdot u_{d_{i,j}}(k) \cdot [p_i(k) - x_i(k)] \\
+ z_2 \cdot U_{d_{i,j}}(k) \cdot [p_j(k) - x_i(k)]
\]

where \( z_2 \) is a function of the results of \( y_2(t) \) of the Zaslavskii map with scaled values between 0.5 and 2.5.

Approach 3— CPSO3: Parameters \( c_1 \) and \( c_2 \) of (6) are modified by the following equation:

\[
u_i(k+1) = w \cdot \nu_i(k) + z_1 \cdot u_{d_{i,j}}(k) \cdot [p_i(k) - x_i(k)] \\
+ z_2 \cdot U_{d_{i,j}}(k) \cdot [p_j(k) - x_i(k)]
\]

where \( z_1 \) and \( z_2 \) are functions of the results of \( y_2(t) \) of the Zaslavskii map with scaled values between 0.5 and 2.5.
Approach 4— CPSO4: Parameter $w$ of (6) is modified by the following equation:

$$
\nu_i(k + 1) = w_z \cdot \nu_i(k) + c_1 \cdot ud_{i,j}(k) \cdot [p_i(k) - x_i(k)] \\
+ c_2 \cdot Ud_{i,j}(k) \cdot [p_y(k) - x_i(k)]
$$

(14)

where $w_z$ is a function based on the Zaslavskii map with scaled values between 0.4 and 0.9.

Approach 5— CPSO5: Parameters $w$ and $c_1$ of (6) are modified by the following equation:

$$
\nu_i(k + 1) = w_z \cdot \nu_i(k) + z_1 \cdot ud_{i,j}(k) \cdot [p_i(k) - x_i(k)] \\
+ c_2 \cdot Ud_{i,j}(k) \cdot [p_y(k) - x_i(k)]
$$

(15)

where $w_z$ is a function based on the results of $y_2(t)$ of the Zaslavskii map with scaled values between 0.4 and 0.9, and $z_1$ is a function based on the Zaslavskii map with values between 0.5 and 2.5.

Approach 6— CPSO6: Parameters $w$ and $c_2$ of (6) are modified by the following equation:

$$
\nu_i(k + 1) = w_z \cdot \nu_i(k) + c_1 \cdot ud_{i,j}(k) \cdot [p_i(k) - x_i(k)] \\
+ z_2 \cdot Ud_{i,j}(k) \cdot [p_y(k) - x_i(k)]
$$

(16)

where $w_z$ is a function based on the results of $y_2(t)$ of the Zaslavskii map with scaled values between 0.4 and 0.9, and $z_2$ is a function based on the Zaslavskii map with values between 0.5 and 2.5.

Approach 7— CPSO7: Parameters $w$, $c_1$, and $c_2$ of (6) are modified by the following equation:

$$
\nu_i(k + 1) = w_z \cdot \nu_i(k) + z_1 \cdot ud_{i,j}(k) \cdot [p_i(k) - x_i(k)] \\
+ z_2 \cdot Ud_{i,j}(k) \cdot [p_y(k) - x_i(k)]
$$

(17)

where $w_z$ is a function based on the results of $y_2(t)$ of the Zaslavskii map with scaled values between 0.4 and 0.9, and $z_2$ is a function based on the Zaslavskii map with values between 0.5 and 2.5.

C. PSO or CPSO Combined With GK Clustering Algorithm

Clustering based on fuzzy approaches is a relevant application of fuzzy set theory and concepts of membership functions. The membership function of an object describes to what degree that object is a member of a given set [2].

Cluster analysis is a technique that is used to seek out data, dividing all objects (samples) into smaller subgroups and classifying them according to the similarities among them. A fuzzy cluster is a fuzzy subset of the set of objects, with the membership function of each object representing the degree to which it belongs to that cluster [58].

There are many conceptions of clustering method in the literature [28], [59]–[66]. One of the most efficient of these clustering methods is the GK clustering algorithm. Gustafson and Kessel [27] extended the standard fuzzy $c$-means algorithm by employing an adaptive distance norm in order to detect clusters of different geometrical shapes in one data set. Details of GK clustering algorithm are presented in [2], [27], and [63].

Clustering techniques, including the GK algorithm, belong to the classes of unsupervised learning methods since they do not use prior class identifiers. The GK algorithm is based on the minimization of a cost function (objective function) regarding the degree to which the data belong to the clusters and the degree of dissimilarity between them. The combination of the GK algorithm and the PSO or CPSO is very useful. In this case, the GK algorithm optimizes the centers of Gaussian functions of membership functions (premise part of production rules) of the TS fuzzy model. The PSO or CPSO then employs the solution of the GK algorithm in the initial population of particles and optimizes the centers and spreads of Gaussian functions. In each evaluation of a solution, the classical method of least-mean-squares calculates the consequent part of production rules of the TS fuzzy model. Fig. 2 shows the diagram of the proposed hybrid method.

IV. DESCRIPTION OF THE YO-YO MOTION SYSTEM

Yo-yo playing is considered a representative example of open-loop unstable control problems that involve intermittent dynamic environments. Stable control of yo-yo playing relies on a proper phase relationship between the controller’s action and the motion of the yo-yo [67].

Control of a yo-yo requires an asymmetric nontrivial controller with nonlinearity due to the unique features of the yo-yo system [68]. Due to its asymmetric nonlinearity, it seems difficult to control a yo-yo by a linear controller. In this context, the development of automatic control systems that efficiently control a yo-yo represents a challenge for the development of electromechanical designs [68]–[71]. One of the main difficulties is the lack of sensors to obtain the motion measure of the yo-yo position. Another difficulty is the lack of mathematical models of this measurement device type, which justifies the use of the TS fuzzy model to identify the dynamic behavior of a yo-yo motion in a real system.

The control system prototype uses a yo-yo, and a dc motor for its motion presents nonlinearity and complex behavior. The reason for using this process is the possibility of proving the efficiency and flexibility of model-based control systems. A block diagram of the described system and a photograph of the system are shown in Figs. 3 and 4, respectively [72], [73]. The components of this prototype are divided into software and hardware modules, which are described as follows.

1) Control module (software): consists of the implementation of control techniques, such as proportional–integral–derivative and fuzzy logic controllers integrated into a computer with communication with the yo-yo system using an input/output interface.

2) Sensor module (hardware/firmware): The sensors employed include digital electronic circuits (power amplification), A/D and D/A converters, and microcontroller running firmware.
3) Actuator module (hardware/firmware): consists of dc motors integrated to the sensor module, electronic circuits, and microcontroller running firmware.

4) Sensor submodule: made up of 16 infrared light-emitting diodes (LEDs) that are able to inform the position of the yo-yo.

The prototype modules are composed of hardware and firmware and are connected to the same printed circuit board (control board). The control board contains two hardware
V. Analysis Results of Nonlinear Identification

The goal of the system identification is to allow the adjustment of a mathematical model to a dynamic system structure, based on measurements collected by the adjustment of parameters and/or of the model, until the system output is as close as possible to the samples of the measured outputs. The procedure for experimental identification of a process comprises basic stages, as shown in Fig. 5.

In practice, system identification is an iterative procedure. The lack of a priori information regarding the process model will require that each step be initially examined in a superficial manner. The mathematical model, which is employed in this paper, to represent the yo-yo motion system is a Nonlinear AutoRegressive with eXogenous inputs (NARX), as shown in Fig. 6. In this case, the NARX model with series–parallel conception is used for one-step-ahead prediction of the TS fuzzy model. In this case, the resulting fuzzy inference system is a first-order TS model [28] [see (1)].

A computer with a data-acquisition board for generating the control signal (identification in closed loop using a proportional controller) and position value of the yo-yo was used to obtain system measurements. In the identification procedure based on the TS fuzzy model, 290 samples of input (tension applied to the dc motor in volts) and output (position of yo-yo in cm) were collected with a time sampling of 40 ms (see Fig. 7). The tension value corresponds to the maximum-value configuration of the driver in PWM control of a dc motor.

Experiments for the estimation phase of the mathematical model of the yo-yo motion system are carried out using samples 1 to 150. For the validation phase, the fuzzy model uses the input and output signals of samples 151 to 290. The system identification by the TS fuzzy model is appropriate if a performance index is in permissible values for the user’s needs. In this case, the fitness function for maximization proposes using the PSO and CPSO approaches and is given by the harmonic mean of multiple correlation indices of estimation and validation phases. The fitness is calculated using the following expressions:

\[
R^2_h = R^2_{est} + R^2_{val}
\]

\[
R^2_{est} = 1 - \frac{\sum_{t=1}^{150} [y(t) - \hat{y}(t)]^2}{\sum_{t=1}^{150} [y(t) - \bar{y}]^2}
\]

\[
R^2_{val} = 1 - \frac{\sum_{t=151}^{290} [y(t) - \hat{y}(t)]^2}{\sum_{t=151}^{290} [y(t) - \bar{y}]^2}
\]

where \(R^2_h\) is the harmonic mean of the multiple correlation index (fitness function to be optimized), \(R^2_{est}\) is the multiple correlation index of the estimation phase, \(R^2_{val}\) is the multiple correlation index of the validation phase, \(y(t)\) is the output of the real system, \(\hat{y}(t)\) is the output estimated by the TS fuzzy model, and \(\bar{y}\) is the mean value of the system’s output.

When \(R^2 = 1.0\), it indicates an overfitting phenomenon, i.e., a model error exists. An \(R^2\) value between 0.9 and 1.0 is considered sufficient for applications in identification and model-based control [74].
Based on previous experience with the PSO and CPSO approaches (trial and error, mostly), we set the population size $N$ equal to 20, and $k_{\text{max}}$ was set to 50 generations (stopping criterion) for the 30 runs for the TS fuzzy-model optimization. The values of $c_1 = c_2 = 0.7$ were adopted for the classical PSO1 with (6)–(8) and the CPSO1 to CPSO7 approaches.

The three chosen vectors of input for the TS fuzzy model were the following: $[u(t), y(t-1), y(t-2)]$. The space searches for centers and spreads of Gaussian membership functions of TS fuzzy models by the CPSO optimization approaches are $[-5; 5]$ and $[0.001; 5]$, respectively.

Tables I and II present the simulation results (best of 30 experiments with 50 generations for each run) for the different PSO and CPSO strategies for optimization of the TS fuzzy model. In Table I, the PSO and CPSO approaches optimize 12 parameters (decision variables), e.g., antecedent of two rules (six centers and six spreads of Gaussian membership functions), and in Table II, this optimization method tunes 18 parameters (9 centers and 9 spreads).

As indicated in Tables I and II, the results of the optimized TS fuzzy model present precision and provide an appropriate experimental mathematical model for the yo-yo motion system. The TS fuzzy model presents a black-box model of a nonlinear yo-yo system, with adequate treatment of the nonlinearities of the dynamic system due to the inherent features of the TS fuzzy model, which deal with complex processes.

For the case study of the TS fuzzy-model optimization with two rules, there is a consistent performance pattern across all the CPSO approaches. The CPSO approaches present better results in relation to the mean fitness than does the classical PSO1 algorithm. The CPSO5 shows the best performance with fitness $R^2_h = 0.9331$. However, the CPSO2 and CPSO3 provide results very close to those of the CPSO5. The CPSO3 presents the best mean, and the CPSO6 has significant values with a small standard deviation in convergence and also the best minimum fitness of the approaches tested here.

In general, for the case study of optimization of the TS fuzzy model with three rules, the PSO1 and CPSO approaches presented similar results. However, the CPSO2 is the best with $R^2_h = 0.9357$. This result is slightly better than the case of the TS fuzzy model with two rules. The CPSO5 is superior in terms of mean, minimum, and standard deviations to the other tested PSO1 and CPSO approaches in this particular case.

Tables I and II indicate that, for two and three rules, all the results of the TS fuzzy model showed multiple correlation indexes of best results higher than 0.9230, which are necessary and therefore appropriate for nonlinear identification and controller-design applications. The best results shown in Figs. 8 and 9 represent the TS design with two and three rules, respectively. Fig. 10 shows the membership functions of the best result of two cases using CPSO2 with three rules.

**VI. CONCLUSION**

In multimodal problems such as the TS fuzzy design, the classical PSO tends to suffer from premature convergence. This
is due to a decrease of diversity in the search space, which leads to a fitness stagnation in the optimization process.

In this paper, alternative optimization methods, called the CPSO approaches combined with the GK algorithm, for the premise part of the design of the TS fuzzy models are analyzed and compared. These approaches are the PSO algorithms based on Zaslavskii chaotic sequences in the design of $w$, $c_1$, and $c_2$ parameters.

The most striking feature of chaos is the unpredictability of the future despite a deterministic time evolution, a feature that is useful in the design of stochastic optimization methods. The proposed CPSO approaches deal with the maintenance of the diversity of particle populations of classical PSO for preventing premature convergence.

The experimental results showed that the TS fuzzy model with the CPSO approaches and the GK algorithm presented successful results due to precision in the case study of the prediction of the nonlinear dynamics of a yo-yo motion system.

All CPSO approaches improve the performance and convergence of classical PSO for the optimization of the premise part of the TS fuzzy model. The proposed CPSO approaches perform better in terms of mean fitness than the classical PSO1.

For the case study of the TS fuzzy-model optimization with two rules, it is interesting to note that the CPSO5 presents the best result of fitness but worse in terms of mean convergence than the CPSO2 and CPSO3. For the case study with three rules, CPSO5 outperforms the other tested approaches in terms of mean-fitness results.

However, the precision, computational complexity, and orders of the input vectors of the TS fuzzy model must be analyzed in detail in future works.

As prospects for future works are linked to the yo-yo motion system design, there is a possibility of assessing a comparison of the CPSO approaches with other chaotic sequences in the PSO design reported in recent literature [75]–[77] in multivariable nonlinear identification problems.
referees for their valuable comments which improved the initial manuscript.

REFERENCES


M. S. Tavazoei and M. Haeri, “Comparison of different one-dimensional

K. E. Parsopoulos, E. I. Papageorgiou, P. P. Groumpos, and M. N. Vrahatis, “A first study of fuzzy cognitive maps learning using par-

D. E. Gustafson and W. C. Kessel, “Fuzzy clustering with a fuzzy covari-

R. Babuška, Fuzzy Modeling for Control, International Series in Intelli-


R. C. Eberhart and J. Kennedy, “A new optimizer using particle swarm
to...
Leandro dos Santos Coelho received the B.S. degree in computer science and the B.S. degree in electrical engineering from the Federal University of Santa Maria, Santa Maria, Brazil, in 1994 and 2000, respectively. He earned the M.S. and Ph.D. degrees in computer science and electrical engineering from the Federal University of Santa Catarina, Florianópolis, Brazil, in 1997 and 2000, respectively.

He is currently an Associate Professor of industrial and systems engineering with the Pontifical Catholic University of Paraná, Curitiba-PR, Brazil. He has published several papers on computational intelligence applied to control systems and engineering and is a Coauthor with M. Jamshidi, R. A. Krohling, and P. Fleming of the book *Robust Control Systems with Genetic Algorithms* (CRC Press, 2002). His research interests are electrical power systems, computational intelligence, nonlinear identification, optimization methods, quantum computation, chaos theory, and advanced control systems.


Bruno Meirelles Herrera received the B.S. in computer engineering from the Pontifical Catholic University of Paraná, Curitiba-PR, Brazil, in 2004. He is currently working toward the M.S. degree in industrial and systems engineering at the Pontifical Catholic University of Paraná, Curitiba-PR, Brazil.

His research interests include hardware design, optimization, and quantum computation.