Available online at www.sciencedirect.com
ScienceDirect
Applied Soft Computing 8 (2008) 1354–1364
Applied Soft Computing
www.elsevier.com/locate/asoc

Particle swarm approaches using Lozi map chaotic sequences to fuzzy modelling of an experimental thermal-vacuum system

Ernesto Araujo [a,b,*], Leandro dos S. Coelho [c]

[a] Instituto Nacional de Pesquisas Espaciais-INPE, Integration and Testing Laboratory-LIT, Cx.P. 515, 12.227-010, Sao Jose dos Campos (SP), Brazil
[b] Universidade Federal de Sao Paulo-UNIFESP, Computer Science/Health Informatics Department-DIS, CEP 04023-062, Sao Paulo (SP), Brazil
[c] Pontifical Catholic University of Parana-PUC/PR, Laboratory of Automation and Systems-LAS/PPGEPS, Cx.P. 16.210, 80.215-901, Curitiba (PR), Brazil

Received 8 September 2006; received in revised form 19 August 2007; accepted 4 October 2007
Available online 30 October 2007

Abstract

Particle Swarm Optimization (PSO) approach intertwined with Lozi map chaotic sequences to obtain Takagi–Sugeno (TS) fuzzy model for representing dynamical behaviours are proposed in this paper. The proposed method is an alternative for nonlinear identification approaches especially when dealing with complex systems that cannot always be modelled using first principles to determine their dynamical behaviour. Since modelling nonlinear systems is normally a difficult task, fuzzy models have been employed in many identification problems due its inherent nonlinear characteristics and simple structure, as well. This proposed chaotic PSO (CPSO) approach is employed here for optimizing the premise part of the IF–THEN rules of TS fuzzy model; for the consequent part, least mean squares technique is used. The proposed method is utilized in an experimental application; a thermal-vacuum system which is employed for space environmental emulation and satellite qualification. Results obtained with a variety of CPSO's are compared with traditional PSO approach. Numerical results indicate that the chaotic PSO approach succeeded in eliciting a TS fuzzy model for this nonlinear and time-delay application.

Keywords: Particle swarm optimization; Fuzzy modelling; Chaotic sequences; Nonlinear identification

1. Introduction

System identification is a procedure to obtain a model from observed data for forecasting and/or analysing the dynamical behaviour of a system. However, finding out these models is generally not a simple task due the fact that most real-life physical systems are, actually, nonlinear in nature. In recent decades, diverse approaches for modelling and identification of dynamical systems have been proposed in the literature, including frequency methods, techniques based on estimates of Wiener and Hammerstein models, bilinear systems, the Volterra model, nonlinear regression, and recursive identification [4,5]. Efforts have been devoted to developing techniques for the identification of nonlinear systems [1–3].

Additionally, nonlinear identification techniques employing fuzzy models have received a great deal of attention and have been employed in many applications in system identification [6–13]. A central feature of fuzzy modelling is that it is based on fuzzy coding of information and operating with fuzzy sets instead of crisp numbers. Takagi–Sugeno (TS) fuzzy model, for instance, exhibits both high nonlinearity and simple structure [14,15]. One of the advantages of TS models when compared with the Mamdani-type fuzzy models is related to its ability of approximating a complex system using a reduced set of fuzzy rules [16].

The identification problem in TS modelling consists of (i) structure identification and (ii) parameter identification. The first one is related to both the determination of the premise part and the consequent part of the production rules. It consists of determining the premise space partition and extracting the number of rules as well as determining the equations of the output. In the other hand, the parameter identification task consists of determining the membership functions, so that a performance measure based on the output errors is minimized.

There are several optimization techniques for extracting fuzzy models by using data, such as artificial neural network (ANN) [37,38], genetic algorithm (GA) [12], clustering approaches [10,13] and so on. In this paper a new TS fuzzy

* Corresponding author at: Instituto Nacional de Pesquisas Espaciais-INPE, Integration and Testing Laboratory-LIT, Cx.P. 515, 12.227-010, Sao Jose dos Campos (SP), Brazil.
E-mail address: ernesto.araujo@lit.inpe.br (E. Araujo).

1568-4946/$ – see front matter
doi:10.1016/j.asoc.2007.10.016

system design employing PSO [24] approach working in conjunction with Lozi map chaotic sequences [25] for figuring out the premise part meanwhile least mean squares is used for the calculus of consequent part of production rules of a TS fuzzy system for nonlinear identification.

The hybrid PSO fuzzy approach working in a synergetic manner presents diverse applications in the field of computational intelligence [17–23]. Particle Swarm Optimization (PSO) approach is a population-based evolutionary algorithm inspired in simulation of social behaviour instead of the survival of the fittest individual. PSO is randomly initialized with a population of individuals (potential solutions). Unlike the most of the evolutionary algorithms, each particle in PSO is also associated with a randomized velocity that "flown" through the problem space.

In PSO, a uniform probability distribution to generate random numbers is used. However, the use of other probability distributions may improve the ability to fine-tuning or even to escape from local optima, such as the distributions presented in [26–29]. In the meantime, it has been proposed the use of the chaotic sequences to generate random numbers to update the velocity equation. Chaos describes the complex behaviour of a nonlinear deterministic system [30]. The application of Lozi map chaotic sequences instead of random sequences with uniform distribution in PSO has been demonstrating a powerful strategy to diversify the PSO population and improve the performance of PSO, preventing premature convergence to local minima [31–34]. All these approaches attempted to improve the performance of the standard PSO, but the amount of parameters of the algorithm to tune remained the same.

This paper addresses the question of obtaining a fuzzy TS model through a Particle Swarm Optimization approach using chaotic sequences by employing data supplied by real-world industrial engineering problem.

An experimental case study using a nonlinear thermal-vacuum system is analyzed by using the proposed approach. Thermal-vacuum systems are employed mainly for emulating space environmental conditions in order to qualify space systems, such as satellites, spacecrafts and so forth. Requirements for the space sector establish that the controlled variable must be the temperature on the specimen surface but the original controller was previously designed to use the temperature on the shroud (set of pipes). Experts and operators use their experience and reasoning for operating such a system and, thus, suppress such a problem. An alternative for avoiding human failures and transform the system automatic is both to mimic the human reasoning and to incorporate feedback control techniques. A fuzzy control approach was designed in order to emulate the human behaviour and thus being able to cope with this sort of nonlinear, time-delay system [35,36]. While succeeding in doing so, this fuzzy control approach was designed through interviews and by try-and-error approach in order to model the knowledge and expertise of these specialists and operators. Alternatively, the fuzzy control design may be established automatically if there is a model for the thermal-vacuum chamber.

The proposed chaotic PSO approach for TS fuzzy modelling is used to represent the dynamical behaviour of the before mentioned thermal-vacuum system and so to explore the effectiveness of PSO approaches using Lozi map chaotic sequences in constructing a good TS fuzzy model for nonlinear identification.

2. Optimal Takagi–Sugeno *fuzzy systems*

2.1. Takagi–Sugeno fuzzy systems

Takagi–Sugeno fuzzy modelling is a powerful practical engineering tool for modelling and control of complex systems. The TS model often provides computational attractive solutions to a wide range of modelling problems introducing a powerful multiple model structure that is capable to approximate nonlinear dynamics, multiple operating modes and significant parameter and structure variations [39].

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 a suitable partition of the premise space.

The essential idea of TS fuzzy model is partitioning the input space into fuzzy areas and the approximation of each area through a linear model in such a way that a global nonlinear model is computed. It is characterized as a set of IF–THEN rules where the consequent part is composed of linear sub-models describing the dynamical behaviour of distinct operational conditions meanwhile the antecedent part is in charge of interpolating these sub-systems. The "IF statements" define the premise part that is featured as linguistic terms while the THEN functions constitute the consequent part of the fuzzy system characterized, but not limited to, as linear polynomial terms. The global model is then obtained by the interpolation between these various local models. This model can be used to approximate a highly nonlinear function through simple structure using a small number of rules [21]. The TS models consist of linguistic IF–THEN rules that can be represented by the following general form:

$$R^{(j)}: \quad \begin{array}{l} \text{IF}\,(z_1\,\text{is}\,A_1^j)\,\text{AND}\cdots\text{AND}\,(z_m\,\text{is}\,A_m^j) \\ \text{THEN}\,y_j = b_0^j + b_1^j x_1^j + b_{q_j}^j x_{q_j}^j \end{array}. \tag{1}$$

The IF statements define the premise part while the THEN functions constitute the consequent part of the fuzzy system; $z = [z_1, \ldots, z_m]^T$, such as $i = 1, \ldots, m$, is the input vector of the premise p, and A_i^j are labels of fuzzy sets. The parameters $u = [u_1^j, \ldots, u_{q_j}^j]^T$ represents the input vector to the consequent part of $R^{(j)}$ that comprising q_j terms; $y_j = y_j(u^j)$ denotes the jth rule output which is a linear polynomial of the consequent input terms u_i^j, and $w^j = [w_0^j, \ldots, w_{q_j}^j]^T$ are the polynomial coefficients that form the consequent parameter set. Each linguistic label A_i^j is associated with a Gaussian membership function, $\mu_{A_i^j}(z_i)$, described in (2) where m_{ij} and σ_{ij} are, respectively, the centers (mean value) and the spreads (standard deviations) of the Gaussian membership function:

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

The union of all these parameters formulates the set of premise parameters. The firing strength of rule $R^{(j)}$ represents its excitation level and it is given by:

$$u_j(\underline{z}) = \mu_{A_1^j}(z_1) \cdot \mu_{A_2^j}(z_2) \cdots \mu_{A_m^j}(z_m). \quad (3)$$

The fuzzy sets pertaining to a rule form a fuzzy region (cluster) within the premise space, $A_1^j \times A_2^j \times \cdots \times A_m^j$, with a membership distribution described by Eq. (3). Given the input vectors z and u^j, $j = 1, \ldots, M$, the final output of the fuzzy system is inferred by taking the weighted average of the local outputs, $g_j(\underline{u}_j)$, that is given by:

$$y = \sum_{j=1}^{M} v_j(\underline{z}) \cdot y_j(\underline{u}^j), \quad (4)$$

where M denotes the number of rules and $v_j(z)$ is the normalized firing strength of $R^{(j)}$ which is defined as:

$$v_j(\underline{z}) = \frac{\mu_j(\underline{z})}{\sum_{j=1}^{M} \mu_j(\underline{z})}. \quad (5)$$

The structure identification of TS system is obtained through PSO for premise part optimization while the consequent part optimization is accomplished by batch least mean squares method (pseudo-inversion method) [4].

2.2. Particle Swarm Optimization for TS fuzzy modelling

Particle Swarm Optimization, originally developed by Kennedy and Eberhart in 1995, is a population-based swarm algorithm [40,41]. Similarly to genetic algorithms [42], an evolutionary algorithm approach, PSO is an optimization tool based on a population where the position of each member/particle is a potential solution to an analyzed problem. Each particle is associated to a randomized velocity that moves throughout the problem space.

One advantage of PSO over genetic algorithms is that this method does not have operators, i.e., crossover and mutation. Moreover, PSO does not implement the survival of the fittest individuals; instead, it implements the simulation of social behaviour.

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. The synchrony of the behaviour of the swarm was seen as an effort to keep an optimal distance between them.

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 [25].

Each particle in 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 *pbest*. Another "*best*" value that is tracked by the global version of the particle swarm optimizer is the overall best value. Its location, called *gbest*, is obtained indeed by any particle in the population. The past best position and the entire best overall position of the group are employed to minimize (maximize) the solution. The PSO concept consists of, at each time step, changing the velocity (acceleration) of each particle flying toward its *pbest* and *gbest* locations (global version of 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 shown in Fig. 1. It is worth mentioning that the terms r_1^n and r_2^n are generated in the classical approach for PSO by using a probability density function in the interval [0, 1]. In what

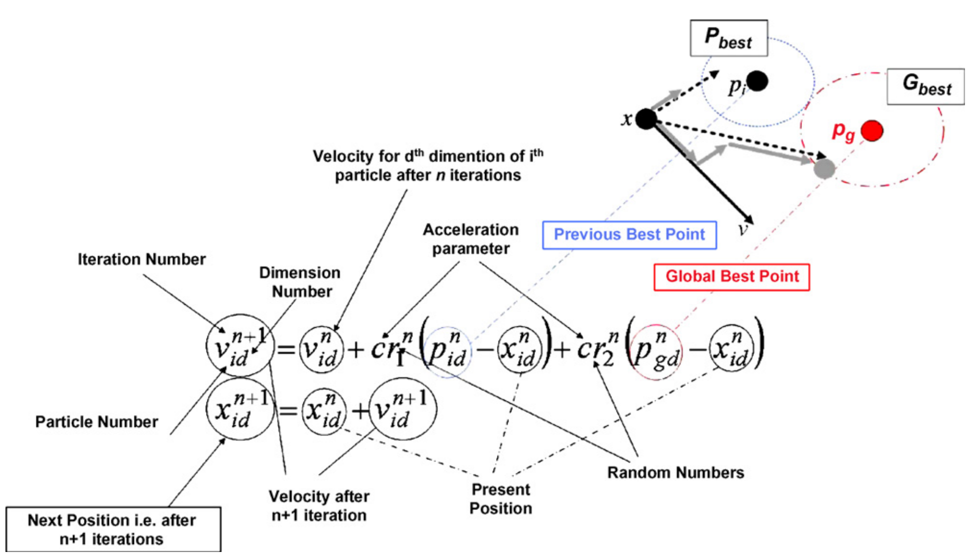

Fig. 1. Geometric view for PSO algorithm.

follows there are the steps for the proposed chaotic PSO approach when those terms are substitute for c_1 and c_2, respectively:

(i) Initialize a population (array) of particles with random positions and velocities in the n-dimensional problem space using uniform probability distribution function.
(ii) For each particle, evaluate its fitness value.
(iii) Compare each particle's fitness with the particle's *pbest*. If current value is better than *pbest*, then set *pbest* value equal to the current value and the *pbest* location equal to the current location in n-dimensional space.
(iv) Compare the fitness with the population's overall previous best. If current value is better than *gbest*, then reset *gbest* to the current particle's array index and value.
(v) Change the velocity, v_i, and position, x_i, of the particle according to Eqs. (6) and (7), respectively:

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot ud_i(t) \cdot (p_i(t) - x_i(t)) + c_2 \cdot Ud(t) \cdot (p_g(t) - x_i(t)) \quad (6)$$

$$x_i(t+1) = x_i(t) + \Delta t \cdot v_i(t+1) \quad (7)$$

(vi) Return to step (ii) until a stop criterion is achieved. Usually it is chosen a pre-established good fitness value or a maximum number of iterations (generations).

In this approach $x_i = [x_{i1}, x_{i2}, \ldots, x_{in}]^{\mathrm{T}}$ stands for the position; $v_i = [v_{i1}, v_{i2}, \ldots, v_{in}]^{\mathrm{T}}$ for the velocity of the ith particle; and $p_i = [p_{i1}, p_{i2}, \ldots, p_{in}]^{\mathrm{T}}$ represents the best previous position of the ith particle (the position giving the best fitness value). The first part in Eq. (6) is associated to the momentum of the particle; the other part is related to the "cognition", which represents the independent thinking of the particle itself. Eq. (7) represents the position update, according to its previous position and velocity, considering $\Delta t = 1$.

The inertia weight, w, corresponds to the degree of the momentum of the particles is responsible for dynamically adjusting the velocity of the particles as proposed by Shi and Eberhart [43]. This parameter is accountable for balancing between local and global search, consequently, needing less or more iterations for the algorithm to converge. A small value of inertia weight implies in a local search; a high one leads to a global search, yet with a high computational cost. It is worth mentioning that if it is interested in reducing the influence of past velocities during the optimization process, linear decreasing inertia function may also be used.

The index, g, represents the best particle among all the particles in the group. Variables $ud_i(t)$ and $Ud_i(t)$ are two random numbers generated using uniform probability distribution functions in the range [0,1].

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 *pbest* and *gbest*. The constriction coefficient method, as suggested by Clerc and Kennedy [44], is used in this paper. According to it the velocity equation is updated by:

$$v_i(t+1) = K \cdot [v_i(t) + c_1 \cdot ud_i(t) \cdot (p_i(t) - x_i(t)) + c_2 \cdot Ud_i(t) \cdot (p_g(t) - x_i(t))]. \quad (8)$$

The constriction coefficient, K, is given by:

$$K = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|} \quad (9)$$

where $\varphi = c_1 + c_2, \varphi > 4$. Usually, φ is set to 4.1 ($c_1 = c_2 = 2.05$) and, then, the constriction coefficient K is 0.729.

2.3. New approaches of PSO using Lozi map chaotic sequences for TS fuzzy modelling

Chaos theory studies pertinent phenomenon to dynamical systems are nonlinear and that present complex behaviour to be treat mathematically. The theory of the chaos studies the unexpected phenomenon apparently, in the search of hidden standards and simple laws that conduct the complex behaviours. This field became effective after 1960s when computers had started to possess reasonable graphical capacity and of processing, giving to the physicists and mathematicians the power to discover answers for basic questions of the science in general way.

Chaos theory is recognized as very useful in many engineering applications, such as complex behaviours in electric circuits, behaviour of stock exchange and economy, telecommunications, chemistry, physics, control systems, and others. Chaos is a phenomenon that can appear in solutions for nonlinear differential equations. An essential feature of chaotic systems is that small changes in the parameters or the starting values for the data lead to vastly different future behaviours, such as stable fixed points, periodic oscillations, bifurcations, and ergodicity. These behaviours can be analyzed based on Lyapunov exponents and the attractor theory [25,30].

Optimization algorithms based on the chaos theory are search methodologies that differ from any of the existing traditional stochastic optimization techniques. Due to the non-repetition of chaos, it can carry out overall searches in the solution space at higher velocities when compared to stochastic ergodic searches, which has its computing based on probabilities. In this context, the literature contains several optimization algorithms using chaotic sequences for solving design problems [45–57].

Optimization based on chaotic sequences is promising when used to provide diversity in populations of PSO approaches. Different types of equations have been considered in literature for applications in optimization methods. The logistic equation and other equations, such as logistic map, Lozi map, sinusoidal iterator, Chua's oscillator, Lorenz system, and Ikeda map, have been adopted instead of random ones and very interesting results [25,30,46].

The parameters $ud_i(t)$ and $Ud_i(t)$ (6) are important parameters that affect the PSO convergence. This paper provides new approaches introducing chaotic mapping with ergodicity, irregularity and the stochastic property in PSO to improve the global convergence in substitution of parameters $ud_i(t)$ and $Ud_i(t)$. The use of chaotic sequences in PSO can be helpful to escape more easily from local minima than the traditional PSO methods.

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 Lozi's map [46,58].

The Lozi map is a simplification of the Henon map [59] and it admits strange attractors. This chaotic map involves also non-differentiable functions which difficult the modelling of the associate time series. The Lozi map is given by:

$$y_1(t+1) = 1 - a|y_1(t)| + y_2(t) \tag{10}$$

$$y_2(t+1) = by_1(t) \tag{11}$$

The parameters used in this work are $a = 1.7$ and $b = 0.5$ as suggested in [46]. These new PSO approaches combined with chaotic sequences (CPSO) based on Lozi's map are described as follows:

Approach 1—CPSO1: Parameter $ud_i(t)$ (8) is modified to:

$$v_i(t+1) = K \cdot [v_i(t) + c_1 \cdot y_{2,i}(t) \cdot (p_i(t) - x_i(t)) + c_2 \cdot Ud_i(t) \cdot (p_g(t) - x_i(t))] \tag{12}$$

where $y_{2,i}(t)$ is given by Lozi's map scaled within 0 and 1.

Approach 2—CPSO2: Parameter $Ud_i(t)$ (8) is modified to:

$$v_i(t+1) = K \cdot [v_i(t) + c_1 \cdot ud_i(t) \cdot (p_i(t) - x_i(t)) + c_2 \cdot y_{2,i}(t) \cdot (p_g(t) - x_i(t))] \tag{13}$$

where $y_{2,i}(t)$ is given by Lozi's map scaled within 0 and 1.

Approach 3—CPSO3: Parameter $ud_i(t)$ and $Ud_i(t)$ (8) is modified to:

$$v_i(t+1) = K \cdot [v_i(t) + c_1 \cdot y_{2,i}(t) \cdot (p_i(t) - x_i(t)) + c_2 \cdot y_{3,i}(t) \cdot (p_g(t) - x_i(t))] \tag{14}$$

where $y_{2,i}(t)$ and $y_{3,i}(t)$ are given by Lozi's map scaled within 0 and 1. The CPSO3 has utilization probability p of Eq. (14) else the classical Eq. (8) is utilized.

3. Thermal-vacuum system identification

3.1. Problem statement

Finding out a model for representing dynamical behaviour is of particular importance when dealing with nonlinear, time-delay thermal-vacuum chambers used for satellite qualification. Thermal-vacuum chambers are used to reproduce as close as possible environmental conditions of expected post-launch environments which satellites will experience during their in-flight life, that is, their operational life [60,61]. Once in space, satellites are exposed, but not limited, to sunshine, Albedo radiation, earth radiation, shadow/eclipse conditions, and earthshine infrared.

Thermal-vacuum systems consist of a chamber, a shroud (set of pipes) which heats or cools off the environment, and auxiliary equipment [62]. In the thermal-vacuum system used at the Brazilian National Institute for Space Research (INPE), the original controller was designed to control the temperature on the shroud [21].

A thermal-vacuum system employed to reproduce expected space post-launch environments conditions for qualifying space systems consists of a chamber, a set of tubes (shroud) – from where heat and cold are transmitted through radiation – and other auxiliary devices (Fig. 2). The operation of the thermal shroud is achieved by means of a re-circulating, dense, and gaseous nitrogen (GN2) system. Resistance type heaters provide heat as required. Cooling the circulating gas stream is accomplished by spraying liquid nitrogen (LN2) into the circuit [62].

This system is not only nonlinear [63] but presents time-delay, is time-varying, and works in diverse operational conditions defined by various levels of temperature (set points)

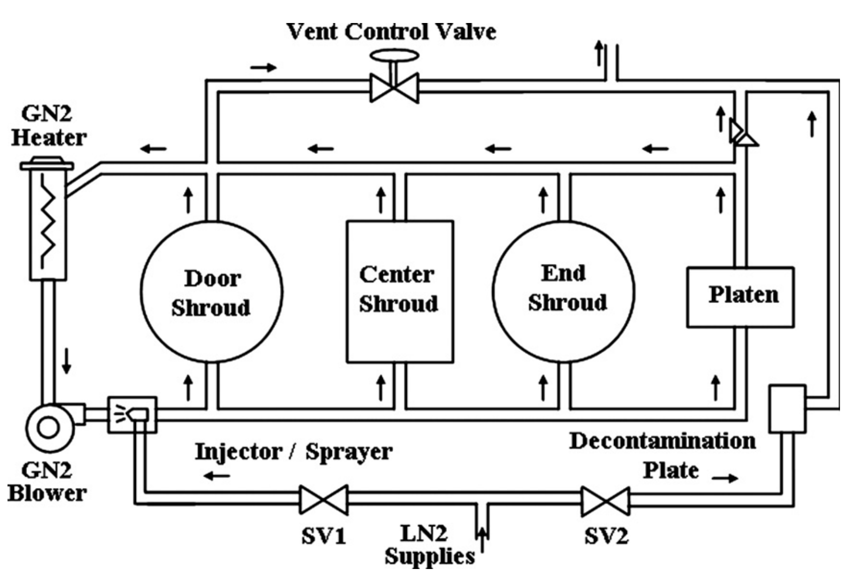

Fig. 2. Schematic diagram for thermal-vacuum chamber.

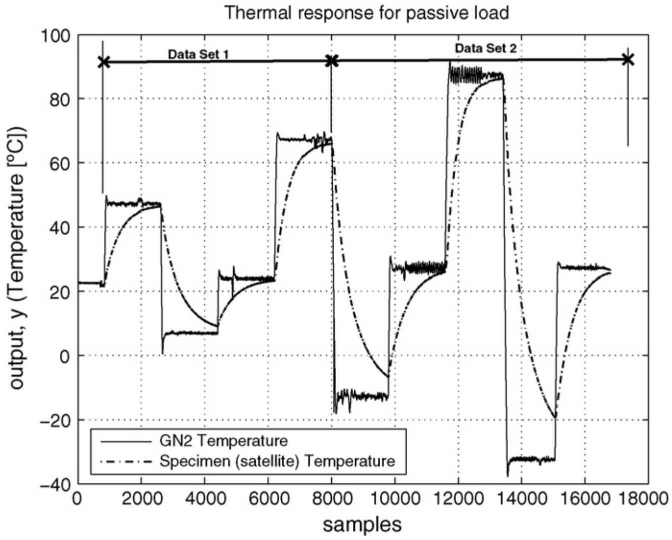

Fig. 3. Temperature on the satellite and in the gas used for nonlinear identification.

used during the test. For illustrate these characteristics, a dynamical response for a passive load is shown in Fig. 3. Continuous and dashed lines represent input and output data, respectively. It is possible to notice, for instance, the chamber-load set presents as many heating and cooling rates as the operational conditions.

In a previous work, a fuzzy controller was designed to work as a supervisory-control mechanism in helping specialists to operate the thermal-vacuum chamber. Results and the suggested Fuzzy Reference Gain-Scheduling (FRGS) control approach are available in [35,36,63]. An alternative to improve the fuzzy controller performance is to find out a fuzzy model for representing the thermal dynamical behaviour and thus to use it to tune the fuzzy control system. Here, PSO and fuzzy approaches work in a complementary and synergistic manner by using experimental data.

3.2. Identification of TS fuzzy model

Identification of dynamic systems can be performed with a series–parallel or parallel model. Series–parallel structure is the type of mathematical model adopted for identification (one-step ahead forecasting) of thermal-vacuum system when using the hybrid piece wise, gain-scheduling PSO-TS modelling approach as shown in Fig. 4.

Assume that there is a TS fuzzy model that produces an estimated output, $\hat{y}(k)$, based on an input $u(k)$ and the noise contribution, $n(k)$, present in the modelled process. The estimated TS fuzzy model output based on PSO, $\hat{y}(k)$, used for computing the minimum square error when compared with the actual output, $y(k)$, was computed by using one-step ahead forecasting. Denote n_y, n_u, and n_n as the time maximum lags of the model output, control input, and noise, respectively. Depending on the time-lagged inputs that are used for the TS fuzzy model, different configurations of models can be used. In this work, a NARX (Nonlinear Auto Regressive with eXogenous inputs) model was adopted, given by:

$$\hat{y}(k) = f_{\mathrm{TS}}(u(k-1), \ldots, u(k-n_u), \ldots, y(k-1), \ldots, y(k-n_y), \theta) \tag{15}$$

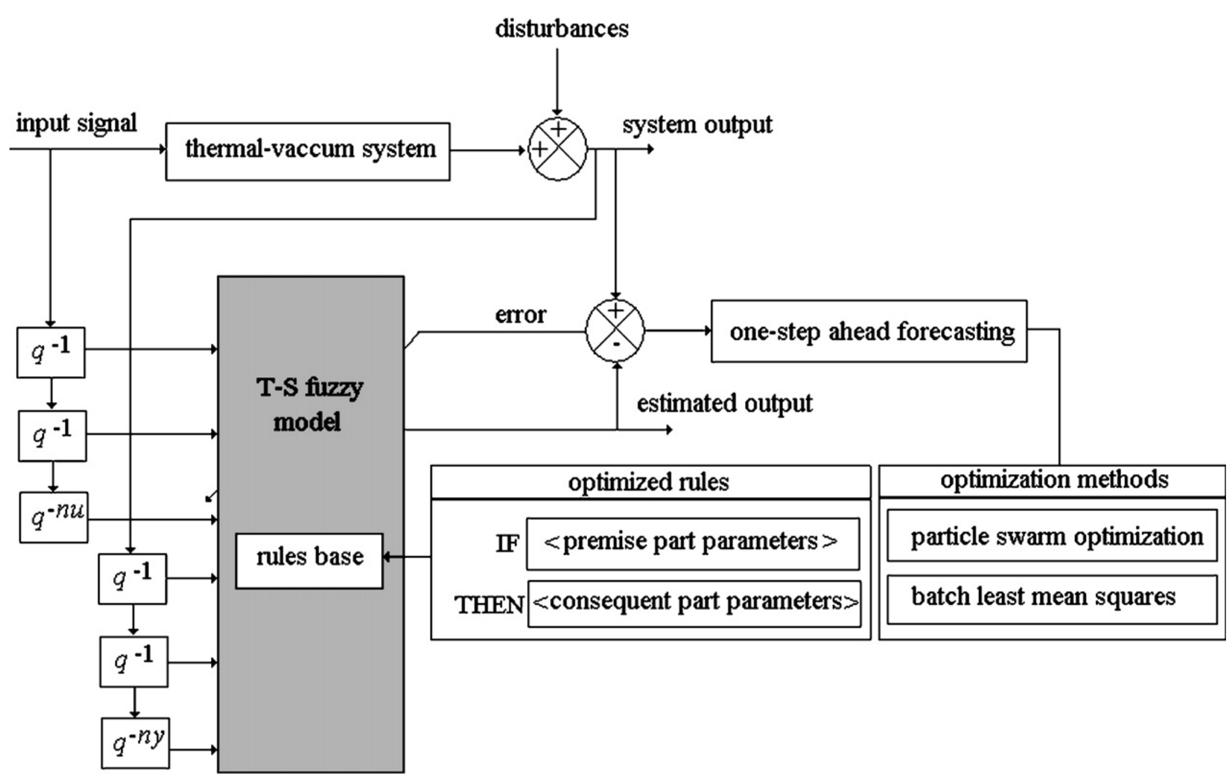

Fig. 4. One-step ahead forecasting using TS fuzzy model using PSO approaches and batch mean least squares.

Table 1
Parameters for PSO and CPSO application in TS fuzzy model

Parameter	Selection
Number of particles	5
Number of iterations (t_{max})[a]	20
Inertia weight setup, K	0.729
Cognitive component, c_1	2.05
Social component, c_2	2.05

[a] Generations = stopping criterion.

where the unknown nonlinear function f_{TS} is the TS fuzzy model of the system, k represents the kth instant of time, and the vector θ representing the parameters in the consequent of the TS rules.

One of the most important tasks in building an efficient forecasting model based in TS fuzzy model is the selection of the relevant input variables. The input selection problem can be stated as follows: among a large set of potential input candidates, choose those variables that highly affect the model output. Unfortunately, there is no systematic procedure, currently available, which can be followed in all circumstances. In this work, input selection is heuristically performed. The inputs of TS fuzzy system are process output and control input signals of reduced order with $n_y = 2$, $n_u = 1$, and $n_n = 0$. In this work, the three vectors of input for the TS fuzzy system are $[u(k-1); y(k-1); y(k-2)]$ and the model output is $y(k)$.

The first part of data shown in Fig. 3 was employed to elicit the fuzzy model through traditional PSO and CPSO. The first 200 initial data samples were discharged because they have constant values and would interfere in the identification process. The remaining data were used to validate the results. In this case, 3400 samples in training (estimation) phase of TS system using PSO approaches, and other 500 in validation (test or generalization) phase of TS.

Although, PSO allows to extract the number of rules and to determine the premise and consequent elements, here this method is applied to obtain membership functions and thus to determine the premise space partition.

Setting up this parameter as three production rules, PSO needs to deal with a vector of particles positions and velocity whose elements are nine centres and three spreads of a Gaussian function, respectively, core and support of membership functions. In this case, the spread of Gaussian membership function adopted for each input of vectors $[u(k-1); y(k-1); y(k-2)]$ of TS fuzzy model is the same.

Table 2
Summary of the results (best of 30 independent runs with 20 generations) for different PSO and CPSO approaches of TS fuzzy system design in training phase

Optimization method	Fitness maximization: $R^2_{training}$				
	Best	Mean	Minimum	Median	Standard deviation
PSO (Eqs. (8) and (9))	**0.9999**	0.9908	0.9321	0.9974	0.0208
CPSO1	**0.9999**	0.9936	0.9770	0.9991	0.0087
CPSO2	**0.9999**	0.9938	0.9769	0.9960	0.0071
CPSO3, $p = 0.1$	**0.9999**	0.9796	0.8881	0.9991	0.0419
CPSO3, $p = 0.2$	**0.9999**	0.9770	0.8670	0.9942	0.0430
CPSO3, $p = 0.3$	**0.9999**	0.9948	0.9846	0.9954	0.0056
CPSO3, $p = 0.4$	**0.9999**	**0.9986**	**0.9906**	**0.9999**	**0.0029**
CPSO3, $p = 0.5$	**0.9999**	0.9953	0.9778	0.9993	0.0077
CPSO3, $p = 0.6$	**0.9999**	0.9937	0.9455	0.9996	0.0170
CPSO3, $p = 0.7$	**0.9999**	0.9816	0.8246	0.9996	0.0552
CPSO3, $p = 0.8$	**0.9999**	0.9867	0.9344	0.9997	0.0220
CPSO3, $p = 0.9$	**0.9999**	0.9941	0.9789	0.9981	0.0073
CPSO3, $p = 1.0$	**0.9999**	0.9858	0.9542	0.9947	0.0165

Table 3
Summary of the results (best of 30 independent runs with 20 generations) for different PSO and CPSO approaches of TS fuzzy system design in validation phase

Optimization method	$R^2_{validation}$ of optimized TS model				
	Best	Mean	Minimum	Median	Standard deviation
PSO (Eqs. (8) and (9))	0.9990	0.8929	0.2671	0.9626	0.2237
CPSO1	**0.9999**	0.9180	0.6771	0.9791	0.1177
CPSO2	0.9994	0.9246	0.5708	0.9833	0.1317
CPSO3, $p = 0.1$	**0.9999**	0.8020	0.000.0	0.9893	0.4001
CPSO3, $p = 0.2$	0.9994	0.7606	0.0000	0.9077	0.3581
CPSO3, $p = 0.3$	**0.9999**	0.9411	0.7927	0.97	0.0755
CPSO3, $p = 0.4$	**0.9999**	**0.9873**	**0.9119**	**0.9996**	**0.0273**
CPSO3, $p = 0.5$	0.9998	0.9420	0.8220	0.9875	0.0814
CPSO3, $p = 0.6$	**0.9999**	0.9739	0.8798	0.9943	0.0405
CPSO3, $p = 0.7$	**0.9999**	0.8924	0.0000	0.9962	0.3189
CPSO3, $p = 0.8$	**0.9999**	0.8980	0.6091	0.9915	0.1565
CPSO3, $p = 0.9$	**0.9999**	0.9421	0.6551	0.9718	0.1055
CPSO3, $p = 1.0$	0.9995	0.8221	0.1499	0.9798	0.2859

The system identification by TS fuzzy model is appropriate if a suitable performance index is available according to the necessities of users. Among a population of potential solution to a problem, every particle of PSO has a fitness value for expressing appropriate optimization result. The function representing this quality measure employs the position of all particles, x_i, which is calculated after each iteration.

The performance criterion (fitness function) chosen for evaluate the relationship between the real output and the estimate output during the optimization process (maximization problem) was the *Pearson multiple correlation coefficient index*. This coefficient represents the R^2 of training phase of TS fuzzy model conducted by R^2 as given by:

$$R^2_{\text{training}} = 1 - \frac{\sum_{k=1}^{0.5\text{Na}} [y(k) - \hat{y}(k)]^2}{\sum_{k=1}^{0.5\text{Na}} [y(k) - \bar{y}]^2} \tag{16}$$

where Na is the total number of samples evaluated, and $\bar{y}(k)$ is the system real output. When $R(\cdot)^2$ is close to unit a sufficient accurate model for the measured data of the system is found. A R^2 between 0.9 and 1.0 is suitable for applications in identification and model-based control [64]. In this context, the performance evaluation of validation phase of optimized TS fuzzy system is realized by:

$$R^2_{\text{validation}} = 1 - \frac{\sum_{k=0.5\text{Na}+1}^{\text{Na}} [y(k) - \hat{y}(k)]^2}{\sum_{k=0.5\text{Na}+1}^{\text{Na}} [y(k) - \bar{y}]^2}. \tag{17}$$

The main parameters deeply related to the success of PSO for tuning the premise part of TS fuzzy model are: (i) the number of particles (size of population), (ii) the initial position and velocity of particles, (iii) the cognitive and social components (c_1 and c_2), (iv) the form of inertia factor updating, and (v) stopping criterion, $t_{\max}$ (adopted $t_{\max}$ = 100 iterations). One of advantage of this technique is that the initial population of particles is randomly generated through a uniform probability distribution function. The sufficient number of particles for this application was setup as 10. The main parameters of PSO and CPSO approaches are shown in Table 1.

Each PSO approach was implemented in Matlab (MathWorks). To illustrate the effectiveness of the TS fuzzy model several simulations were performed. All the programs were run on a 3.8 GHz Pentium IV processor with 2 GB of RAM. In each case study, 30 independent runs were made for each of the optimization methods involving five different initial trial solutions (number of particles) for each optimization method. The mean CPU time of each run is approximately 5.18 min.

The results (best of 30 independent runs with 20 generations) for different PSO and CPSO strategies of TS fuzzy system design in training and validation phases are presented in Tables 2 and 3, respectively.

The best mean fitness value, R^2_{training}, when using 20 iterations (30 runs) was obtained by CPSO3 with $p = 0.4$. The results using CPSO3 with $p = 0.4$ present small standard

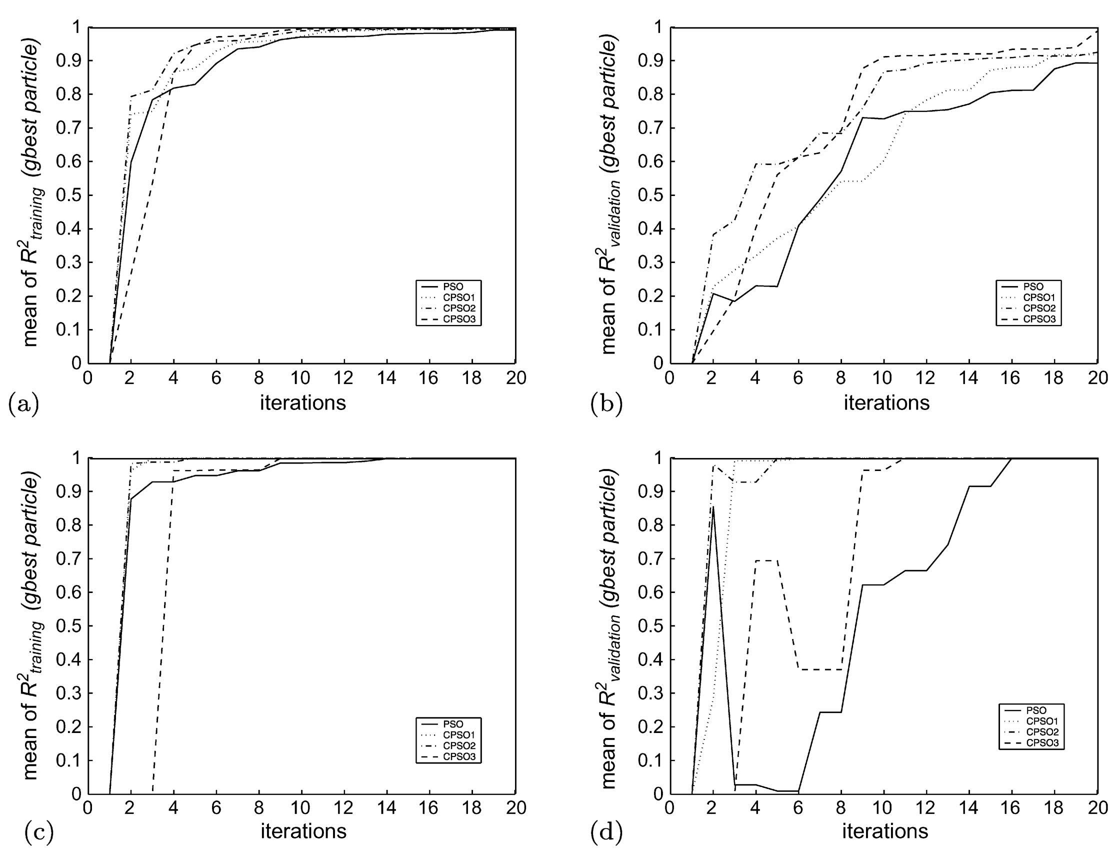

Fig. 5. Best response (in training phase) obtained through resulting TS model using PSO. (a) Mean value of fitness function, R^2_{training}, for *gbest* particle (30 runs). (b) Mean value of $R^2_{\text{validation}}$ for *gbest* particle when using with best R^2_{training} (30 runs). (c) Best fitness function, R^2_{training} (30 runs). (d) Evaluation function, $R^2_{\text{validation}}$ for *gbest* particle when using best fitness function, R^2_{training} (30 runs).

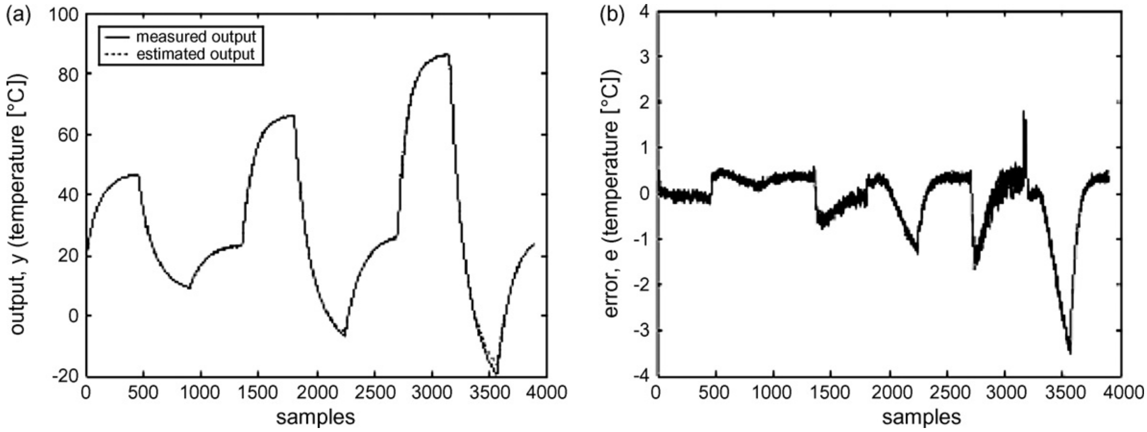

Fig. 6. Best response (in training phase) obtained through resulting TS model using PSO.

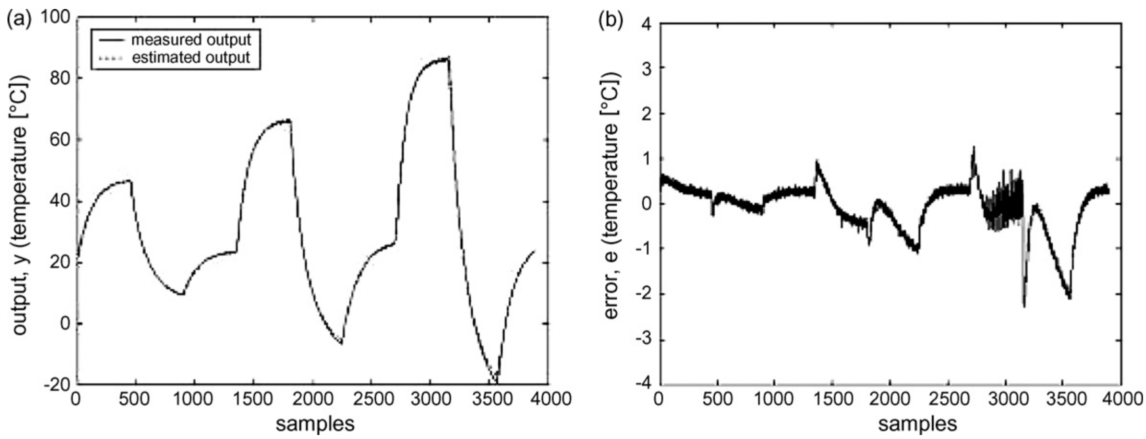

Fig. 7. Best response (in training phase) obtained through resulting TS model using CPSO1.

deviation and also the best R^2_{training} minimum value as it is presented in Table 2.

All optimized TS fuzzy model fitted the training data with best R^2_{training} values equal 0.99999. However, in the validation phase the all PSO and CPSO approaches present best results of $R^2_{\text{validation}}$ greater than 0.9990. Although PSO-TS and CPSO(1–3)-TS fuzzy models achieved a good approximation for experimental data, it has not generalized well to new data. This feature can be observed by $R^2_{\text{validation}}$ minimum values in Table 3. Convergence data of PSO and CPSO approaches are presented in Fig. 5(a–d). In these figures, the mean and best results using PSO, CPSO1, CPSO2, and CPSO3 with $p = 0.4$ are shown.

Continuous and dashed lines represent measured and simulated outputs in results presented in Figs. 6–9. Experimental results had shown that the hybrid TS fuzzy system and

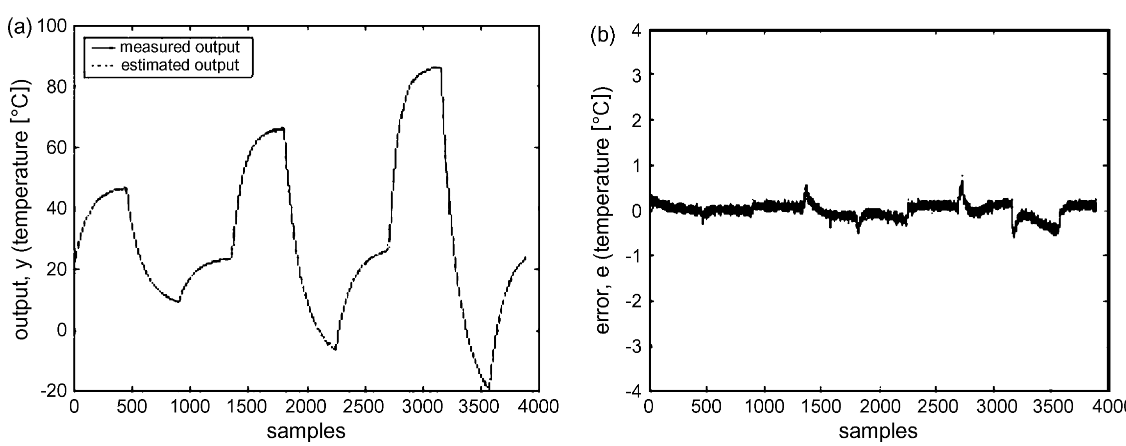

Fig. 8. Best response (in training phase) obtained through resulting TS model using CPSO2.

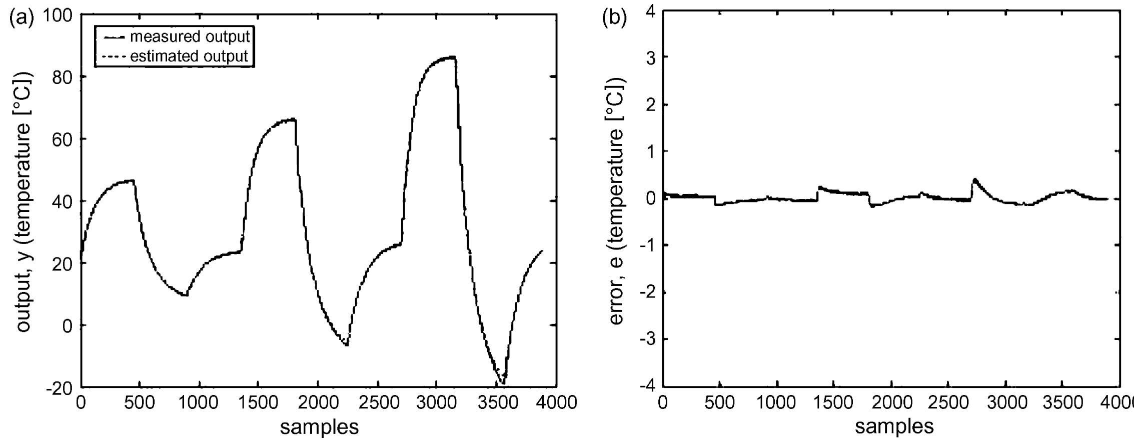

Fig. 9. Best response (in training phase) obtained through resulting TS model using CPSO3 with $p = 0.4$.

PSO approaches presented successful results due precision in predicting nonlinear dynamics.

4. Conclusion and future research

Particle Swarm Optimization is a population-based swarm intelligence algorithm driven by the simulation of a social psychological metaphor instead of the survival of the fittest individual. Inspired by the swarm intelligence theory and chaos concepts, this work presents the use of PSO and CPSO approaches and pseudo-inverse method in TS fuzzy system design. In this context, experimental tests using PSO and CPSO to find out a TS fuzzy model for a thermal-vacuum system were realized.

The most striking feature of chaos is the unpredictability of the future despite a deterministic time evolution, a feature that is important in the design of efficient stochastic optimization methods. The proposed CPSO approaches using Lozi's chaotic map deal with the maintenance of the diversity of particle populations of classical PSO for preventing premature convergence. The elicited fuzzy model with only three membership functions determining the premise space partition demonstrated its effectiveness in emulating the time response for the thermal-vacuum system. Future work must be carried out to verify the influence of other parameters in obtaining the model. For example, the number of membership functions may be increased, and since there is large amount of data other step ahead forecasting modes may be exploited.

Future research will include the hybridization of the PSO and CPSO and other local search methods, such as simulated annealing and quasi-Newton methods. This approach, combining of the efficient global search of PSO and the effectiveness of deterministic local search, possibly will give good results for identification of thermal-vacuum system.

References

[1] M. Gevers, A personal view on the development of system identification, in: Proceedings of the 13th IFAC Symposium on System Identification, Rotterdam, The Netherlands, (2003), pp. 773–783.

[2] R.K. Pearson, Selecting nonlinear model structures for computer control, J. Process Control 13 (2003) 1–26.

[3] D.F. Specht, A general regression neural network, IEEE Trans. Neural Networks 1 (6) (1991) 568–576.

[4] L. Ljung, System Identification: Theory for the User, Prentice-Hall, New York, NY, USA, 1987.

[5] R. Haber, H. Unbehauen, Structure identification of nonlinear dynamic systems: a survey on input/output approaches, Automatica 26 (4) (1990) 651–677.

[6] M.L. Hadjili, V. Wertz, Takagi–Sugeno fuzzy modelling incorporating input variables selection, IEEE Trans. Fuzzy Syst. 10 (6) (2002) 728–742.

[7] W. Pedrycz, F. Gomide, An Introduction to Fuzzy Sets: Analysis and Design, MIT Press, Cambridge, MA, 1998.

[8] J. Abonyi, R. Babuska, M.A. Botto, F. Szeifert, L. Nagy, Identification and control of nonlinear systems using fuzzy Hammerstein models, Ind. Eng. Chem. Res. 39 (2000) 4302–4314.

[9] M.R. Emami, A.A. Goldenberg, I.B. Tiirksen, Fuzzy-logic control of dynamical systems, Eng. Appl. Artif. Intell. 13 (2000) 47–69.

[10] I.S. Akkizidis, G.N. Roberts, Fuzzy clustering methods for identifying and modelling of non-linear control strategies, Proc. Inst. Mech. Eng. Part I 215 (2001) 437–452.

[11] F. Montoya, A. Cipriano, M. Ramos, A new identification method for use in nonlinear prediction, J. Intell. Fuzzy Syst. 10 (2001) 131–137.

[12] B. Wu, X. Yu, Fuzzy modelling and identification with genetic algorithm based learning, Fuzzy Sets Syst. 113 (2000) 351–365.

[13] X. Jiang, H. Adeli, Fuzzy clustering approach for accurate embedding dimension identification in chaotic time series, Integr. Comput. Aided Eng. 10 (2003) 287–302.

[14] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, IEEE Trans. Syst. Man Cybern. 15 (1) (1985) 116–132.

[15] M. Sugeno, G.T. Kang, Structure identification of fuzzy model, Fuzzy Sets Syst. 28 (1988) 15–33.

[16] E.H. Mandani, S. Assilian, An experiment in linguistic synthesis with a fuzzy logic controller, Int. J. Man-Mach. Studies 7 (1) (1975) 1–13.

[17] Application of particle swarm optimization algorithm for weighted fuzzy rule-based system, in: Proceedings of the Annual Conference of IEEE Industrial Electronics Society, IECON, Busan, Korea, pp. 2188–2191.

[18] C.-F. Juang, Y.-C. Liou, TSK-type recurrent fuzzy network design by the hybrid of genetic algorithm and particle swarm optimization, in: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, vol. 3, The Hague, The Netherlands, (2004), pp. 2314–2318.

[19] M. Ma, C.-G. Zhou, L.-B. Zhang, Q.-S. Dou, Automatic fuzzy rule extraction based on particle swarm optimization, in: Proceedings of the International Conference on Machine Learning and Cybernetics, vol. 4, Shanghai, China, (2004), pp. 2242–2245.

[20] A.A.A. Esmin, A.R. Aoki, G.L. Torres, Particle swarm optimization for fuzzy membership functions optimization, in: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, vol. 3, Hammamet, Tunisia, (2002), p. 6.

[21] R. Marinke, J.E. Araujo, L.S. Coelho, I. Matko, Particle swarm optimization (PSO) applied to fuzzy modeling in a thermal-vacuum system, in: Proceedings of the 5th International Conference on Hybrid Intelligent Systems, Rio de Janeiro, Brazil, (2005), pp. 67–72.
[22] E. Araujo, L.S. Coelho, Piecewise fuzzy reference-driven Takagi–Sugeno modelling based on particle swarm optimisation, in: N. Nedjah, L. Mourelle (Eds.), Systems Engineering using Swarm Particle Optimisation, Nova Science Publishers, NY, USA, 2006, pp. 85–112.
[23] A.P. Ghoshal, Optimizations of PID gains by particle swarm optimizations in fuzzy based automatic generation control, Electr. Power Syst. Res. 72 (3) (2004) 203–212.
[24] J.F. Kennedy, R.C. Eberhart, R.C. Shi, Swarm Intelligence, Morgan Kaufmann Pub., San Francisco, CA, 2001.
[25] H.-O. Peitgen, H. Jiirgens, D. Saupe, Chaos and Fractals: New Frontiers of Science, second ed., Springer, New York, NY, 2004.
[26] R.A. Krohling, Gaussian swarm: a novel particle swarm optimization algorithm, in: Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems (CIS), Singapore, (2004), pp. 372–376.
[27] B.R. Secrest, G.B. Lamont, Visualizing particle swarm optimization—Gaussian particle swarm optimization, in: Proceedings of the IEEE Swarm Intelligence Symposium, Indianapolis, IN, USA, (2003), pp. 198–204.
[28] L.S. Coelho, R.A. Krohling, Predictive controller tuning using modified particle swarm optimisation based on Cauchy and Gaussian distributions, in: Soft Computing: Methodologies and Applications, Springer Engineering series in Advances in Soft Computing, 2005, pp. 287–298.
[29] R.A. Krohling, L.S. Coelho, PSO-E: particle swarm with exponential distribution, in: IEEE World Congress on Computational Intelligence, Proceedings of the IEEE Congress on Evolutionary Computation, Vancouver, Canada, 2006.
[30] S.H. Strogatz, Nonlinear Dynamics and Chaos, Perseus Publishing, Massachussetts, 2000.
[31] B. Liu, L. Wang, Y.-H. Jin, F. Tang, D.-X. Huang, Improved particle swarm optimization combined with chaos, Chaos Solitons Fractals 25 (2005) 1261–1271.
[32] J. Chuanwen, E. Bompard, A hybrid method of chaotic particle swarm optimization and linear interior for reactive power optimization, Math. Comput. Simul. 68 (2005) 57–65.
[33] J. Chuanwen, E. Bompard, A self-adaptive chaotic particle swarm algorithm for short term hydroelectric system scheduling in deregulated environment, Energy Conver. Manage. 46 (2005) 2689–2696.
[34] L.S. Coelho, B.M. Herrera, Fuzzy modeling using chaotic particle swarm approaches applied to a yo-yo motion system, in: IEEE World Computational Intelligence Conference, Proceedings of the IEEE Congress on Fuzzy Systems, Vancouver, Canada, 2006.
[35] J.E. Araujo, S.A. Sandri, E.E.N. Macau, Fuzzy reference gain scheduling control systems, in: Proceedings of the NAFIPS, Atlanta, (2000), pp. 461–464.
[36] J.E. Araujo, S.A. Sandri, E.E.N. Macau, A new class of adaptive fuzzy control system applied in industrial thermal vacuum process, in: Proceedings of the of IEEE-ETFA, vol. 1, France, (2001), pp. 426–431.
[37] M.F. Azeem, M. Hanmandu, N. Ahmad, Structure identification of generalized adaptive neuro-fuzzy inference systems, IEEE Trans. Fuzzy Syst. 11 (5) (2003) 666–681.
[38] D. Kukolj, E. Levi, Identification of complex systems based on neural and Takagi–Sugeno fuzzy model, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 34 (1) (2004) 272–282.
[39] P.P. Angelov, D.P. Filev, An approach to online identification of Takagi–Sugeno fuzzy models, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 34 (1) (2004) 484–498.
[40] R.C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of the International Symposium on Micro Machine and Human Science, Nagoya, Japan, (1995), pp. 39–43.
[41] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, (1995), pp. 1942–1948.
[42] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison Wesley, Reading, MA, 1989.
[43] Y. Shi, R.C. Eberhart, Parameter selection in particle swarm optimizer, in: V.W. Porto, N. Saravan, D. Waagen, A.E. Eiben (Eds.), Proceedings of the Annual Conference on Evolutionary Programming, Springer-Verlag, Berlin, 1998, pp. 591–601.
[44] M. Clerc, J. Kennedy, The particle swarm—explosion, stability, and convergence in a multidimensional complex space, IEEE Trans. Evol. Comput. 6 (1) (2002) 58–73.
[45] B. Li, W. Jiang, Optimizing complex functions by chaos search, Cybern. Syst. 29 (4) (1998) 409–419.
[46] R. Caponetto, L. Fortuna, S. Fazzino, M.G. Xibilia, Chaotic sequences to improve the performance of evolutionary algorithms, IEEE Trans. Evol. Comput. 7 (3) (2003) 289–304.
[47] G. Zilong, W. Sun'an, Z. Jian, A novel immune evolutionary algorithm incorporating chaos optimization, Pattern Recog. Lett. 27 (1) (2006) 2–8.
[48] D.F. Wang, P. Han, Q. Ren, Chaos optimization variable arguments PID controller and its application to main steam pressure regulating system, in: Proceedings of the International Conference on Machine Learning and Cybernetics, Beijing, China, (2002), pp. 2116–2120.
[49] L.S. Coelho, V.C. Mariani, Chaotic differential evolution applied electromagnetic optimization, in: Proceedings of the 12th Biennial IEEE Conference on Electromagnetic Field Computation (CEFC), Miami, FL, USA, 2006.
[50] X. Zou, M. Wang, A. Zhou, B. Mckay, Evolutionary optimization based on chaotic sequence in dynamic environments, in: Proceedings of the IEEE International Conference on Networking, Sensing & Control, Taipei, Taiwan, (2004), pp. 1364–1369.
[51] G.-C. Liao, T.-P. Tsao, Application of a fuzzy neural network combined with a chaos genetic algorithm and simulated annealing to short-term load forecasting, IEEE Trans. Evol. Comput. 10 (2006) 330–340.
[52] L.S. Coelho, H.S. Lopes, Supply chain optimization using chaotic differential evolution method, in: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Taipei, Taiwan, 2006.
[53] M. Hasegawa, T. Ikeguchi, K. Aihara, K. Itoh, A novel chaotic search for quadratic assignment problems, Eur. J. Oper. Res. 139 (2002) 543–556.
[54] L. Wang, K. Smith, On chaotic simulated annealing, IEEE Trans. Neural Networks 9 (4) (1998) 716–718.
[55] K. Masuda, E. Aiyoshi, Global optimization method using chaos of discrete inertial gradient dynamics, in: SICE Annual Conference in Fukui, Fukui University, Japan, (2003), pp. 3353–3358.
[56] K. Wang, X. Ye, W. Feng, Study of the fusion among fuzzy, chaos and neural network, in: Proceedings of the IEEE International Conference on Intelligent Mechatronics and Automation, Chengdu, China, (2004), pp. 500–504.
[57] L.S. Coelho, V.C. Mariani, Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect, IEEE Trans. Power Syst. 21 (2) (2006) 989–996.
[58] K.T. Alligood, T.D. Sauer, J.A. Yorke, Chaos: An Introduction to Dynamical Systems, Springer, London, 1996.
[59] M. Henon, A two-dimensional mapping with a strange attractor, Commun. Math. Phys. 50 (1976) 69–77.
[60] J.T. Garner, Satellite Control—A Comprehensive Approach, John Wiley & Sons Ltd. and Praxis Publishing Ltd., Chichester, 1996.
[61] D.G. Gilmore, Satellite Thermal Control Handbook, The Aerospace Corporation Press, California, 1994.
[62] High Vacuum Systems Inc., Operations and Maintenance Manual, Thermal Vacuum System with Thermally Conditioned Platen for Brazilian Space Research Institute, 1987.
[63] J.E. Araujo, K.H. Kienitz, Adaptive reference-driven decision-making process, in: Proceedings of the FUZZ-IEEE, vol. 1, St. Louis, USA, (2003), pp. 452–457.
[64] B. Schweizer, H. Xie, Y.C. Lee, Fuzzy logic models for ranking process effects, IEEE Trans. Fuzzy Syst. 5 (4) (1997) 545–555.