Particle Swarm Optimization with Turbulence (PSOT) applied to Thermal-Vacuum Modelling

Ernesto Araujo, *Member, IEEE*, Fernando P. A. Araujo, Jose C. Becceneri and Haroldo F. Campos Velho

***Abstract*—Particle Swarm Optimization with Turbulence (PSOT) is, in this paper, applied to find out fuzzy models to represent dynamic behavior of space systems that lie underneath the space qualification process. In optimization area, each minimal improvement in results may represents a maximal, precious meaning and PSOT improve the performance of the established Particle Swarm Optimization (PSO) by introducing a slight variation, which simulates the action of an atmosphere turbulence to escape from local minima. This paper trades off the results of original PSO presented in a previous paper and PSOT both intertwined with Takagi-Sugeno (TS) fuzzy modeling dealing with experimental results of a thermal-vacuum system. Particle Swarm Optimization with turbulence has demonstrated to be a good alternative by taking into account the velocity of convergence to better solution and the total optimization time in generating dynamical models to the proposed system.**

I. Introduction

Efforts have been dispensed to improve the capacity of modeling dynamic systems. With high complexity characterizing many of these systems, techniques of nonlinear modeling are used to approximate a model to the given dynamic system.

Optimization algorithms are into these nonlinear techniques. To solve a problem it starts from an initial configuration, which is a problem solution candidate. Based on a quality parameter (usually a function) the initial configuration is modified in order to minimize (maximize) this quality, i.e., cost function.

In areas of space research and industry, the thermal-vacuum system is very important to represent the dynamic behavior of space systems outside the earth atmosphere. Some of these systems are, only to mention few, communication satellites, spacecrafts etc. To better plan space missions, tests and simulations into laboratories have to describe with more fidelity as possible the conditions that these systems will meet outside earth atmosphere.

Other advantages of identifying a model that well represents this real world are the ability to detect loss of vacuum, presence of unknown heat sources or sinks, training of thermal-vacuum operators, development of a supervisor decision-support system for helping to control the whole operation, checking the instantaneous operation or even the operator's behavior or performance, and, ultimately design an automatic control for the whole system [1], [2], [3].

Through a database of thermal-vacuum behavior this paper employs the Takagi-Sugeno (TS) fuzzy-modeling technique to describe space systems. Fuzzy systems has being an alternative to model nonlinear systems [1], [4], [5], [6] since they are universal approximators [7], [8]. Due to that it is used here to represent the main dynamic characteristics of space systems that lie underneath space qualification processes. The system identification of the TS fuzzy-modeling is accomplished by an optimization approach named Particle Swarm Optimization with Turbulence (PSOT), which is employed to improve input-output values represented in premises (IF) and consequent (THEN) parts of system description. The fuzzy modeling of a thermal-vacuum system is, then, tuned by applying an adaptation of Particle Swarm Optimization (PSO) algorithm to approximate this model to the best solution describing the actual system. The PSO is a stochastic optimization technique that mimics the sociological, collective behavior inspired in flock of birds (swarm). First, introduced by Kennedy and Eberhart [9], [10], this search mechanism in which each particle (individual) interact locally with their environment and globally with other particles to find out the minimum (maximum) in a optimization space and so coherent and convergent global solution [11], [12], [13], [14].

Results show that this hybrid, modified approach has also succeeded on simulating a real system and can be applied to support operators to decide what is the best control action for conducting a thermal-vacuum qualification testing.

II. Takagi-Sugeno Fuzzy Modeling

Proposed by Lofti Zadeh [15], fuzzy logic can be understood as a generalization of the Boolean logic. Fuzzy logic is based on the syllogism and the fuzzy set theory. While the first is well known, the latter needs explanation. A fuzzy set is composed of elements that can assume infinite values between true and false. Fuzzy sets are represented by

Ernesto Araujo is with Integration and Testing Laboratory (LIT) and Space Engineering and Technology (ETE), at Instituto Nacional de Pesquisas Espaciais (INPE), Av. Astronautas, 1758, 12.227-010, São José dos Campos, SP, Brazil; with Health Informatics Department (DIS), at Universidade Federal de Sao Paulo (UNIFESP), R. Botucatu, 862, 04023-062, São Paulo, SP, Brazil, (email: ernesto.araujo@unifesp.br); with the Hospital Municipal Dr. Jose de Carvalho Florence (HMJCF), Av. Saigiro Nakamura, 800, 04023-062, São José dos Campos, SP, Brazil; and Assoc. Paulista para o Desenvolvimento da Medicina (SPDM), R. Napoleo de Barros, 715, 04024-002, São Paulo, SP, Brazil.

Fernando P. A. Araujo is with Applied Intelligent Systems Lab., Computer Sci. Dept. (DCC), Universidade Federal de Lavras (UFLA), Lavras, MG, Brazil, (email: fernandoaraujo.ti@gmail.com)

Jose C. Becceneri, Computer Sci. and Applied Math. Associated Lab. (LAC), Instituto Nacional de Pesquisas Espaciais (INPE), Av. Astronautas, 1758, 12.227-010, São José dos Campos, SP, Brazil, (email: becceneri@lac.inpe.br)

Haroldo F. Campos Velho, Computer Sci. and Applied Math. Associated Lab. (LAC), Instituto Nacional de Pesquisas Espaciais (INPE), Av. Astronautas, 1758, 12.227-010, São José dos Campos, SP, Brazil, (email: haroldo@lac.inpe.br)

978-1-4244-3597-5/09/$25.00 ©2009 IEEE

membership functions $\mu_M(x) \to [0, 1]$ where each element inside a fuzzy set, $M = \{x \in X\}$, assumes a degree of conformity between 0 and 1. Due to that, fuzzy logic allows to represent by propositions, P = $< x_1$ is $M >$, the inherent imperfection (uncertainty, imprecision, vagueness) in human descriptions of nature phenomena. It provides a conceptual base to build up a sort of reasoning that is neither exact nor inexact, i.e., an approximate reasoning. It can better represent verbal or mental modeling by using qualitative expressions, inherent of human communication, into computational values. Concerned to knowledge or cognitive model, it is termed as deductive model, as well.

This technique also assumes an important role when attempting to represent dynamical systems. It represents a *nonlinear mapping* from input space vectors, X_n, to a scalar output space, Y, in the form, $f : X_n \to Y$, such that X_n and Y are universe of discourses that define the input-output space, $X_n \times Y$, and an associated fuzzy inference mechanism. For instance, when using fuzzy sets to part the input universe of discourse, $A = \{x \in X_1\}$, into an output universe of discourse, $B = \{y \in Y\}$, it is represented as $f : A \to B$. These items of input space will be compared to the linguistic terms (i.e. labels) using IF-THEN rules.

Takagi-Sugeno (T-S) fuzzy model [16] is an alternative for fuzzy system representation when dealing with data. It is able to approximate highly nonlinear functions and exhibits simple structure by using a small number of implication rules [16], [17]. T-S model divides the input space in the same manner that Mamdani fuzzy system [18] but aims to approximate structure of the local models to a linear model in the consequent of the rule. Another advantage of employing this approach is that it reduces the problem complexity by restricting the number of rules that will be processed in each subsystem, and then interpolating them to obtain the global model.

The T-S fuzzy model is characterized as a set of IF-THEN rules where the consequent part are linear sub-models describing the dynamical behavior of distinct operational conditions meanwhile the antecedent part is in charge of interpolating these sub-systems. This model can be represented as follows:

$$\begin{array}{l} R_j : \text{IF } x_1 \text{ is } A_{1j} \text{ AND} \ldots \text{AND } x_m \text{ is } A_{mj} \\ \quad\ \ \text{THEN } y_j = f(\cdot) \end{array} \tag{1}$$

The “THEN functions” constitutes the consequent part of the j-th rule of the fuzzy system that is characterized, but not limited to, as a linear polynomial, $y_j = b_0^j + b_1^j u_1^j + \ldots + b_{q_j}^j u_{q_j}^j$. The j-th rule output, $y_j = f(\mathbf{u}, \mathbf{b}^j)$, is function of the consequent input vector, $\mathbf{u} = [u_1^j, \ldots, u_{q_j}^j]^T$, comprising q_j terms and the polynomial coefficient vector, $\mathbf{b} = [b_1^j, \ldots, b_{q_j}^j]^T$, that compose the consequent parameter set.

The global model is, then, obtained by the interpolation between these various local models:

$$y = \sum_{j=1}^{N} h_j(\mathbf{x}) y_j(\mathbf{u}^j) \ , \tag{2}$$

where N denotes the maximal number of rules and $h_j(z)$ is the normalized firing strength of $R^(j)$, defined as:

$$h_j(\mathbf{x}) = \frac{\mu_j(\mathbf{x})}{\sum_{j=1}^{M} \mu_j(\mathbf{x})} \ , \tag{3}$$

with:

$$\mu_j(\mathbf{x}) = \mu_{A_1^j}(x_1) \mu_{A_2^j}(x_2) \ldots \mu_{A_m^j}(x_m) \ , \tag{4}$$

for linguistic labels, A_i^j, associated to a membership function.

III. Particle Swarm Optimization with Turbulence

In order to differentiate the Particle Swarm Optimization with Turbulence (PSOT) from the original Particle Swarm Optimization (PSO) the latter is presented first.

A. Original Particle Swarm Optimization

Particle Swarm Optimization (PSO) technique was suggested by Kennedy and Eberhart [9], [10] as a bioinspired meta-heuristic that simulates a social behavior and interactions of population's members [11].

Like other computational approaches that simulate movement of organisms, the main idea concerning PSO is to manipulate the distance between each member in the solution space. To simulate this behavior, each member (i.e., particle) is described by its position and a velocity of displacement in the solution space that is related to a potential solution to an analyzed problem.

The simulation process has a pre-defined iteration number, where in each iteration the particle's position is updated based on the best position of all iterations (i.e. best local, $pbest$) and on best position of population (i.e. best global, $gbest$).

The first position and velocity adopted by each particle is generated randomly using uniform probability distribution function. Then, for each particle the algorithm proceeds as follows:

(i) evaluate its fitness value;

(ii) 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;

(iii) 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;

(iv) change the velocity, v_i, and position of the particle, x_i, respectively according to eq. (5) and (6):

$$\begin{aligned} v_i(t+1) = {} & w \cdot v_i(t) + \ldots \\ & c_1 \cdot ud_i(t) \cdot (p_i(t) - x_i(t)) + \ldots \\ & c_2 \cdot Ud(t) \cdot (p_g(t) - x_i(t)) \end{aligned} \tag{5}$$

$$x_i(t+1) = x_i(t) + \Delta t \cdot v_i(t+1) \tag{6}$$

(v) Return to step (i) until a stop criterion is met, usually a sufficiently good fitness or a maximum number of iterations (generations).

The inertia weight, w, in (5), given as:

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} iter \ , \tag{7}$$

helps in controlling the movement of each particle. Based on experimentation [19] [20] it describes the influence of inertia weight in particle's exploration as follows:

- $w < 0,8$: induces a local search. This type of exploration aims to optimize the solution, searching close around the actual point;
- $w > 1,2$: induces a global search. This exploration aims to find a further and better solution in the space by searching in unexplored areas;
- $0,8 < w < 1,2$: these intermediary values represent a balance between global and local search. In general, values of this interval are often the best ones to solve problems.

In turn, the constants c_1 e c_2 are always positive and represents a social component which limits the velocity, respectively, to local best ($pbest$) direction and to global best direction ($gbest$).

B. Turbulence for improving Particle Swarm Optimization

In the original representation of PSO, only the interactions among particles are considered to update position values. In order to improve the performance of the original algorithm, a slight variation, which simulates the action of atmosphere turbulence, was introduced, becoming the process of escaping from local minima more efficient [21], [22]. It modifies randomly and independently the particle's pathway aiming approximation of this model to the real world. The turbulence value can be obtained by the following equation:

$$i_{turb} = (w + c_1 + c_2) \cdot G(x_g best) \cdot (1/(T - t)) \tag{8}$$

where T is total number of iterations, t is current iteration.

Others ways can be used in order to calculate this value. It could have as many dimensions as distinct variables considered by PSO. To include turbulence in PSO, it is used

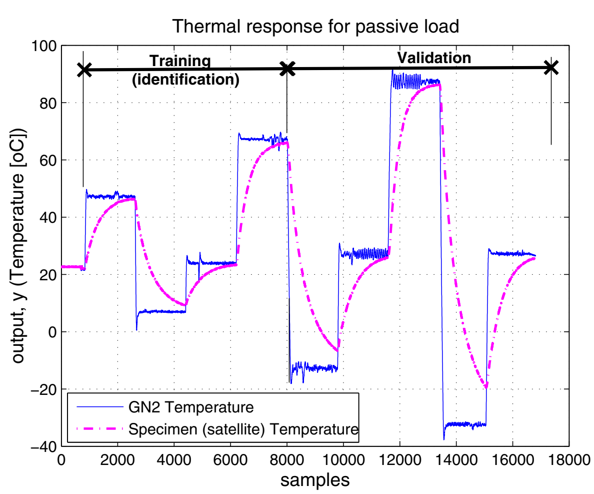

Fig. 1. Thermal response for passive load

TABLE I
PARAMETERS FOR PSOT APPLICATION, ACCORDING TO [23], [24]

Parameter	Value
Number of particles	5
Number of iterations	30
Number of experiments	30
Learning rate	$c_1 = c_2 = 2.05$
Optimization function	Maximization of Pearson multiple correlation coefficient

uniform probability (this value is defined by user) to active, or no, the turbulence in the computing of the speed in each dimension. For instance, the case where turbulence is applied in some dimension is given as:

$$v_{id} = v_{id} \cdot i_{turb} \cdot rand_i \tag{9}$$

In so doing, this equation replaces eq. (5) in the velocity update step of the PSO algorithm.

It is worth mentioning that most turbulence representation adopt the Osborn Reynolds assumption on turbulence, where the flow is split into a *deterministic* component (similar to the laminar flow, but obtained by the Reynolds average) plus a *stochastic* component. However, if the average flow is zero, therefore the turbulent velocity follows a law according to Eq. (9).

IV. QUALIFYING SPACE SYSTEMS BY USING PSOT-TS FUZZY MODELING IN THERMAL-VACUUM SYSTEMS

The thermal-vacuum system aims to reproduce space conditions for space qualification. In order to achieve that, it is composed by a chamber, a set of tubes (shroud), and other auxiliary devices. The function of shroud is to regulate temperature into the chamber. This operation is executed by a system of re-circulating gaseous (GN2) and liquid nitrogen (LG2) providing cooling while resistances provide heat [25]. This system presents time-delay, is time-varying, and works in diverse operational conditions defined by various levels of temperature (set points) used during the test, but is not nonlinear [23], [24]. It is illustrated through experimental data in which a dynamical response for passive load is depicted in Fig. 1. The solid line represents the temperature of the gas inside the shroud and corresponds to a typical temperature set up for thermal-vacuum environmental simulation for satellite qualification. The dashed line, in turn, is the temperature on the satellite and represents its thermal response when submitted to step values of temperature.

Takagi-Sugeno (TS) fuzzy system is widely accepted as a tool for designing and analysis of fuzzy systems. It became a powerful tool for nonlinear modeling/identification and control of dynamical systems mainly if intertwined with Particle Swarm Optimization [3]. The parameters employed to run the PSOT algorithm to find a TS fuzzy model for thermal-vacuum systems are given in Table I, aiming performance benchmark [23], [24].

TABLE II
FITNESS EVOLUTION FOR PSOT-TS FUZZY SYSTEMS

Iterations	PSO	PSOT, % Turbulence																			
		5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0.9337	0.8538	0.1917	0	0.9618	0.9298	0	0	0	0.4052	0.2688	0.5126	0.5126	0.2614	0.5292	0.4762	0.2499	0.2499	0.2309	0
3	0.3409	0.9337	0.9817	0.1917	0.2212	0.9618	0.9298	0	0.4469	0	0.4052	0.2688	0.5126	0.5126	0.2614	0.8834	0.4762	0.2864	0.2499	0.2718	0
4	0.3409	0.9337	0.9817	0.1917	0.2212	0.9618	0.9298	0	0.4469	0	0.4052	0.4586	0.5126	0.8707	0.4253	0.8834	0.4762	0.3307	0.2499	0.2718	0
5	0.8534	0.9337	0.9817	0.1917	0.2854	0.9618	0.9298	0	0.4469	0	0.4052	0.5138	0.7598	0.8707	0.4253	0.8834	0.9753	0.3307	0.3728	0.2718	0
6	0.8534	0.9337	0.9817	0.1917	0.2854	0.9618	0.9298	0	0.4469	0.2420	0.4052	0.5138	0.7598	0.8707	0.4253	0.8834	0.9753	0.3307	0.3728	0.2718	0.5400
7	0.8534	0.9337	0.9817	0.1917	0.4993	0.9618	0.9298	0	0.4469	0.2420	0.4052	0.7494	0.7598	0.8707	0.8120	0.8834	0.9753	0.3713	0.3728	0.2718	0.5797
8	0.8534	0.9337	0.9817	0.1917	0.4993	0.9618	0.9298	0	0.4469	0.2420	0.4052	0.7494	0.7598	0.8707	0.8120	0.8834	0.9753	0.4074	0.8702	0.2718	0.5797
9	0.9896	0.9337	0.9817	0.9562	0.4993	0.9618	0.9298	0	0.4469	0.2420	0.4052	0.7494	0.9427	0.8835	0.8120	0.9536	0.9753	0.4074	0.8741	0.2718	0.5797
10	0.9896	0.9870	0.9817	0.9562	0.4993	0.9618	0.9298	0	0.4469	0.8131	0.4052	0.7494	0.9427	0.8835	0.8120	0.9536	0.9753	0.4990	0.8741	0.5931	0.6780
11	0.9896	0.9870	0.9817	0.9562	0.4993	0.9618	0.9298	0.7158	0.4469	0.8131	0.4052	0.7494	0.9616	0.9790	0.8120	0.9536	0.9753	0.4990	0.8741	0.5931	0.6780
12	0.9998	0.9883	0.9817	0.9562	0.4993	0.9618	0.9298	0.7158	0.7457	0.8131	0.4052	0.7494	0.9616	0.9790	0.8120	0.9536	0.9753	0.9799	0.8741	0.5931	0.6780
13	0.9998	0.9883	0.9817	0.9562	0.4993	0.9618	0.9298	0.7158	0.7457	0.8131	0.9231	0.7494	0.9616	0.9790	0.8120	0.9536	0.9753	0.9799	0.8741	0.5931	0.6780
14	0.9998	0.9883	0.9817	0.9562	0.4993	0.9618	0.9298	0.7158	0.7457	0.8394	0.9231	0.7494	0.9616	0.9790	0.8120	0.9665	0.9753	0.9799	0.8741	0.5931	0.9687
15	0.9999	0.9883	0.9936	0.9562	0.4993	0.9618	0.9820	0.7158	0.8129	0.8394	0.9231	0.7850	0.9799	0.9790	0.8120	0.9789	0.9753	0.9999	0.8741	0.5931	0.9687
16	0.9999	0.9883	0.9936	0.9562	0.4993	0.9618	0.9820	0.7158	0.8129	0.8394	0.9231	0.7850	0.9799	0.9790	0.9202	0.9789	0.9753	0.9999	0.8741	0.7536	0.9687
17	0.9999	0.9883	0.9936	0.9562	0.4993	0.9618	0.9831	0.7158	0.8129	0.8394	0.9231	0.7850	0.9799	0.9790	0.9577	0.9822	0.9753	0.9999	0.8741	0.7594	0.9687
18	0.9999	0.9883	0.9936	0.9562	0.4993	0.9618	0.9831	0.7158	0.8444	0.8394	0.9902	0.7850	0.9991	0.9790	0.9577	0.9822	0.9753	0.9999	0.8741	0.9797	0.9687
19	0.9999	0.9883	0.9936	0.9562	0.4993	0.9618	0.9831	0.7158	0.8444	0.8394	0.9902	0.7850	0.9991	0.9790	0.9577	0.9849	0.9829	0.9999	0.9904	0.9797	0.9687
20	0.9999	0.9883	0.9936	0.9562	0.4993	0.9618	0.9831	0.7158	0.8444	0.9877	0.9902	0.7850	0.9991	0.9790	0.9577	0.9849	0.9893	0.9999	0.9904	0.9797	0.9905
21	0.9999	0.9883	0.9936	0.9562	0.4993	0.9618	0.9831	0.7158	0.8444	0.9877	0.9902	0.7850	0.9991	0.9790	0.9577	0.9849	0.9893	0.9999	0.9904	0.9966	0.9954
22	0.9999	0.9883	0.9936	0.9562	0.4993	0.9618	0.9831	0.7158	0.8444	0.9877	0.9902	0.8584	0.9991	0.9790	0.9631	0.9849	0.9893	0.9999	0.9904	0.9966	0.9954
23	0.9999	0.9883	0.9936	0.9562	0.4993	0.9618	0.9831	0.7158	0.9579	0.9877	0.9902	0.8584	0.9991	0.9790	0.9631	0.9868	0.9981	0.9999	0.9904	0.9983	0.9954
24	0.9999	0.9883	0.9936	0.9562	0.4993	0.9618	0.9831	0.7158	0.9805	0.9877	0.9902	0.8584	0.9991	0.9790	0.9631	0.9868	0.9981	0.9999	0.9904	0.9983	0.9954
25	0.9999	0.9883	0.9936	0.9562	0.4993	0.9618	0.9831	0.8117	0.9805	0.9877	0.9902	0.8584	0.9991	0.9790	0.9631	0.9868	0.9981	0.9999	0.9904	0.9983	0.9976
26	0.9999	0.9883	0.9936	0.9562	0.4993	0.9618	0.9996	0.8117	0.9980	0.9877	0.9902	0.8584	0.9991	0.9790	0.9631	0.9868	0.9989	0.9999	0.9904	0.9983	0.9976
27	0.9999	0.9883	0.9936	0.9562	0.4993	0.9618	0.9996	0.8117	0.9980	0.9877	0.9902	0.8584	0.9991	0.9790	0.9631	0.9868	0.9989	0.9999	0.9904	0.9986	0.9976
28	0.9999	0.9883	0.9936	0.9562	0.4993	0.9618	0.9996	0.8117	0.9980	0.9877	0.9902	0.8803	0.9997	0.9790	0.9631	0.9868	0.9995	0.9999	0.9904	0.9986	0.9976
29	0.9999	0.9883	0.9936	0.9562	0.4993	0.9618	0.9996	0.9604	0.9980	0.9877	0.9902	0.8803	0.9997	0.9790	0.9631	0.9868	0.9995	0.9999	0.9904	0.9986	0.9976
30	0.9999	0.9883	0.9936	0.9921	0.4993	0.9618	0.9996	0.9604	0.9980	0.9877	0.9902	0.8803	0.9997	0.9790	0.9783	0.9868	0.9995	0.9999	0.9904	0.9986	0.9976

In order to achieve better results and using universal approximation characteristics of fuzzy modeling when using PSOT, Gaussian membership functions are chosen and set to 3 [23]. In so doing, the number of particles in population is predefined. Here, positions of particles in population are generated randomly using uniform distribution. In order to verify the effectiveness of the modified PSO, 21 experiments for all turbulence randomized algorithm are carried out. These experiments are runned with different random seeds. It intends to improve the confidence interval of experimental results.

An original PSO algorithm and 20 other ones with a variation of turbulence ranging from 5 to 100 with step of 5% are accomplished. Based on those parameters, the fitness evolution in the best iteration of each experiment (each variant of PSO) is presented in Fig. 2 and detailed in Table II.

As it may be noticed, the PSOT with 85% probability of turbulence achieve the same result as original PSO (with no turbulence within) – the best fitness value as 0.999. Nevertheless, PSOT with 5%, 10%, 25%, and 30% probability of turbulence has a much higher speed of convergence than original PSO.

The estimation of PSOT-TS Fuzzy model output, $\hat{y}(k)$, used for computing the minimum square error when compared with the current output, $y(k)$, is computed by using one-step ahead forecasting. To compute the error in the modeling was used the output of PSOT-TS Fuzzy model, $\hat{y}(k)$, as could be seen next:

$$\min \theta = \sum_{k=1}^{N} ||\hat{y}(k) - y(k)|| \qquad (10)$$

and then compared to current output.

The maximization of *Pearson multiple correlation coefficients* is chosen to evaluate the output during the optimization process. This coefficient gives the rate between the variability of two measures (variables) in which one is described by the variability of the other [23]. Suitable correlation factors considered for practical control applications are bigger than 0.9 and lower (or equal to) than 1.0 as presented in [26].

Complementary, the similarity of actual time measures and the estimated ones computed by PSOT with 85% turbulence probability is depicted in Fig. 3. The measured output and estimated output match each other and so, achieving the desired objective established for this paper.

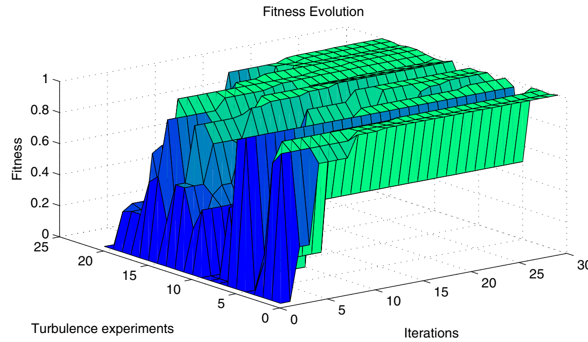

Fig. 2. Simulation Results

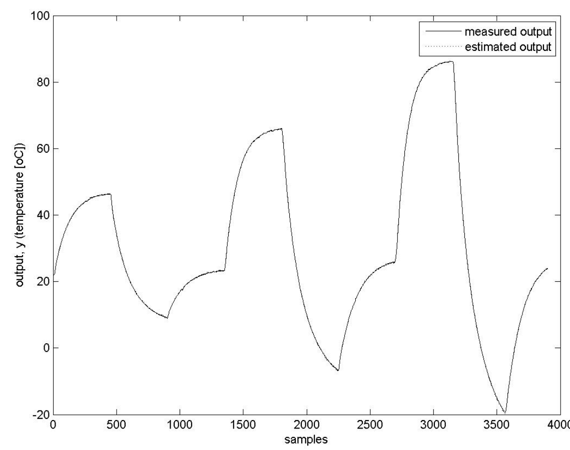

Fig. 3. Real time response for PSO with 85% turbulence probability

V. Conclusion

In this work Particle Swarm Optimization with Turbulence (PSOT) is applied to Takagi-Sugeno fuzzy model optimization. The TS fuzzy model is used to represent and forecast future dynamical behavior of a thermal-vacuum system in space system qualification.

As result, PSOT technique has showed reduced computational cost by using a lower number of iterations to obtain satisfactory results. Through fitness evaluation the PSOT with 25% probability of turbulence needs 86,7% less iterations to achieve best fitness, compared to original, classical PSO, i.e., PSO without turbulence. Other PSO implementation with probability of turbulence (5%, 10%, 25% and 30%) obtained good results with higher convergence speed what may be interesting, for instance, in real-time application. It may be used in some sort of application where performance is required but accuracy is not the main requirement.

Additionally, when comparing fitness, it is possible to notice that PSO obtain the same best fitness value as PSOT with turbulence (85% of probability).

References

[1] E. Araujo, U. Freitas, L. Coelho, E. Macau, and L. Aguirre, "Particle Swarm Optimization (PSO) fuzzy system and NARMAX ERR approach trade-off applied to thermal-vacuum chamber identification," in *ASME Pressure Vessels and Piping/ICPVT-11 Conference*. Vancouver, Canada: IEEE, 2006.

[2] C.-L. Jen and L. Tilwick, "On-line, self-learning, predictive toll for determining payload thermal response," in *Proc. of Space Space Simulation*, 2000, pp. 193–200.

[3] E. Araujo and L. Coelho, “Piecewise fuzzy reference-driven Takagi-Sugeno modelling based on Particle Swarm Optimization (PSO),” in *Systems Engineering using Particle Swarm Optimization*, N. Nedjah and L. de Macedo Mourelle, Eds. Nova Science Publishers Inc., 2007, p. 133.
[4] L. dos S. Coelho and E. Araujo, “Identification of the Henon chaotic map by fuzzy modeling and Nelder-Mead simplex method,” *Chaos, Solitons and Fractals*, 2009, dx.doi.org/10.1016/j.chaos.2008.10.013.
[5] E. Araujo and L. dos S. Coelho, “Particle swarm approaches using lozi map chaotic sequences to fuzzy modeling of an experimental thermal-vacuum system,” *Applied Soft Computing Journal*, vol. 8, pp. 1354–1364, 2008.
[6] E. Araujo and L. Coelho, “Fuzzy model and Particle Swarm Optimization for nonlinear identification of Chuas oscillator,” in *International Conference on Fuzzy Systems*. London: IEEE, 2007.
[7] B. Kosko, “Fuzzy systems as universal approximator,” *IEEE Transactions on Computers*, vol. 43, no. 11, pp. 1329–1333, 1994.
[8] R. Rovatti, “Fuzzy piecewise multilinear and piecewise linear systems as universal approximators in sobolev norms,” *IEEE Transactions on Fuzzy Systems*, vol. 6, no. 2, pp. 235–249, 1998.
[9] R. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” in *Intern. Symposium on Micro Machine and Human Science*, Nagoya, Japan, 1995, pp. 39–43.
[10] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *IEEE Intern. Conf. on Neural Networks*, Perth, Australia, 1995, pp. 1942–1948.
[11] J. F. Kennedy, R. C. Eberhart, and R. C. Shi, *Swarm Intelligence*. Morgan Kaufmann Pub, 2001.
[12] M. Clerc and J. Kennedy, “Particle swarm – explosion, stability, and convergence in a muntidimensional complex space,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.
[13] J. Robinson and Y. Rahmat-Samii, “Particle swarm optimization in electromagnetics,” *IEEE Transactions on Antennas and Propagation*, vol. 52, no. 2, pp. 397–407, 2004.
[14] Y. del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J.-C. Hernandez, and R. G. Harley, “Particle swarm optimization: Basic concepts, variants and applications in power systems,” *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 2, pp. 171–195, 2008.
[15] L. A. Zadeh, “Fuzzy sets,” *Information and Control*, vol. 8, pp. 338–353, 1965.
[16] T. Takagi and M. Sugeno, “Fuzzy identification of systems and its applications to modeling and control,” *IEEE Trans. on Systems, Man and Cybernetics*, vol. 15, no. 1, pp. 116–132, Jan-Feb 1985.
[17] M. Sugeno and G. Kang, “Structure identification of fuzzy model,” *Fuzzy Sets and Systems*, vol. 28, no. 1, pp. 15–33, 1988.
[18] E. H. Mamdani and S. Assilan, “An experiment in linguistic synthesis with a fuzzy logic controller,” *Intern. Journal of Man-Machine Studies*, vol. 7, pp. 1–13, 1975.
[19] Y. Shi and R. Eberhart, “A modified particle swarm optimizer,” in *IEEE International Conference on Evolutionary Computation*, Seoul, Korea, 1998, pp. 69–73.
[20] R. Eberhart and Y. Shi, “Particle swarm optimization: developments, applications and resources,” in *IEEE International Conference on Evolutionary Computation*, Seoul, Korea, 2001, pp. 81–86.
[21] E. F. P. da Luz, “Estimation of atmospheric polluting source using particle swarm optimization,” São José dos Campos, p. 86, 2007, thesis (PhD in Applied Computing), (INPE-15227-TDI/1319).
[22] J. C. Becceneri, S. Stephany, H. F. C. Velho, and E. F. P. Luz, “Addition of atmosphere turbulence in the particle swarm optimization algorithm,” in *National Computing and Applied Math. Congress*, Campinas, Brazil, 2006, p. CD.
[23] R. Marinke, E. Araujo, L. S. Coelho, and I. Matiko, “Particle Swarm Sptimization (PSO) applied to fuzzy modelling in a thermal-vacuum system,” in *Anais...*, Internacional Conference on Hybrid Intelligent Systems. IEEE, 2005, pp. 67–72.
[24] D. G. Gilmore, *Satellite Thermal Control Handbook*. El Segundo, California: The Aerospace Corporation Press, 1994.
[25] *Operations and Maintenance Manual, Thermal Vacuum System with Thermally Conditioned Plate for Brazilian Space Research Institute*, High Vacuum Systems Inc., 1987.
[26] B. Schiweizer, H. Xie, and Y. Lee, “Fuzzy logic models for ranking process effects,” *IEEE Trans. on Fuzzy Systems*, vol. 5, pp. 545–55, 1997.