Nonlinear System Identification of A Twin Rotor MIMO System

Bidyadhar Subudhi, *Senior Member IEEE* and Debashisha Jena

Center for Industrial Electronics & Robotics, Department of Electrical Engineering, National Institute of Technology, Rourkela, India-769008
bidyadhar@nitrkl.ac.in

***Abstract*— This work presents system identification using neural network approaches for modelling a laboratory based twin rotor multi-input multi-output system (TRMS). Here we focus on a memetic algorithm based approach for training the multilayer perceptron neural network (NN) applied to nonlinear system identification. In the proposed system identification scheme, we have exploited three global search methods namely genetic algorithm (GA), Particle Swarm Optimization (PSO) and differential evolution (DE) which have been hybridized with the gradient descent method i.e. the back propagation (BP) algorithm to overcome the slow convergence of the evolving neural networks (EANN). The local search BP algorithm is used as an operator for GA, PSO and DE. These algorithms have been tested on a laboratory based TRMS for nonlinear system identification to prove their efficacy.**

Keywords- ***Differential evolution, Evolutionary computation, Nonlinear system identification, Back propagation, Twin rotor system***

I. INTRODUCTION

SYSTEM identification using neural networks has been considered as a promising approach due to its function approximation properties [1] and for modeling nonlinear system dynamics. However, a lot more research is needed to achieve its faster convergence and obtaining global minima. Hence there has been a great interest in combining training and evolution with neural networks in recent years. The major disadvantage of the EANN [2] approach is that it is computationally expensive and has slow convergence. With a view to speed up the convergence of the search process, a number of different gradient methods such as LM and BP are combined with evolutionary algorithms. These new class of hybrid algorithms i.e. global evolutionary search supplemented by local search techniques are commonly known as memetic algorithms (MAs). MAs have been proven very successful across a wide range of problem domains such as combinatorial optimization [3], optimization of non-stationary functions [4], multi-objective optimization [5], bioinformatics [6] etc.

A variant of evolutionary computing namely the Differential Evolution [7-9] is a population based stochastic optimization method similar to genetic algorithm [4] that finds an increasing interest in the recent year as an optimization technique in the identification of nonlinear systems due to its achievement of a global minimum. However, a little work has been reported on memetic differential evolution learning of neural network. Therefore, it attracts the attention of the present work for neural network training. In this work, a differential evolution hybridized with back propagation has been applied as a optimization method for feed-forward neural network. Differential Evolution (DE) is an effective, efficient and robust optimization method capable of handling nonlinear and multimodal objective functions. The beauty of DE is its simple and compact structure which uses a stochastic direct search approach and utilizes common concepts of EAs. Furthermore, DE uses few easily chosen parameters and provides excellent results for a wide set of benchmark and real-world problems. Experimental results have shown that DE has good convergence properties and outperforms other well known EAs. Therefore, there is scope of using DE approach to neural weight optimization. In comparison to a gradient based method differential evolution seems to provide advantage in terms convergence speed and finding global optimum.

In this work, the authors propose a hybrid approach in which the local search methods (LM, BP) acts as an operator in the global search algorithm in view of achieving global minimum with good convergence speed. Here, genetic algorithm, particle swarm optimization and differential evolution are acting as global search methods which are individually combined with BP for training a feed-forward neural network. In the proposed scheme, in each generation back-propagation acts as an operator which is applied after crossover and mutation operator.

The main contributions of the paper are as follows:

- The paper proposed a new training paradigm of neural networks combining an evolutionary algorithm i.e. DE with a local search algorithm i.e. BP for getting faster convergence in comparison to only evolutionary computation and to avoid the possibility of the search

978-1-4244-4547-9/09/$26.00 ©2009 IEEE TENCON 2009

process being trapped in local minima which is the greatest disadvantage of local search optimization.

- BP has been integrated as an operator in global searches for optimizing the weights of the neural network training enabling faster convergence of the EANN employed for nonlinear system identification.

II. Proposed Differential Evolution Back-Propagation Training Algorithm For Non-Linear System Identification

In this section, we describe how a memetic differential evolution (DE) is applied for training neural network in the frame work of system identification (see Algorithm-1). DE can be applied to global searches within the weight space of a typical feed-forward neural network. Output of a feed-forward neural network is a function of synaptic weights $\mathbf{w}$ and input values $\mathbf{x}$, i.e. $\mathbf{y}=f(\mathbf{x},\mathbf{w})$ .The role of BP in the proposed algorithm has been described in section I. In the training process, both the input vector $\mathbf{x}$ and the output vector $\mathbf{y}$ are known and the synaptic weights in $\mathbf{w}$ are adapted to obtain appropriate functional mappings from the input $\mathbf{x}$ to the output $\mathbf{y}$. Generally, the adaptation can be carried out by minimizing the network error function $\mathbf{E}$ which is of the form $\mathbf{E}(\mathbf{y},f(\mathbf{x},\mathbf{w}))$. In this work we have taken $\mathbf{E}$ as mean squared error i.e. $\mathbf{E}=\frac{1}{N}\sum_{k=1}^{N}\left[\mathbf{y}-f(\mathbf{x},\mathbf{w})\right]^2$, where N is the number of data considered. The optimization goal is to minimize the objective function $\mathbf{E}$ by optimizing the values of the network weights, $\mathbf{w}=(w_1,.....,w_d)$.

ALGORITHM-1: Differential Evolution Back-Propagation (DEBP) Identification Algorithm:

Step 1.

Initialize population **pop**: Create a population from randomly chosen object vectors with dimension P , where P is the number of population

$$\mathbf{P}_g=(\mathbf{w}_{1,g},......,\mathbf{w}_{P,g})^T,\ g=1,......,g_{\max}$$

$$\mathbf{w}_{i,g}=(w_{1,i,g},......,w_{d,i,g}),\quad i=1,.......,P$$

where d is the number of weights in the weight vector. In $\mathbf{w}_{i,g}$, i is index to the population and g is the generation to which the population belongs.

Step 2.

Evaluate all the candidate solutions inside the **pop** for a specified number of iterations.

Step 3.

For each i^{th} candidate in **pop,** select the random population members, $r_1,r_2,r_3\in\{1,2,......P\}$

Step 4.

Apply a mutation operator to each candidate in a population to yield a mutant vector i.e.

$$v_{j,i,g+1}=w_{j,r_1,g}+F(w_{j,r_2,g}-w_{j,r_3,g}),\ \text{for}\ j=1,.....,d$$

$$(i\neq r_1\neq r_2\neq r_3)\in\{1,......,P\}\ \text{and}\ F\in(0,1+]$$

where "F" denotes the mutation factor.

Step 5.

Apply crossover i.e. each vector in the current population is recombined with a mutant vector to produce trial vector.

$$t_{j,i,g+1}=\begin{cases} v_{j,i,g+1} & \text{if } rand_j[0,1)\le C \\ w_{j,i,g} & \text{otherwise}\end{cases}$$

where $C\in[0,1]$

Step 6.

Apply Local Search (back propagation algorithm) i.e. each trial vector will produce a lst-trial vector $lst_{j,i,g+1}=bp\left(t_{j,i,g+1}\right)$

Step 7.

Apply selection i.e. between the local search trial (lst-trial) vector and the target vector. If the lst-trial vector has an equal or lower objective function value than that of its target vector, it replaces the target vector in the next generation; otherwise, the target retains its place in the population for at least one more generation

$$\mathbf{w}_{i,g+1}=\begin{cases} lst_{i,g+1}, & \text{if } \mathbf{E}(\mathbf{y},f(\mathbf{x},\mathbf{w}_{i,g+1}))\le\mathbf{E}(\mathbf{y},f(\mathbf{x},\mathbf{w}_{i,g})) \\ \mathbf{w}_{i,g}\ , & \text{otherwise}\end{cases}$$

Step 8.

Repeat steps 1 to 7 until stopping criteria (i.e. maximum number of generation) is reached

III. Proposed Differential Evolution Back-Propagation Training Algorithm For Non-Linear System Identification

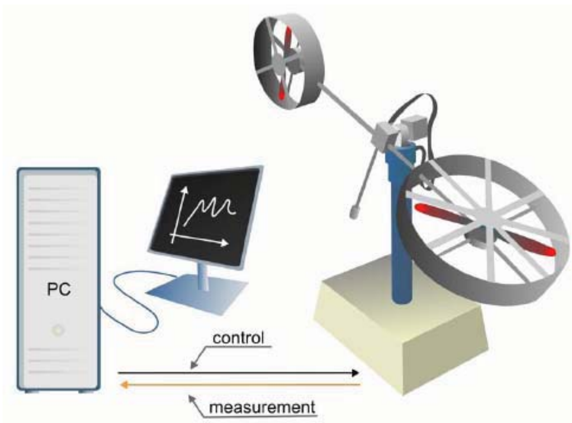

Fig. 1 The laboratory set-up: TRMS system

The TRMS used in this work is supplied by Feedback Instruments designed for control experiments. This TRMS setup serves as a model of a helicopter. It consists of two rotors placed on a beam with a counterbalance. These two rotors are driven by two d.c motors. The main rotor produces a lifting force allowing the beam to rise vertically making the rotation around the pitch axis. The tail rotor which is smaller than the main rotor is used to make the beam turn left or right around the yaw axis. Both the axis of either or both axis of rotation can be locked by means of two locking screws provided for physically restricting the horizontal and or vertical plane of the TRMS rotation. Thus, the system permits both 1 and 2 DOF experiments. In this work we have taken only the 1 DOF around the pitch axis and identified the system using proposed method discussed in section II. The model has three inputs and eleven neurons in the hidden layer. The inputs are the main rotor voltage at the present time *v(t)*, main rotor voltage at previous time *v(t-1)* and the pitch angle of the beam at previous time instant'*s(t-1)*.

A. Differential Evolution (DE) and Differential Evolution Back propagation (DEBP) Identification

Figure 2-6 shows the identification performance of 1 degree of freedom (DOF) vertical DE and DEBP based model. Figure 2 compares the actual output , $y(t)$ and identified plant output $\hat{y}(t)$ within the time step of 0 to 500. As the identification performances shown in Figure 2 are overlapping each other, in Figure 3 we have shown the results within the time step of 88 to96. From this it is clear that the proposed DEBP exhibits better identification ability compared to DE approach. Fig.4 and 5 shows the error between the actual and identified model. Fig.6 gives the sumsquared error (SSE) where it is found that the value of SSE for DEBP is 0.0036 whereas for DE identification is 0.0110

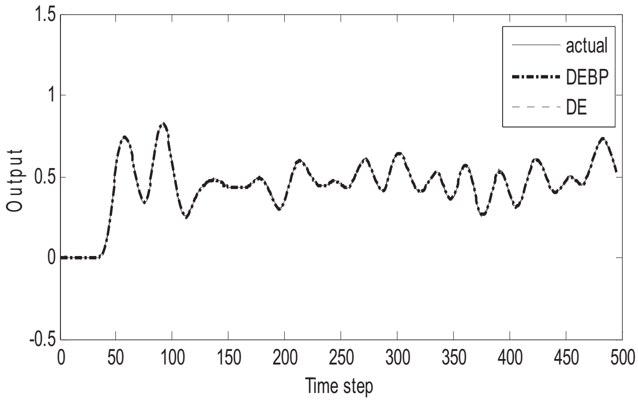

Figure 2 DE and DEBP identification performance

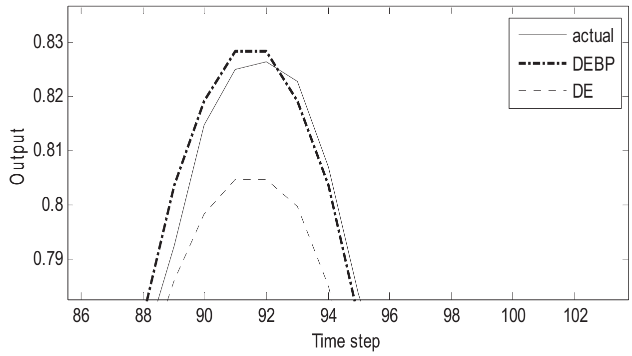

Figure 3 DE and DEBP identification performance

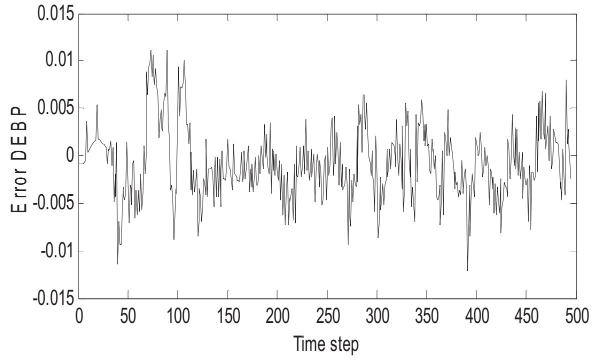

Figure 4 Error in modeling (DEBP identification)

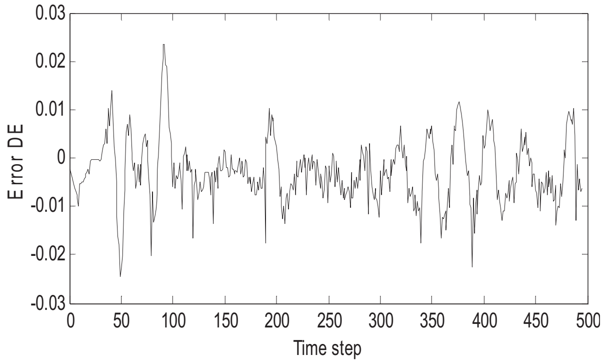

Figure 5 Error in modeling (DE identification)

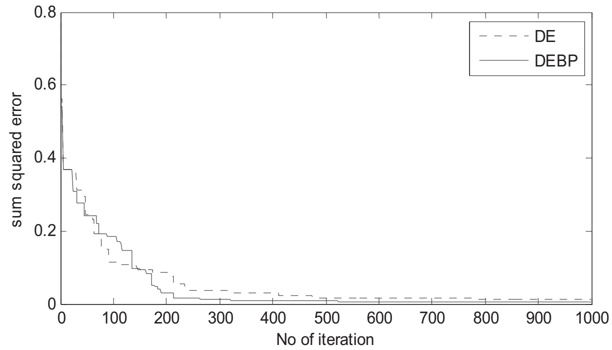

Figure 6 A comparisons on the convergence on the sum squared error (SSE) (DE, DEBP)

B. Genetic algorithm (GA) and Genetic algorithm Back propagation (GABP) Identification

Figure 7-11 shows the identification performance of 1 degree of freedom (DOF) vertical GA and GABP based model. Figure 7 compares the actual output , $y(t)$ and identified plant output $\hat{y}(t)$ within the time step of 0 to 500. As the identification performances shown in Figure 7 are overlapping each other, in Figure 8 we have shown the results within the time step of 208 to 218. From this it is clear that the GABP identification approach exhibits better identification ability compared to GA approach. Fig.9 gives the sumsquared error (SSE) where it is found that the value of SSE for GABP is 0.0.0197 whereas for GA identification is 0.0327. Fig.10 and 11 shows the error between the actual and identified model for both the identification scheme.

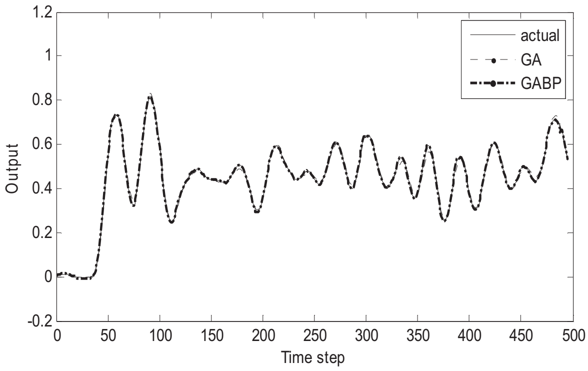

Figure 7 GA and GABP identification performance

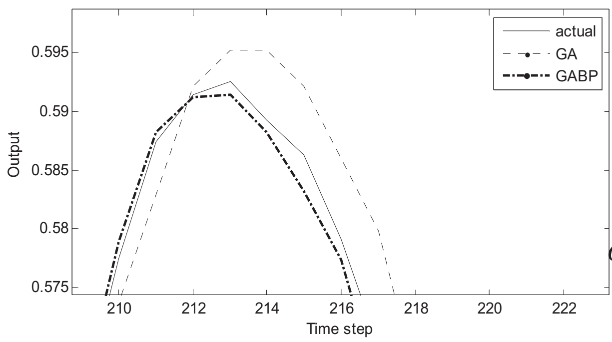

Figure 8 GA and GABP identification performance

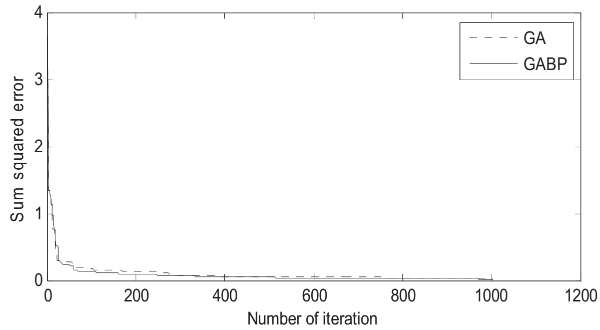

Figure 9 A comparisons on the convergence on the sum squared error (SSE) (GA, GABP)

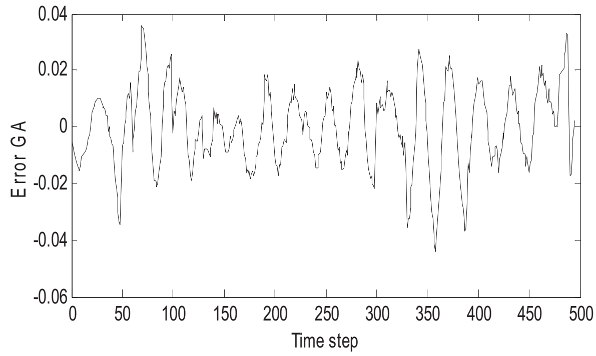

Figure 10 Error in modeling (GA identification)

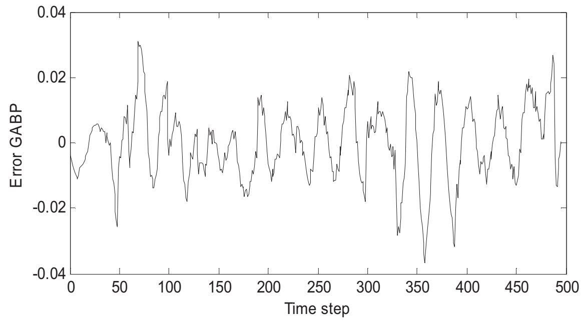

Figure 11 Error in modeling (GABP identification)

C. Particle Swarm Optimization (PSO) and Particle Swarm Optimization (PSOBP) Identification

Figure 12-16 shows the identification performance of 1 degree of freedom (DOF) vertical PSO and PSOBP based model. Figure 12 compares the actual output , $y(t)$ and identified plant output $\hat{y}(t)$ within the time step of 0 to 500. As the identification performances shown in Figure 12 are overlapping each other, in Figure 13 we have shown the results within the time step of 87 to 96. From this it is clear that the PSOBP approach exhibits better identification ability compared to PSO approach. Fig.14 gives the sumsquared error (SSE) where it is found that the value of SSE for PSOBP is 0.0235 whereas for PSO identification is 0.0505. Fig.15 and 16 shows the error between the actual and identified model for both the identification scheme.

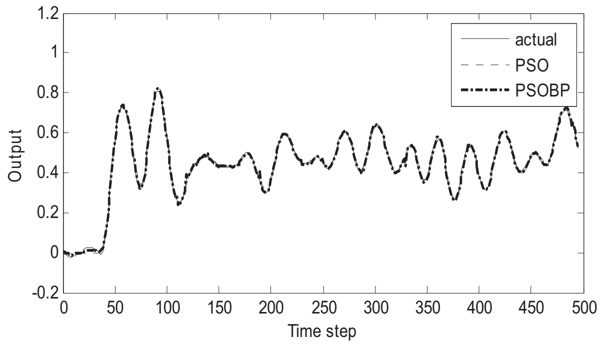

Figure 12 PSO and PSOBP identification performance

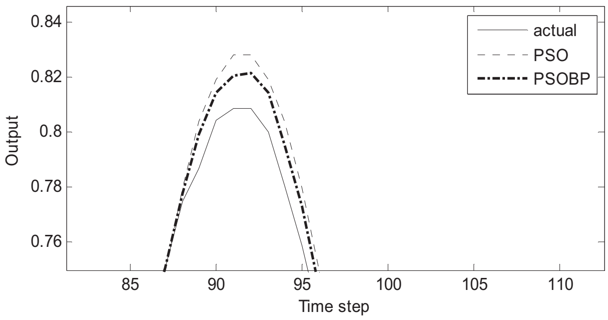

Figure 13 PSO and PSOBP identification performance

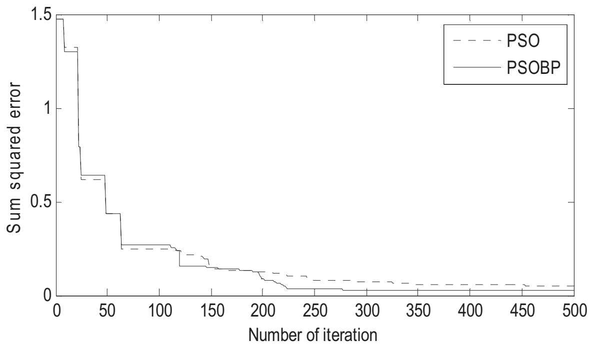

Figure 14 A comparisons on the convergence on the sum squared error (SSE) (PSO, PSOBP)

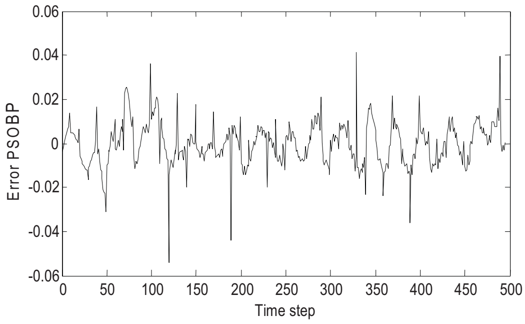

Figure 15 Error in modeling (PSOBP identification)

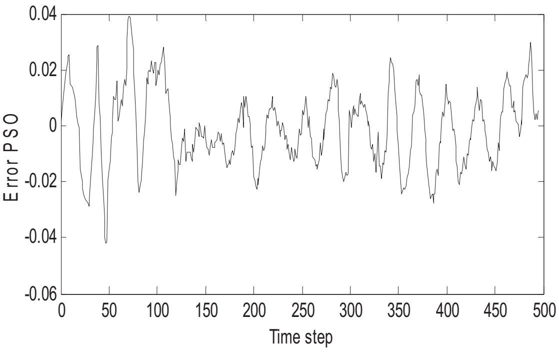

Figure 16 Error in modeling (PSO identification)

Finally it has been seen that among all the methods the proposed DEBP method is having lowest SSE i.e. 0.0036 amongst all the methods disused.

IV. CONCLUSIONS

In this paper we have provided an extensive study of memetic algorithms (MAs) applied to nonlinear system identification. From the results presented in this paper it has been found that the proposed DEBP memetic algorithm applied to neural network learning exhibits better result in terms of faster convergence and lowest mean squared error (MSE) amongst all the six methods (i.e. GA, GABP, PSO, PSOBP, DE. and DEBP). The proposed method DEBP exploits the advantages of both the local search and global search. It is interesting to note that the local search pursued after the mutation and crossover operation that helps in intensifying the region of search space which leads to faster convergence. We investigated the performance of the proposed version of the DEBP algorithm using a real time multi input multi output highly nonlinear TRMS system. The simulation studies showed that the proposed algorithm of DEBP outperforms in terms of convergence velocity among all the discussed algorithms. This shows it is advantageous to use DEBP over other evolutionary computation such as GA and PSO in nonlinear system identification.

REFERENCES

[1] Narendra, K.S., and Parthaasarathy, K., "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Trans Neural Networks*, vol.1, pp 4-27, 1990.

[2] Yao,X. “Evolutionary artificial neural networks”. *Int. Journal of Neural Systems*, vol.4, pp 203-222, 1993.

[3] Merz, P., “Memetic Algorithms for Combinatorial Optimization Problems: Fitness Landscapes and Effective Search Strategies”, PhD thesis, Department of Electrical Engineering and Computer Science, University of Siegen, Germany, 2000.

[4] Vavak, F., Fogarty, T. and Jukes, K “A genetic algorithm with variable range of local search for tracking changing environments”. In *Proceedings of the 4th Conference on Parallel Problem Solving from Nature*, 1996.

[5] Knowles, J. and Corne, D., “A comparative assessment of memetic, evolutionary and constructive algorithms for the multi-objective d-msat problem”, *Genetic and Evolutionary Computation Workshop Proceeding,* 2001.

[6] Krasnogor, N., “Self-generating metaheuristics in Bioinformatics: The protein structure comparison case”. In *Genetic Programming and Evolvable Machines*, Kluwer academic Publishers, vol.5, pp181-201, 2004.

[7] Storn, R., and Price, K., “Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces”. *Journal of Global Optimization*, vol.11, 341–359, 1997.

[8] Storn, R. "System Design by Constraint Adaptation and Differential Evolution." *IEEE Trans. Evol. Comput.* vol.3, pp 22-34, 1999.

[9] Ilonen, J., Kamarainen, J. K.; and Lampinen, J. “Differential Evolution Training Algorithm for Feed Forward Neural Networks.” *Neurol. Proc. Lett.* vol.17, pp 93-105, 2003.

[10] Feedback Instruments Ltd., ***Twin Rotor** MIMO **System Manual 33-007-0,*** Sussex, UK, 1996.

[11] Ahmad, S. M., Shaheed, M. H., Chipperfield, A. J., and Tokhi, M. O. Nonlinear modelling of a twin rotor MIMO system using radial basis function networks. IEEE National Aerospace and Electronics Conference, 2000, pp. 313–320.

[12] Ahmad, S. M., Chipperfield, A. J., and Tokhi, M. O. Dynamic modelling and optimal control of a twin rotor MIMO system. IEEE National Aerospace and Electronics Conference, 2000, pp. 391–398.