APPLICATION OF DE STRATEGY AND NEURAL NETWORK

In position control of a flexible servohydraulic system

Hassan Yousefi, Heikki Handroos
Institute of Mechatronics and virtual Engineering, Mechanical Engineering Department,
Lappeenranta University of Technology, 53851,Lappeenranta, Finland

Keywords: Differential Evolution, Backpropagation, Position Control, Servo-Hydraulic, Flexible load.

Abstract: One of the most promising novel evolutionary algorithms is the Differential Evolution (DE) algorithm for solving global optimization problems with continuous parameters. In this article the Differential Evolution algorithm is proposed for handling nonlinear constraint functions to find the best initial weights of neural networks. The highly non-linear behaviour of servo-hydraulic systems makes them idea subjects for applying different types of sophisticated controllers. The aim of this paper is position control of a flexible servo-hydraulic system by using back propagation algorithm. The poor performance of initial training of back propagation motivated to apply the DE algorithm to find the initial weights with global minimum. This study is concerned with a second order model reference adaptive position control of a servo-hydraulic system using two artificial neural networks. One neural network as an acceleration feedback and another one as a gain scheduling of a proportional controller are proposed. The results suggest that if the numbers of hidden layers and neurons as well as the initial weights of neural networks are chosen well, they improve all performance evaluation criteria in hydraulic systems.

1 INTRODUCTION

Problems which involve global optimization over continuous spaces are ubiquitous throughout the scientific community. In general, the task is to optimize certain properties of a system by pertinently choosing the system parameters. For convenience, a system's parameters are usually represented as a vector. The standard approach to an optimization problem begins by designing an objective function that can model the problem's objectives while incorporating any constraints.

Consequently, we will only concern ourselves with optimization methods that use an objective function. In most cases, the objective function defines the optimization problem as a minimization task. To this end, the following investigation is further restricted to minimization problems. For such problems, the objective function is more accurately called a "cost" function.

One of the most promising novel evolutionary algorithms is the Differential Evolution (DE) algorithm for solving global optimization problems with continuous parameters. The DE was first introduced a few years ago by Storn (Storn, 1995) and Schwefel (Schwefel, 1995).

When the cost function is nonlinear and non-differentiable Central to every direct search method is a strategy that generates variations of the parameter vectors. Once a variation is generated, a decision must then be made whether or not to accept the newly derived parameters. Most stand and direct search methods use the greedy criterion to make this decision. Under the greedy criterion, a new parameter vector is accepted if and only if it reduces the value of the cost function.

The extensive application areas of DE are testimony to the simplicity and robustness that have fostered their widespread acceptance and rapid growth in the research community. In 1998, DE was mostly applied to scientific applications involving curve fitting, for example fitting a non-linear function to photoemissions data (Cafolla AA., 1998). DE enthusiasts then hybridized it with Neural Networks and Fuzzy Logic (Schmitz GPJ, Aldrich C., 1998) to enhance or extend its performance. In 1999 DE was applied to problems involving multiple criteria as a spreadsheet solver application (Bergey PK., 1999). New areas of interest also emerged, such as: heat transfer (Babu BV, Sastry KKN., 1999), and constraint satisfaction problems (Storn R., 1999) to name only a few. In

Yousefi H. and Handroos H. (2005).
APPLICATION OF DE STRATEGY AND NEURAL NETWORK - In position control of a flexible servohydraulic system.
In *Proceedings of the Second International Conference on Informatics in Control, Automation and Robotics*, pages 133-140
DOI: 10.5220/0001190301330140

2000, the popularity of DE continued to grow in areas of electrical power distribution (Chang TT, Chang HC., 2000), and magnetics (Stumberger G. et al, 2000). 2001 furthered extensions of DE in areas of environmental science (Booty WG et al, 2000), and linear system models (Cheng SL, Hwang C., 2001). By the year 2002, DE penetrated the field of medical science (Abbass HA., 2002). Most recently in 2003, there has been a resurgence of interest in applying DE to problems involving multiple criteria (Babu BV, Jehan MML. 2003).

Electro-hydraulic servomechanisms are known for their fast dynamic response, high power-inertia ratio and control accuracy. In the fluid power area, neural network systems have been used for control, identification and modeling of the system (Chen, 1992). The popularity of neural networks can be attributed, in part, to their ability to deal with non-linear systems. In addition, a neural network approach uses a parallel distributed processing concept (D.E.Rumelhart, et al, 1986). It has the capability of improving its performance through a dynamic learning process and, thus, provides powerful adaptation ability. Since neural networks are fashioned after their human neural counterparts, they can be 'trained' to do a specific job *by* exposing the networks to a selected set of input–output patterns. A comparatively convenient method is to have a reference response model, which can be the tracking object of the control system. Following the model reference adaptive control theory (Franklin, 1984), an adaptive reference model is used in this study and implemented in the microcomputer to control a hydraulic variable motor.

Gain scheduling based on the measurements of the operation conditions of the process is often a good way to compensate for variations in the process parameters or the known nonlinearities of the process.

The main aim of the following study is to apply the DE strategy to find the best initial weights of the proposed neural networks to improve the performance of position tracking of the reference model.

The structure of the paper is the following: in section 2 description of the flexible servo-hydraulic system, section 3 controller design, section 4 differential evolution algorithm, section 5 presents the main results of this work, and section 6 draws conclusions.

2 SERVO HYDRAULIC SYSTEM

The hydraulic system with flexible load shown in figure 1 is comprised of a servo-valve, a hydraulic cylinder and two masses that are connected by a parallel combination of spring and damper. The schematic diagram of the system is illustrated in figure1.

The nonlinearity of the constitutive equations as well as the sensitivity of the system's parameters to the sign of the voltage fed to the valve make the control of system too complicated (Viersma, 1980).

2.1 System Model

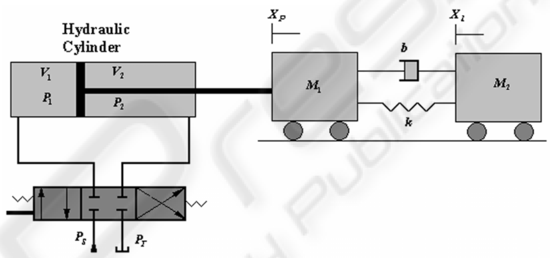

Figure 1: Schematic Diagram of System.

Applying Newton's second law for each mass without consideration of coulomb friction results,

$$m_1\ddot{X}_p = -b_1\dot{X}_p - b_2(\dot{X}_p - \dot{X}_l) - k(X_p - X_l) + p_1A_1 - p_2A_2$$
$$m_2\ddot{X}_l = -b_2(\dot{X}_L - \dot{X}_p) - k(X_l - X_p) \qquad (1, 2)$$

Continuity equations for the output ports of the servo valve results,

$$\begin{cases} \dot{p}_1 = \dfrac{\beta_e}{V_1}(Q_1 - A_1\dot{X}_p) \\ \dot{p}_2 = \dfrac{\beta_e}{V_2}(-Q_2 + A_2\dot{X}_p) \end{cases} \qquad (3)$$

Introducing two new parameters, C_1 and C_2, defined as, $C_1 = \dfrac{V_1}{\beta_e}$, $C2 = \dfrac{V_2}{\beta_e}$

Equation 3 can be written in the form of,

$$\begin{cases} \dot{p}_1 = \dfrac{1}{C_1}(Q_1 - A_1\dot{X}_p) \\ \dot{p}_2 = \dfrac{1}{C_2}(-Q_2 + A_2\dot{X}_p) \end{cases} \qquad (4)$$

The volumes between the valve and each side of piston are calculated as,

$$V_1 = A_1X_p + v_{01}$$
$$V_2 = A_2(L - X_p) + v_{02}$$

If the tank pressure is set to equal to zero ($p_T = 0$), the nonlinear equations of flow rate of valve can be written in the simplest form, as follow,

$$Q_1 = \begin{cases} c_v u \sqrt{p_s - p_1} & u \geq 0 \\ c_v u \sqrt{p_1} & u \prec 0 \end{cases} \tag{5}$$

$$Q_2 = \begin{cases} c_v u \sqrt{p_2} & u \geq 0 \\ c_v u \sqrt{p_s - p_2} & u \prec 0 \end{cases} \tag{6}$$

Table 1: Setup Parameters

$m_1 = 278\ Kg$	$L = 1$ m
$m_2 = 110\ Kg$	$b_1 = 1000$ Ns/m
$A_1 = 8.04 \times 10^{-4}\ m^2$	$b_2 = 500$ Ns/m
$A_2 = 4.24 \times 10^{-4}\ m^2$	$\beta_e = 4 \times 10^8$ Pa
$k = 136670\ N/m$	$P_s = 14$ Mpa
$v_{01} = 2.13 \times 10^{-4}\ m^3$	$v_{02} = 1.07 \times 10^{-4}\ m^3$
$c_v = 2 \times 10^{-8}\ \mathrm{m^3} / (\mathrm{sVPa}^{1/2})$	

Where, $c_v = \dfrac{Q_n}{u_n \sqrt{\Delta p_n}}$

The setup parameters of system are shown in Table1.

Following, the structure of controller for positioning control of each mass is proposed.

3 CONTROLLER DESIGN

Here, the aim of the controller is position tracking of a reference model. Classical approaches, like P or PD regulators for positioning of hydraulic drives, do not give satisfactory performance. For this reason, adaptive control techniques, adaptive reference model, and gain scheduling are used to improve the performance of Controller (4). The variations of parameters depending on the change in the sign of voltage fed to the valve are compensated by gain scheduling block. The second neural network is impelled to improve the lack of damping in hydraulic systems. Following the reference model and each neural network are proposed.

3.1 Reference Model

The desire linear, second order, reference model was selected to run parallel with the nonlinear system. The natural frequency, ω_n , of this model set equal to as 7 rad/sec with the damping ratio, ζ , of 0.9.

$$G_{ref}(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{7}$$

There are kinds of method to find a reference model such as ITAE or Bessel transfer functions. The Bessel transfer functions have not overshot when ω_n is equal to one, but when ω_n is greater than one the overshot appears.

The natural frequency of system was chosen in a manner that the response of system is as fast as possible. The chosen damping ratio provides the minimum overshoot of reference model.

3.2 Neural Network Design

Basic neural networks consist of Neurons, weights and activation function. The weights are adapted to achieve mapping between the input and output sets in the manner to track reference model. Many neural networks are successfully used for various control applications. In this study the backpropagation algorithm with momentum term was used to update the weights and biases of neural network (Rumelhart, 1986). The backpropagation algorithm is a learning scheme in which the error is backpropagated layer by layer and used to update the weights. The algorithm is a gradient descent method that minimizes the error between the desired outputs and the actual outputs calculated by the MLP.

The backpropagation training process requires that the activation functions be bounded, differentiable functions. One of the most commonly used functions satisfying these requirements is the hyperbolic tangent function. This function has a monotonic increasing in the range of –1 to 1. The mathematical model of hyperbolic tangent can be written in the form of,

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{8}$$

The learning procedure requires only that the change in weights and biases are proportional to the $\partial E_p / \partial w$. True gradient descent requires that infinitesimal steps be taken. The constant of proportionality is the learning rate in the procedure. The bigger change of this constant, the bigger change in the weights. For practical purposes we choose a learning rate that is as large as possible without exciting oscillation. This offers the most rapid learning. One way to increase the learning rate without leading to oscillation is to modify the general delta rule to include a *momentum* term. This can be accomplished by the following rule,

$$\Delta w_{ji}(n+1) = \eta(\delta_{pj} o_{pi}) + \alpha \Delta w_{ji}(n) \qquad (9)$$

where n indicates the presentation number. The parameter of η is the learning rate, and α is a constant which determines the effect of past changes on the current direction of movement in weight space.

The proposed controller depicted in figure 2, composed of two neural networks, shown as blocks *Neural Networks* and *Gain Scheduling*. Neural Network Block is used as acceleration feed back to improve the dynamic behavior of system (lack of damping).

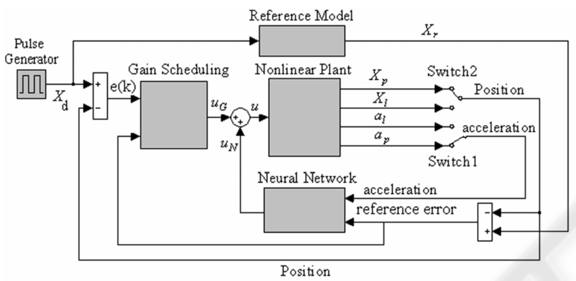

Figure 2: Schematic Diagram of Neural Controller

Neural networks parameters are updated online by learning rate and momentum factors. As it shown in figure, the proposed neural network has two hidden layers, figure3, one input and one output layer. The inputs of neural network are two accelerations ($a_i(k)$, $a_i(k-1)$) and the output of pervious step, $u_n(k-1)$. Several kinds of structure with one or three hidden layers were tested. The proposed structure has this ability to emulate the acceleration feedback and improve the dynamic behavior of system.

The second neural network, gain scheduling network, has the same structure as pervious one, but the inputs are the two controller errors ($e(k)$, $e(k-1)$) and the output of last step, $u_G(k-1)$. Due to the poor results of p-controller in return stage the second neural network tried to compensate this matter. Finally, the controller output, summation of u_N and u_G, is fed to the system.

Choosing the initial weights are too important in back-propagation algorithm. To avoid local minima problem, the genetic algorithm is used to find the off-line weights (Corne. 1999).

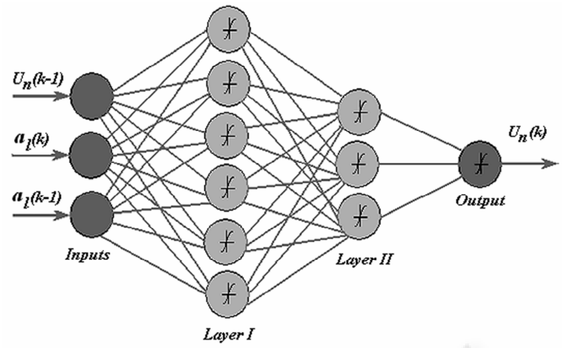

Figure 3: Structure of Neural Network

For online training with backpropagation algorithm the learning rate of $\eta = 2 \times 10^{-6}$ and $\alpha = 0.0001$ are used. The simulation results are depicted in result section.

Following the results of using new controller is compared with common p-controller.

4 DE STRATEGY

Differential evolution (DE) is a simple yet powerful population based, direct-search algorithm for globally optimizing functions defined on totally ordered spaces, including especially functions with real-valued parameters. Real parameter optimization comprises a large and important class of practical problems in science and engineering.

There are two variants of DE that have been reported, DE/rand/1/bin and DE/best/2/bin. The different variants are classified using the following notation:

DE/x /y/z;

where

- x indicates the method for selecting the parent chromosome that will form the base of the mutated vector. Thus, DE/rand/y/z selects the target parent in a purely random manner. In contrast, DE/best/y/z selects the best member of the population to form the base of the mutated chromosome.
- y indicates the number of difference vectors used to perturb the base chromosome.
- z indicates the crossover mechanism used to create the child population. The bin acronym indicates that crossover is controlled by a series of independent binomial experiments.

Following, the schematic procedure of a class of DE (DE/rand/1/bin) is presented.

Input :

$D, G_{max}, NP \ge 4, F \in (0,1+), CR \in [0,1]$ and

initial bounds : $x^{(lo)}, x^{(hi)}$

D= number of parameters.
NP= population size.
F=scale factor.
CR= crossover control constant.
G_{max}=Maximum number of generation.
hi, lo= upper and lower initial parameter bounds, respectively.

Initialize :

$$\begin{cases} \forall i \le NP \wedge \forall j \le D: \\ x_{j,i,G=o} = x_{j,i}^{(lo)} + rand[0,1].\left(x_j^{(hi)} - x_j^{(lo)}\right) \end{cases}$$

(11)

While $G \prec G_{max}$

$\forall i \le NP$:

$$\begin{cases} \text{Mutate and Recombination :} \\ r_1, r_2, r_3 \in \{1,2,...,NP\}, \text{randomly selected} \\ \qquad \text{except} : r_1 \ne r_2 \ne r_3 \ne i \\ j_r \text{ randon selected once each i} \\ \forall\ j \le D: \\ u_{j,i,G+1} = \begin{cases} x_{j,r_3,G} + F.\left(x_{j,r_1,G} - x_{j,r_2,G}\right) \\ \text{If } (rand_j[0,1) \prec CR \vee j = j_r) \\ x_{j,i,G} \quad \text{otherwise} \end{cases} \qquad (12) \\ \textit{Select} : \\ x_{i,G+1} = \begin{cases} u_{i,G+1} & \text{if } F(u_{i,G+1}) \le F(x_{i,G}) \\ x_{i,G} & \text{otherwise} \end{cases} \end{cases}$$

$$G = G + 1 \qquad (13)$$

The aforementioned algorithm finds the global minima more reliable than the other methods. Note that these kinds of algorithms are useful in off-line training as the speed of approaching in not fast.
Here, the numbers of initial weights for each neural network are 49. Following is the selected parameters for this strategy,
D= 49,
NP=$5 \times 49 \approx 250$,
F=0.8,
CR=0.7,
The initial upper and lower bounds are 1,-1, respectively.
The results show that DE can find the global minimal cost for the system. Depend on the expected value of cost, the DE will find the proper weights. Here the cost of system is defined as follow,

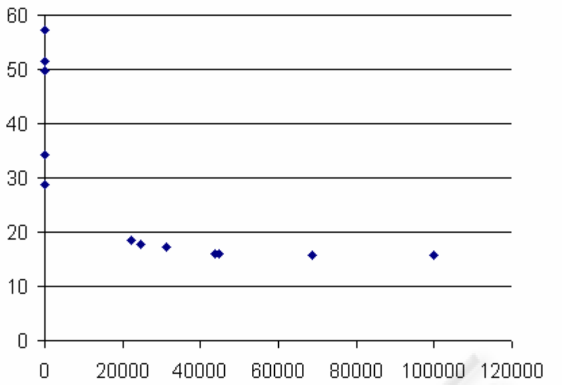

Figure 4: Cost of System against the Number of its Iterations

$$F(x_{i,G}) = \sum_{k=1}^{K=2000} |e(k)| \qquad (14)$$

Where,

$$|e(k)| = |X_r(k) - X_l(k)| \qquad (15)$$

The desire input (X_d) is pulse input with the amplitude of 0.3 (m) and period of 4 (sec.). In the first step the weights of neural network compensator were found. To do that, the gain scheduling was replaced with a constant around 5 and then the cost for half of period was calculated. The sampling time here is one millisecond, so the cost will be calculated 2000 times in each iteration.

Figure 4 shows the amount of cost against its iteration number. The result indicates that at the earlier iteration the amount of cost is too big. Also the algorithm finds new child generations with lower cost to fast (cost bigger than 30). To find the lower cost, the number of iterations must growth up. As it shown, when the number of iteration is around 30000, the related cost is around 17.6, after around 34000 iterations the cost will be fixed at 15.6. This cost will be constant and its related weights has the global minimum cost for the proposed neural compensator.

After aforementioned step for tuning the neural network, using the same strategy, the weights of neural gain scheduling will be found.

In the next section the proposed neural networks will be impelled to position controlling of the system. The important factor in DE is factor of F. The iteration number is incredibly related to this factor. However, the change of crossover factor has not essential effect in decreasing the number of iterations. The chosen value is recommended in many cases to be useful. In this study, the number of generation was 1973 times during 100000 iterations.

5 RESULTS AND DISCUSSIONS

In this section the results of using proposed controller for two cases are provided. These controllers are tracking the reference model for *piston tracking* (m_1), case I, and *flexible load tracking* (m_2), case II, respectively.

The desire input (X_d) is pulse input with the amplitude of 0.3 (m) and period of 4 (sec.).

5.1 Piston Tracking

Figure 5 is the piston tracking of the reference model using neural network with gain scheduling. Figure 6 is its controller effort that will be fed to nonlinear system.

The maximum and minimum voltages fed to the valve are limited to ± 10 (Volt). As it shown in figure 6, the maximum and minimum of voltage are banded because of valve restriction.

5.2 Fexible Load Tracking

In this section the results of using second controller to track of reference model for flexible load is presented. Figure 7 is the flexible load tracking of the reference model using neural network with gain scheduling.

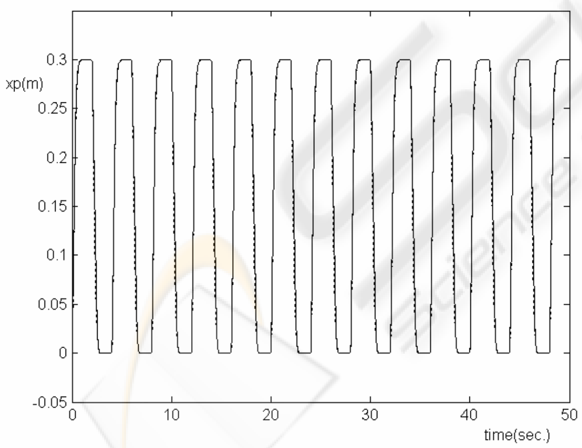

Figure 5: Result of Mass Load Tracking Using Neural Network with Gain Scheduling. The Solid Line Shows the Reference Model and Dots Line Show Mass Load Position

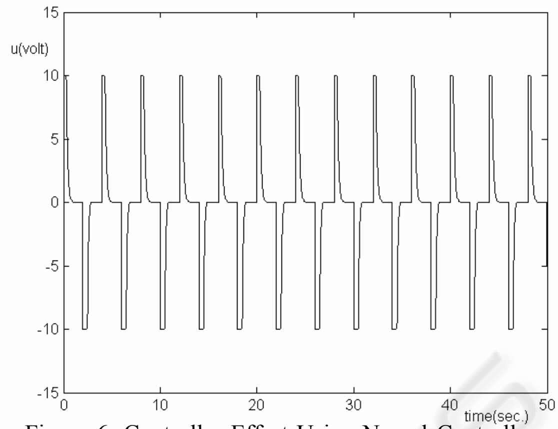

Figure 6: Controller Effort Using Neural Controller with Neural Gain Scheduling

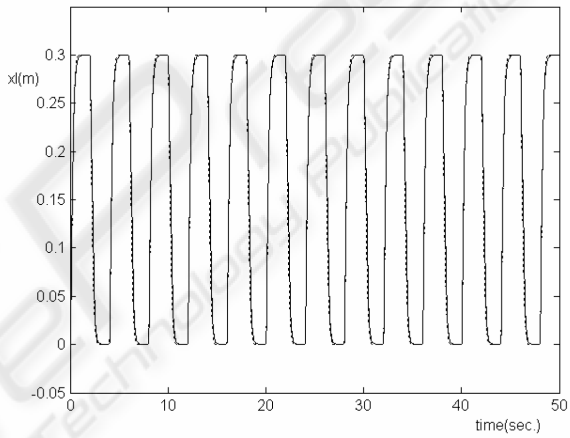

Figure 7: Simulation Result of Flexible Load Tracking Using Neural Network with Gain Scheduling. The Solid Line Shows the Reference Model and Dots Line Show Mass Load Position.

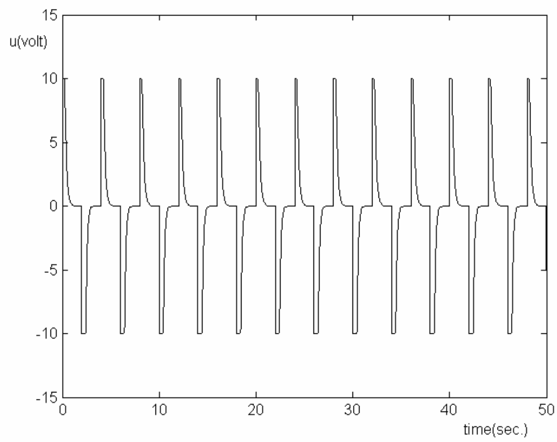

Figure 8: Controller Effort Using Neural Controller with Gain Scheduling

Figure 8 is its controller effort that will be fed to the system.

The acceleration of flexible load is used for the neural network block. The results are satisfactory and illustrate the capability of the neural network in controlling and compensating the dynamic of the servo hydraulic systems.

As it was shown in the figure 5, in rising and falling stages, the flexible load tracking is similar to that of the piston tracking.

Figure 7 shows that the neural network with the gain scheduling has very good performances during the rising and falling. The aforementioned controller can track the reference model very well. Figure 8 shows the output of controller is smooth and has not any oscillation.

6 CONCLUSIONS

In the present paper the application of DE and back-propagation algorithm in position control of a flexible servo-hydraulic system was studied. The results suggest that neural network has good performance if the initial weights of system are set correctly. To avoid the local minima, Differential Evolution Algorithm is essential. These kinds of algorithms are too slow, so they are useful only in off-line training.

In this article the Differential Evolution Algorithm for handling multiple nonlinear functions was proposed and demonstrated with a set of two difficult test problems. With all test cases, the algorithm demonstrated excellent effectiveness, efficiency and robustness. The architectures and the number of hidden layers are the other essential keys to have a well-performed neural network. If the number of hidden layers or neurons in each layer is not enough the results will be poor. Several structures were tested and the aforementioned architecture was adapted.

In light of above, using neural network to compensate the dynamic of hydraulic systems is an essential factor to improve their performances. Also, the implementing of the gain scheduling overcomes the variations of parameters during its perform.

7 NOMENCHLACHERS

A_1 piston area in chamber one
A_2 piston area in chamber two
B_i biases (i =1, 2, 3)
b_0 viscous friction coefficients of cylinder
b_1 viscous friction coefficient of flexible load
c_v flow coefficient
k spring constant
k_p proportional gain
L stroke
m_1 mass of rigid body
m_2 mass of flexible load
p_1 pressure in chamber one
p_2 pressure in chamber two
p_s pressure supply
p_T tank pressure
Q_1 flow rate in chamber one
Q_2 flow rate in chamber two
Q_n nominal flow
u control signal
u_1 proportional controller signal
u_2 neural network controller signal
u_n nominal controller signal
V_1 compressed volume of chamber one
V_2 compressed volume of chamber two
w_{ji} neural network weights (i =1,2,3)
x_l load position
x_p piston position
x_r reference position
X_d desire position
Δp_n nominal pressure difference
α momentum rate
β_e effective bulk modulus
η learning rate
v_0 volume of pipe between valve and cylinder port
ω_n natural frequency

REFERENCES

Abbass HA., 2002. An Evolutionary Artificial Neural Networks Approach for Breast Cancer Diagnosis. *Artificial Intelligence in Medicine*, 25, 265–281.

Babu BV, Jehan MML. 2003. Differential Evolution for Multi-Objective Optimization. *Proceedings of 2003 Conference on Evolutionary Computation (CEC-2003), Canberra, Australia*, 2696–703.

Babu BV, Sastry KKN., 1999. Estimation of Heat Transfer Parameters in a Trickle-Bed Reactor Using Differential Evolution and Orthogonal Collocation. *Computers and Chemical Engineering,* 23(3), 327–339.

Bergey PK., 1999. An Agent enhanced Intelligent Spreadsheet Solver for Multi-Criteria Decision Making. P*roceedings of AIS AMCIS 1999: Americas Conference on Information Systems, Milwaukee, USA,* 966–8.

Booty WG, D Lam CL, Wong IWS, Siconolfi P., 2000. Design and Implementation of an Environmental Decision Support System. *Environmental Modeling and Software*, 16(5), 453–458.

Cafolla AA., 1998. A New Stochastic Optimization Strategy for Quantitative Analysis of Core Level Photoemission Data. *Surface Science,* 402–404,561–565.

Chang TT, Chang HC., 2000. An Efficient Approach for Reducing Harmonic Voltage Distortion in Distribution Systems with Active Power Line Conditioners. *IEEE Transactions on Power Delivery*, 15(3), 990–995.

Chen, S. and Billings, S. A., 1992. Neural Networks for Dynamical System Modeling and Identification. *Int. J. Control*, 56(2), 319-346.

Cheng SL, Hwang C., 2001. Optimal Approximation of Linear Systems by a Differential Evolution Algorithm. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 31(6), 698–707.

David Corne, Marco Dorigo and Fred Glover, 1999, The book. New Ideas in Optimization, McGraw-Hill.

D.E.Rumelhart, et al, 1986. Learning Internal Representations: by Error Propagation. *Parallel Distributed Processing Explorations in the Microstructures of Cognition*, 1,318-362.

Gene F. Franklin, et al., 1986, The Book. Feedback Control of Dynamic Systems, Addison-Wesley.

Schmitz GPJ, Aldrich C., 1998. Neuro-Fuzzy Modeling of Chemical Process Systems with Ellipsoidal Radial Basis Function Neural Networks and Genetic Algorithms. *Computers & Chemical Engineering*, 22, 1001–1004.

Schwefel, H.P., 1995, The Book, Evolution and Optimum Seeking, John Wiley.

Storn R., 1999. System Design by Constraint Adaptation and Differential Evolution. *IEEE Transactions on Evolutionary Computation,* 3(1), 22–34.

Storn, R. and Price, K.V.,1995. Differential evolution - a Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces, *Technical Report TR-95-012, ICSI*.

Stumberger G, Pahner U, Hameyer K., 2000. Optimization of Radial Active Magnetic Bearings Using the Finite Element Technique and the Differential Evolution Algorithm. *IEEE Transactions on Magnetics*, 36(4), 1009–13.

Taco J. Viersma, 1980, The Book. An analysis, Synthesis and Design of Hydraulic Servosystems and Pipelines, Elsevier.