01 Jun 2009

A PSO with Quantum Infusion Algorithm for Training Simultaneous Recurrent Neural Networks

Bipul Luitel

Ganesh K. Venayagamoorthy
Missouri University of Science and Technology

Recommended Citation

A PSO with Quantum Infusion Algorithm for Training Simultaneous Recurrent Neural Networks

Bipul Luitel and Ganesh Kumar Venayagamoorthy

Abstract— Simultaneous Recurrent Neural Network (SRN) is one of the most powerful neural network architectures well suited for estimation and control of complex time varying nonlinear dynamic systems. SRN training is a difficult problem especially if multiple inputs and multiple outputs (MIMO) are involved. Particle swarm optimization with quantum infusion (PSO-QI) is introduced in this paper for training such SRNs. In order to illustrate the capability of the PSO-QI training algorithm, a wide area monitor (WAM) for a power system is developed using a multiple inputs multiple outputs Elman SRN. The SRN estimates speed deviations of four generators in a multimachine power system. Since MIMO structured SRNs are hard to train, a two step approach for training is presented with PSO-QI. The performance of PSO-QI is compared to that of the standard PSO algorithm. Results demonstrate that the SRN trained with the PSO-QI in the two step approach tracks the speed deviations of the generators with the minimum error.

I. INTRODUCTION

SIMULTANEOUS Recurrent Neural Network (SRN) are known to be a powerful class of neural network architectures. As the name signifies, the recurrence is instantaneous i.e. many times within a sampling period [1]. The SRN provides response of a dynamic nonlinear system even when the weights are fixed and therefore is more appropriate for approximating more complex nonlinear systems with less number of neurons. The SRNs have the capability of approximating non-smooth functions which cannot be approximated by conventional Multilayer Perceptrons (MLPs) [2]. Because of the inherent recursive calculation involved in SRN, they are hard to train using traditional training algorithms such as backpropagation through time, which suffer from local minima [3]. Computational Intelligence (CI) based algorithms have got popularity in training of neural networks because of their ability to find global solution in multi-dimensional search space. Swarm and evolutionary based algorithms like Particle Swarm Optimization (PSO) have shown promises in training of SRNs. A hybrid of PSO and Evolutionary Algorithm (EA) called the PSO-EA is used in engine data classification in [3]. By combining the best features of the participating individual algorithms, hybrid algorithms are more robust and have been used in various kinds of optimization problems.

An electric power grid is a geographically large interconnected network of generators, transmission lines, real and reactive power compensators, loads etc. Power sources and generators are widely dispersed in a modern power system configuration. For the stability and security of the power system, distributed control agents are employed to provide reactive control at several places on the power network through Power System Stabilizers (PSSs), Automatic Voltage Regulators (AVRs), Flexible AC Transmission Systems (FACTS) devices, etc. Although local optimization is achieved by the control agents such as PSSs and AVRs, the lack of coordination among the local agents may cause serious problems such as system oscillations (inter-area) in the power network.

Wide Area Control System (WACS) scheme is proposed in [4, 5] in order to minimize the problems encountered in a distributed power network. The increasing complexity and highly nonlinear nature of the power system today requires a Wide Area Monitor (WAM) for fast and accurate monitoring, for effective control of power networks with an adaptive WAC. This is important for different purposes such as reinforcement of power system based on accurate feedback signals obtained during analysis of system dynamics, coordinated approach for the execution of fast stabilizing actions in case of sever network disturbances etc. [6] The WAM provides information to the WAC which then sends appropriate control/feedback signals to the distributed agents in the power network based on some predefined objective functions.

In this study, quantum principle obtained from Quantum PSO (QPSO) has been combined with traditional PSO to form a new hybrid algorithm called as PSO with Quantum Infusion (PSO-QI). A multiple inputs multiple outputs (MIMO) SRN is used to implement the WAM for a two area multimachine system. Since training of a MIMO SRN is computationally complex, a two step training approach is suggested. It is shown through results that PSO-type algorithms can be used to train SRNs. It has been shown in literatures that hybrid algorithms perform better in training of complex neural network architectures [6]. Hence, this study focuses on improving the hybrid technique for accuracy. To improve the training accuracy in implementing a MIMO WAM, PSO-QI is used to train the network using the two step approach. The following sections of the paper are arranged as follows: Multimachine power system is described in Section 2. In Section 3, local and wide area monitors are described. PSO-QI algorithm is described in

Section 4. Results and discussion are given in Section 5 and conclusion in Section 6.

II. MULTIMACHINE POWER SYSTEM

The practical power system is a complex system with thousands of buses, several hundreds of generators and interactions between multiple areas with several inter-area modes of oscillations. The two-area power system [7] of Fig. 1 is a test system which is commonly used to show the effectiveness of controllers in damping slow mode oscillations. The two-area system with the WAM (Fig. 1) consists of two fully symmetrical areas linked together by two transmission lines. Each area is equipped with two identical synchronous generators rated 20kV/900 MVA. All the generators are equipped with identical speed governors and turbines and AVRs and exciters (Fig. 2). Generators G1, G2, G3 and G4 are all equipped with PSSs. The switch S1 (Fig. 2) can be used to provide training and auxiliary control signals to the generators. The switch S2 (Fig. 2) is used to add the PSS signal to the excitation system. The loads are represented as constant impedances and split between the areas in such a way that Area 1 is transferring about 413 MW to Area 2. Three electromechanical modes of oscillation are present in this system; two inter-plant/intra-area modes, one in each area, and one inter-area low frequency mode [8]. The nonlinear behavior of the complete power system in Fig. 1 is simulated in detail in the PSCAD/EMTDC environment (PSCAD, 2004) for this study. The parameters of the two area system are given in [3].
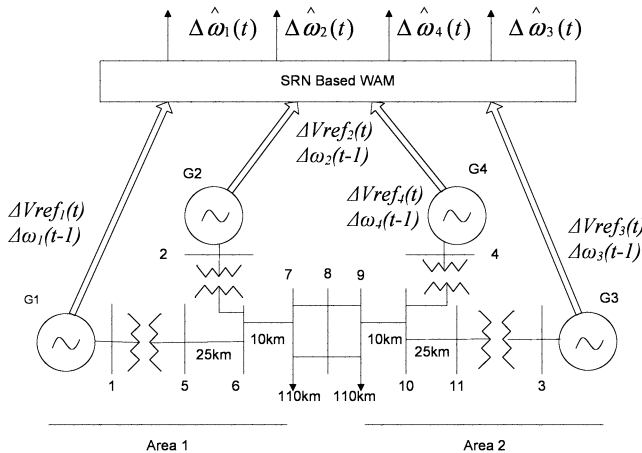

Fig. 1: Two-area power system with a WAM predicting the speed deviations of generators G1, G2, G3 and G4.
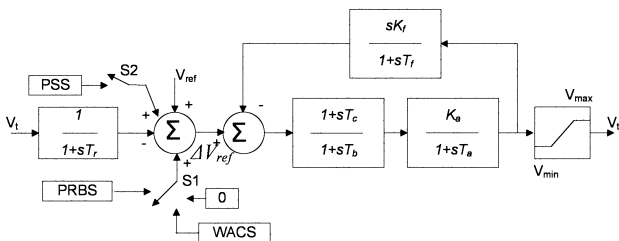

Fig. 2: Block diagram of the AVR-exciter model and showing the interface to the WACS.

III. WIDE AREA MONITOR

A WAM implemented in this work consists of a three "layered" Elman SRN. An Elman SRN has its feedback from the hidden layer output to the context layer inputs. It has an input layer with 8 input nodes, a hidden layer node with 15 hidden nodes and an output layer with 4 output nodes. Being an Elman network, it also has a context layer with 15 nodes whose inputs are the outputs of the corresponding hidden layer nodes. Fig. 3 shows the Elman SRN used as WAM.
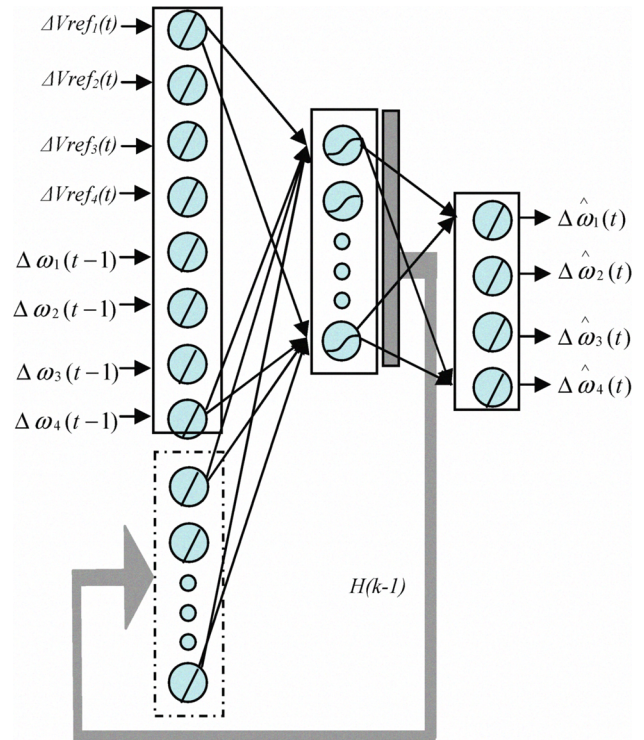

Fig. 3: Elman SRN used in the WAM.

Then inputs to the SRN are the current deviations in reference voltage V_{ref} (Fig. 2), caused by the PRBS excitation, of the four generators. The previous step speed deviations of these generators are also the other set of inputs. The SRN WAM receives these inputs every 10 ms (100 Hz), which is possible with today's phasor measurement unit technology [3]. The hidden nodes have sigmoid activation function and the output nodes are linear. In vector notation, an Elman SRN is implemented as follows [9]:

$$H(t,k) = f(A * I(t,k) + B * H(t,k-1) + K) \quad (1)$$

$$O(t) = g(C * H(t,k) + K') \quad \text{When } k = R \quad (2)$$

1924

where $I = [\Delta Vref_i, \Delta \omega_i]$ for $i = 1$ to 4 is the set of inputs, H is the set of outputs from the hidden nodes and $O = [O_1, O_2, O_3, O_4] = [\Delta \hat{\omega}_1, \Delta \hat{\omega}_2, \Delta \hat{\omega}_3, \Delta \hat{\omega}_4]$ is the set of outputs. A contains weights from input layer to the hidden layer, B contains weights from context layer to the hidden layer, C contains weights from hidden layer to the output layer, k is the time index of internal recurrence, t is the time index of the input sample, R is the total number of internal recurrences, K and K' are the biases, f and g are the two activation functions.

The WAM under considerations is a MIMO system and identification of such system is very difficult. Hence a two step procedure for training the MIMO SRN is implemented. In Step1, SRN as shown in Fig. 3 is used and is trained using PSO and PSO-QI to obtain the input and output weights. In Step 2, the input weights obtained in Step 1 are kept fixed, and the same SRN is trained to obtain the output weights, with only one output at a time. The output of the WAM is the predicted values of the speed deviation for the current sample.

IV. PARTICLE SWARM OPTIMIZATION WITH QUANTUM INFUSION

Particle swarm optimization with quantum infusion is a new approach to hybridization of PSO and Quantum Particle Swarm Optimization (QPSO) [10]. Here, the quantum principle in QPSO is used to create a new offspring. After the position and velocity of the particles are updated using standard PSO equations, a randomly chosen particle from PSO's *pbest* population is utilized to carry out the quantum operation; and thus, create an offspring by mutating the *gbest*. The fitness of the offspring is evaluated and the offspring replaces the *gbest* particle of PSO only if it has a better fitness. This ensures that the fitness of the *gbest* particle is equal to or better than its fitness in the previous iteration. Thus, it is improved and pulled towards the best solution over iterations. By infusing the quantum theory to the standard PSO, a new hybrid algorithm is evolved which incorporates the best features of the respective individual algorithms and thus a better fitness is achieved. In PSO-QI, fast convergence property obtained by PSO in the first few iterations, and the convergence to a lower average error property obtained by QPSO, have been combined and hence the performance is significantly improved, as is shown in the results and figures below. It is described below in detail.

PSO is an evolutionary-like algorithm developed by Eberhart and Kennedy in 1995 [11]. It is a population based search algorithm and is inspired by the observation of natural habits of bird flocking and fish schooling. In PSO, a swarm of particles moves through a D dimensional search space. The particles in the search process are the potential solutions, which move around a defined search space with some velocity until the error is minimized or the solution is reached, which is decided by the fitness function. The particles reach to the desired solution by updating their position and velocity according to the PSO equations. In PSO, each individual is treated as a volume-less particle in the D-dimensional space, with the position and velocity of the ith particle represented as:

$$x_i = (x_{i1}, x_{i2}, \ldots\ldots, x_{iD}) \qquad (3)$$

$$v_i = (v_{i1}, v_{i2}, \ldots\ldots, v_{iD}) \qquad (4)$$

$$v_{id}(t+1) = wv_{id}(t) + c_1 rand_1(P_{id}(t) - x_{id}(t)) + c_2 rand_2(P_{gd}(t) - x_{id}(t)) \qquad (5)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \qquad (6)$$

These particles are randomly initialized over the search space with initial positions and velocities. They change their positions and velocities according to (5) and (6) where c_1 and c_2 are cognitive and social acceleration constants respectively, $rand_1()$ and $rand_2()$ are two random functions uniformly distributed in the range of [0,1] and w is the inertia weight introduced to accelerate the convergence speed of PSO [11]. Vector $P_i = (P_{i1}, P_{i2}, \ldots\ldots\ldots, P_{iD})$ is the best previous position (the position giving the best fitness value) of particle i called the *pbest*, and vector $P_g = (P_{g1}, P_{g2}, \ldots\ldots\ldots\ldots, P_{gD})$ is the position of the best particle among all the particles in the swarm and is called the *gbest*. x_{id}, v_{id}, P_{id} are the dth dimension of vector of x_i, v_i, P_i. PSO has been shown in the flowchart in Fig. 3.

QPSO was introduced by Sun in 2004 [12]. According to the uncertainty principle, position and velocity of a particle in quantum world cannot be determined simultaneously. Thus QPSO differs from standard PSO mainly in the fact that exact values of x and v cannot be determined. In quantum mechanics, a particle, instead of having position and velocity, has a wavefunction given by:

$$\psi(r, t) \qquad (7)$$

which has no physical meaning but its amplitude squared gives the probability measure of its position in any one dimension r at time t. The governing equation of quantum mechanics is the Schrodinger's equation given by:

$$j\hbar \frac{\partial}{\partial t} \psi(r, t) = \hat{H}(r)\psi(r, t) \qquad (8)$$

where H is a time-independent Hamiltonian operator given by:

$$\hat{H}(r) = -\frac{\hbar^2}{2m}\nabla^2 + V(r) \qquad (9)$$

where $\hbar$ is Planck's constant, m is the mass of the particle

1925

and $V_p(r)$ is the potential energy distribution [13]. Based on the probability density function, a particle's probability of appearing in position x can be determined. Therefore in QPSO, a Delta-potential-well based probability density function has been used with center of the well at point $J = (j_1, j_2, .., j_D)$ in order to avoid explosion and help the particles in PSO to converge [14]. Assuming a particle in one-dimensional space having its center of potential at J, normalized probability density function Q and distribution function D_f can be obtained [15]. Let $y = x-j$, then the form of this probability density function is given as follows and depends on the potential field the particle lies in:

$$Q(y) = \frac{1}{L} e^{-2|y|/L} \qquad (10)$$

$$D_f(y) = \int_{-\infty}^{y} Q(y) dy = e^{-2|y|/L} \qquad (11)$$

where the parameter L is the length of the potential field which depends on the energy intensity and is called the creativity or imagination of the particle that determines its search scope [14]. L can be evaluated as the distance between the particles' current position and point J as follows:

$$L = 2\beta | J - x | \qquad (12)$$

The parameter β is the only parameter of the algorithm. It is called the creativity coefficient and is responsible for the convergence speed of the particle. In QPSO, search and solution spaces are two unique spaces of different quality. So a mechanism is necessary to map the position of a particle in the search space to the solution space. This is called 'collapsing' and is achieved by applying the Monte Carlo simulation. This has been explained in [12] as follows.

Let s be any random number uniformly distributed between 0 and $1/L$. For a uniform random number u in the interval $[0, 1]$, s is defined as:

$$s = \frac{1}{L} u \qquad (13)$$

Now, equating (10) and (13), the following relation is achieved:

$$u = e^{-2|y|/L} \qquad (14)$$

$$y = \pm \frac{L}{2} \ln(1/u) \qquad (15)$$

The position equation is given as follows:

$$x = J \pm \frac{L}{2} \ln(1/u) \qquad (16)$$

where the particle's local attractor point J has coordinates given by the following equation:

$$J_d(t) = \alpha_1 P_{gd}(t) + \alpha_2 P_{id}(t) \qquad (17)$$

where $\alpha_1 = a/(a + b)$ and $\alpha_2 = b/(a + b)$, and a and b are two uniformly distributed random numbers.

From (12) and (15), the new position of the particle is calculated as:

$$x(t+1) = J(t) \pm \beta | J(t) - x(t) | \ln(1/u) \qquad (18)$$

This Delta-Potential-well based quantum PSO is called the QDPSO in [12]. This has been improved further by defining a mainstream thought [15] or the Mean Best Position, *mbest*, as:

$$mbest(t) = \frac{1}{S} \sum_{i=1}^{S} P_i(t)$$
$$= \left(\frac{1}{S} \sum_{i=1}^{S} P_{i1}(t), ..., \frac{1}{S} \sum_{i=1}^{S} P_{iD}(t) \right) \qquad (19)$$

where S is the size of the population, D is the number of dimensions and P_i is the *pbest* position of each particle. Now the position update equation in (18) is given as (20), where the addition or subtraction is carried out with 50% probability:

$$x(t+1) = J(t) \pm \beta | mbest(t) - x(t) | \ln(1/u) \quad (20)$$

By using (17) this can also be written as follows to show the mutation on *gbest*:

$$x(t+1) = \alpha_1 P_{gd}(t) + \alpha_2 P_{id}(t)$$
$$\pm \beta | mbest(t) - x(t) | \ln(1/u) \qquad (21)$$

PSO-QI is comparable to Estimation of Distribution PSO (EDPSO) [16] where new particle is created based on the probabilistic models of the search space. Hence the PSO-QI mutations are more likely to produce better offspring than other random mutation techniques.

V. STUDIES AND RESULTS

An Elman SRN based wide area monitor is developed in this paper. A forced training is carried out in which all four generators are subjected to a PRBS excitation and their corresponding speed deviations are measured. One thousand data samples obtained in ten seconds are used for the

training. Previous samples of each generator's speed deviation are fed as input to the SRN along with the deviations in the reference voltage of the generators due to the PRBS excitation. The SRN outputs are one step ahead predicted values of speed deviations. The SRN is trained using PSO and PSO-QI. In Step 1 of the two step training process, SRN has been trained using all the inputs and outputs. In Step 2, the input weights obtained from the first approach have been kept fixed and the SRN is trained only for the output weights, one at a time for each output. After the network has been trained, it has been tested on the same dataset. For both training algorithms, Mean Squared Error (MSE) between the output of the SRN and the actual output of the generator has been used as the measure of fitness. For each generator *i*, MSE for Step 1 can be written as (22).

$$MSE_T = \frac{1}{4} \sum_{i=1}^{4} MSE_i \tag{22}$$

where,

$$MSE_i = \frac{1}{N} \sum_{k=1}^{N} (\Delta\omega_i(k) - \Delta\hat{\omega}_i(k))^2 \tag{23}$$

where $\Delta\omega$ is the actual output of the generator and $\Delta\hat{\omega}$ is the predicted output from the SRN at sample *k*. Eq. (23) gives the MSE for Step 2. Fig. 4 shows the distinction between the two steps of training SRN. The outputs are evaluated and compared in terms of their mean squared error as well as the absolute relative error. The absolute relative error (ARE) is defined as (24).

$$ARE_i = \frac{|\Delta\omega_i(k) - \Delta\hat{\omega}_i(k)|}{|\Delta\omega_i(k)|} \tag{24}$$

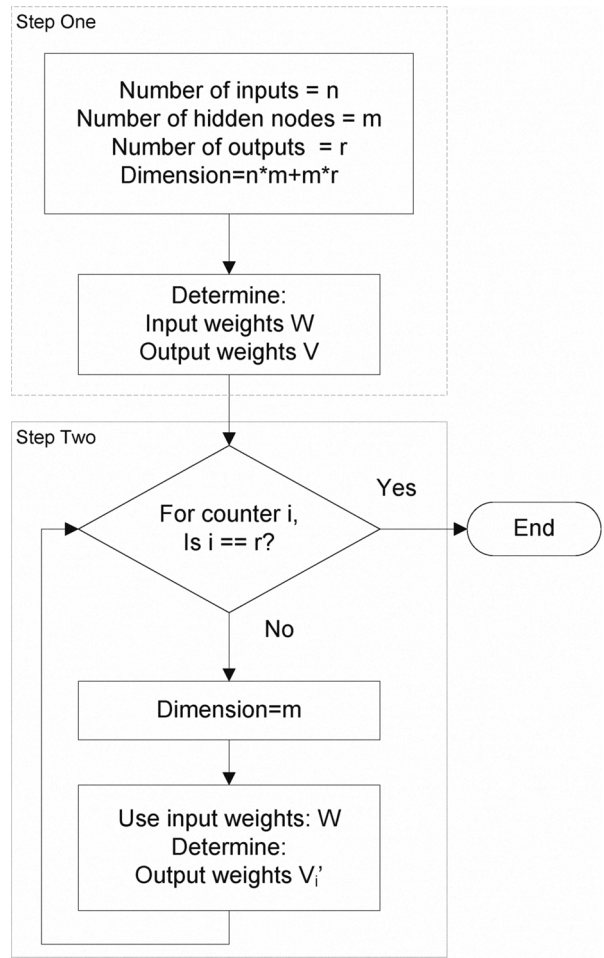where the symbols have the same meaning as (23).

Fig. 4: Two step training of SRN.

In [6], it is shown that a third order model is enough to represent the power system under consideration. Since four of such systems are being considered here, the network should be able to model four third order systems in order to correctly identify the whole system. From trial and experience, a network with the following parameters is used, but is not claimed to be optimal.

Input Nodes (*n*) = 8 (4 PRBS, 4 speed deviations)
Hidden Nodes (*m*) = 15
Output Nodes (*r*) = 4 (Step 1), 1 (Step 2)
Number of samples (*N*): 1000

The following parameters are used for PSO and PSO-QI:
c_1, c_2 = 2
w = linearly decreasing from 0.9 to 0.4
Population Size: 30
Number of iterations: 20
β = linearly increasing from 0.5 to 1
Dimension (*D*) = 405 (Step 1), 15 (Step 2)

The speed deviation output obtained while testing is plotted along with the actual output. The plots shown are for any random trial. Fig. 5 shows the PRBS input to the generator G1. Similar inputs are applied to the other

1927

generators simultaneously. The speed deviation prediction of G1 obtained from the SRN trained using Step 1 is shown in Fig. 6. The figure shows the ability of PSO-QI to better train the neural network and hence it predicts the output more accurately. The same output after the SRN is further trained in Step 2 is shown in Fig. 7. This figure clearly shows a significant amount of improvement in the prediction by both the algorithms. However, the output of PSO-QI is more close to the actual output than PSO. Similar comparison of the outputs of generators G2, G3, and G4 for the two step training process are shown in Figs. 8 to 13. The numerical values of MSE averaged over 10 trials are compared in Table 1. These results show that PSO-QI performs better than PSO in both steps. It also confirms that SRN trained in Steps 1 and 2 is able to predict the speed deviations much better than the SRN trained in Step 1 alone. Although training of MIMO neural network is difficult and computationally complex, using the proposed two step training process and PSO-QI algorithm, better accuracy in training is achieved.
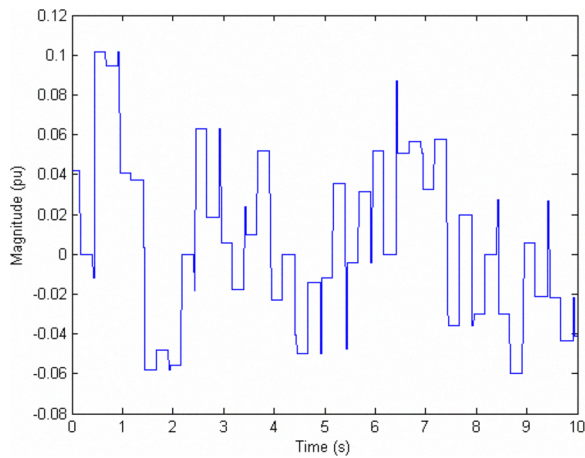

Fig. 7: Testing plot for G1 in step 2.

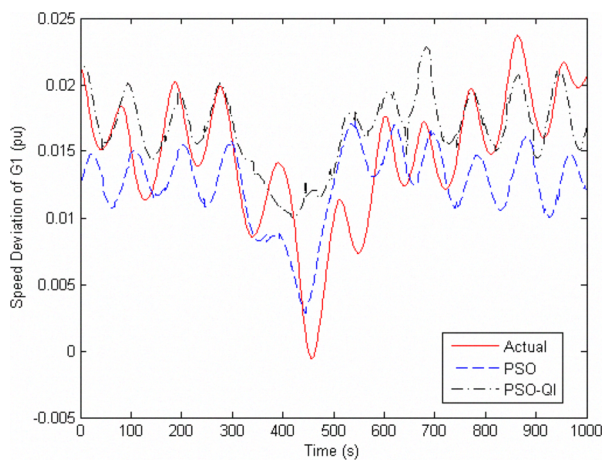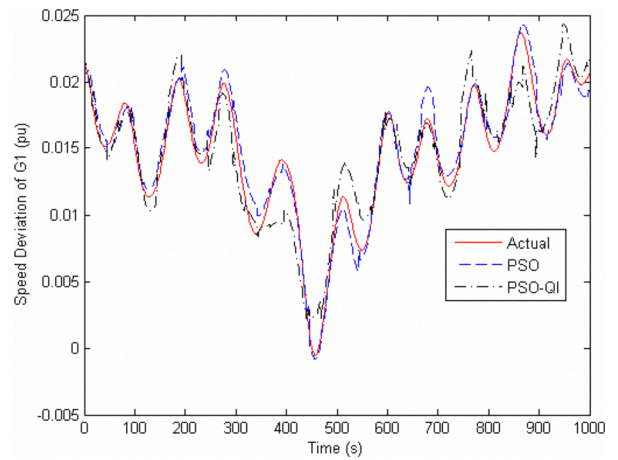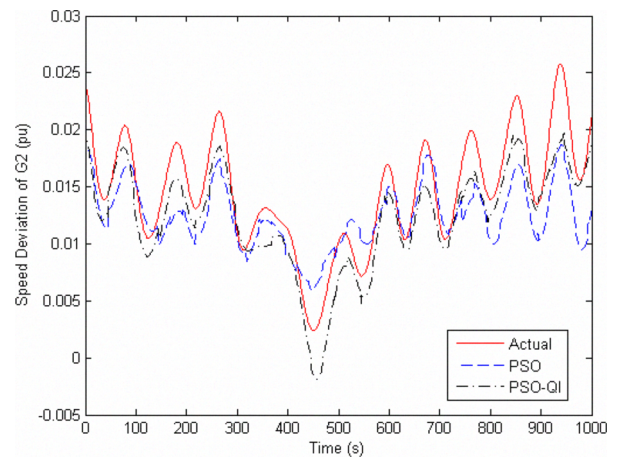Fig. 5: PRBS input to G1.

Fig. 8: Testing plot for G2 in step 1.
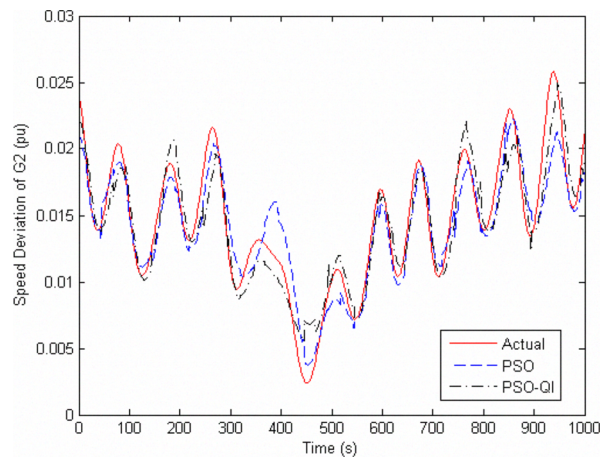

Fig. 9: Testing plot for G2 in step 2.

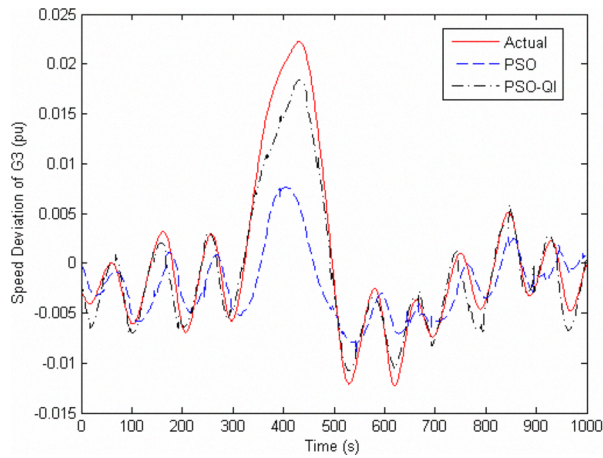Fig. 6: Testing plot for G1 in step 1.
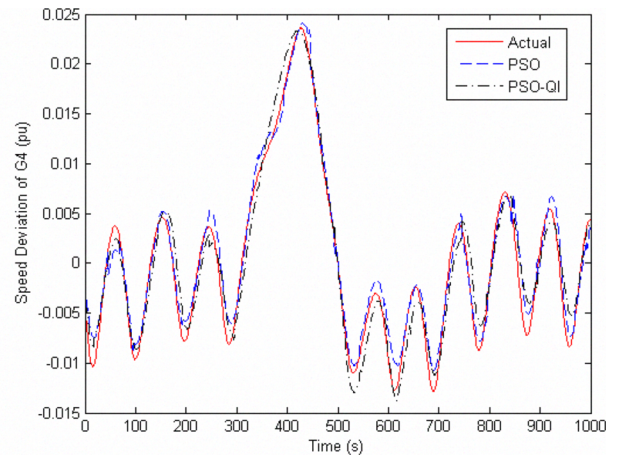
Fig. 10: Testing plot for G3 in step 1.

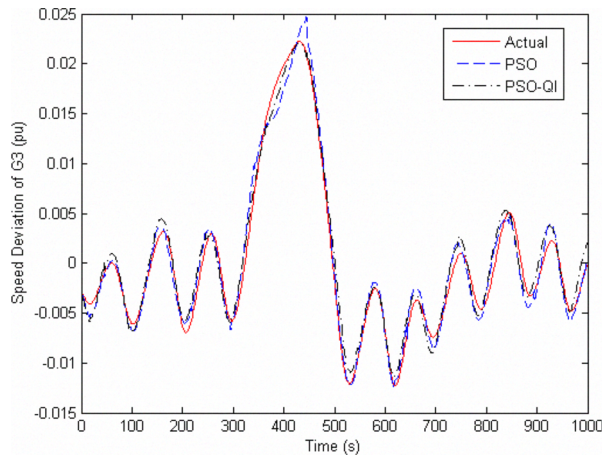Fig. 13: Testing plot for G4 in step 2.

Fig. 11: Testing plot for G3 in step 2.

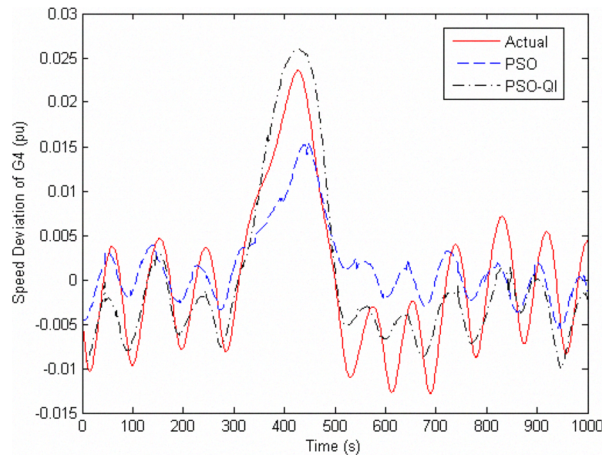Fig. 12: Testing plot for G4 in step 1.

TABLE I
COMPARISON OF RESULTS OBTAINED IN TWO STEPS

G	Algorithm	Mean ARE		MSE (10^{-5})	
		Step One	Step Two	Step One	Step Two
I	PSO	0.1186	0.0321	2.50	0.27
	PSO-QI	0.1240	0.0265	1.99	0.24
II	PSO	0.1092	0.0360	2.75	0.38
	PSO-QI	0.1296	0.0300	2.24	0.27
III	PSO	0.2039	0.0612	3.59	0.27
	PSO-QI	0.1996	0.0425	2.81	0.16
IV	PSO	0.2173	0.0593	3.03	0.38
	PSO-QI	0.1852	0.0494	2.61	0.19
Avg.	PSO	0.1622	0.0471	2.96	0.32
	PSO-QI	0.1596	0.0371	2.41	0.21

VI. CONCLUSION

A new algorithm PSO with quantum infusion and a two step approach for training MIMO SRNs has been presented in this paper. By implementing quantum mechanical concept in mutation of the *gbest* particle, PSO-QI produces offspring more intelligently, than other evolutionary techniques with random mutation, in the vicinity of the solution and thus increasing the speed of convergence. The performance of the PSO-QI algorithm was compared to that of PSO in terms of the mean squared error between the actual and the predicted outputs and the absolute relative error at each sample. Results show that a MIMO SRN performance is improved significantly with PSO-QI and the two step training approach. These significant improvements in SRN performance are at the cost of more training time.

It has been shown that a MIMO SRN can be effectively used as a wide area monitor in multimachine power systems to predict the speed deviations of the generators. For further research, it is important to study if any improvement in reducing the number of iterations required for training in Step 2 to that of Step 1. The application of such training

1929

approach in other MIMO problems also needs to be explored. Along the lines of power system, comparison of multiple local monitoring units with a single wide area monitor and consideration of transmission delays in the prediction time are also topics of future research.

REFERENCES

[1] J. A. Geib, G. Serpen, "Computational Promise of Simultaneous Recurrent Network with a Stochastic Search Mechanism", in *Proceedings of the International Joint Conference on Neural Networks,* vol. 3, Jul. 2004, pp. 2239-2244

[2] M. Sakai, N. Homma, K. Abe, "Statistical Learning Method of Discontinuous Functions using Simultaneous Recurrent Networks", in *Proceedings of the SICE Annual Conference,* vol. 5, Aug. 2002, pp. 3110-3115

[3] X. Cai, D. C. Wunsch II, "Engine Data Classification with Simultaneous Recurrent Network using a Hybrid PSO-EA algorithm", in *Proceedings of the International Joint Conference on Neural Networks,* vol. 4, Jul. 2005, pp. 2319-2323

[4] H. Ni, G. T. Heydt, "Power System Stability Agents using Robust Wide Area Control", *IEEE Transactions on Power Systems,* vol. 17, Nov. 2002, pp. 1123-1131

[5] C. W. Taylor, D. C. Erickson, R. E. Wilson, V. Venkatasubramanian, "WACS-Wide Area Stability and Voltage Control System: R & D and Online Demonstration", in *Proceedings of the IEEE,* vol. 93, 2005, pp. 892-906

[6] G. K. Venayagamoorthy, "Online Design of an Echo State Network based Wide Area Monitor for a Multimachine Power System", *Neural Networks,* 2007, pp. 404-413

[7] P. Kundur, *Power System Stability and Control,* 1994, pp. 813, McGraw-Hill.

[8] M. Klein, G. J. Rogers, P. Kundur, "A Fundamental Study of Inter-area Oscillations in Power Systems"*, IEEE Transactions on Power Systems,* vol. 6-3, Aug. 1991, pp. 914-921

[9] M. Wang, W. Liu, Y. Zhong,, "Simple Recurrent Network for Chinese Word Prediction", in *Proceedings of the International Joint Conference on Neural Networks,* vol. 1, Oct. 1993, pp. 263-266

[10] B. Luitel, G. K. Venayagamoorthy, "Particle Swarm Optimization with Quantum Infusion for the Design of Digital Filters", in *Proceedings of Swarm Intelligence Symposium (SIS),* Sept. 2008, pp. 1-8

[11] Y. del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J. C. Hernandez, R. G. Harley, "Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems", *IEEE Transactions on Evolutionary Computation,* vol. 12, Apr 2008, pp. 171-195

[12] J. Sun, B. Feng, W. Xu, "Particle Swarm Optimization with Particles having Quantum Behavior", in *Congress on Evolutionary Computation,* vol. 1, June 2004, pp. 325-331

[13] R. C. Eberhart, Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization", in *Proceedings of the Congress on Evolutionary Computation,* vol.1, Jul. 2000, pp. 84-88

[14] J. Sun, W. Xu, B. Feng, "Global Search Strategy of Quantum-behaved Particle Swarm Optimization", in *IEEE Conference on Cybernetics and Intelligent Systems,* vol. 1, Dec 2004, pp. 111-116

[15] J. Sun, W. Xu, B. Feng, "Adaptive Parameter Control for Quantum-behaved Particle Swarm Optimization on Individual Level"*, in IEEE International Conference on Systems, Man and Cybernetics,* vol. 4, Oct. 2005, pp. 3049-3054

[16] M. El-Abd, M. S. Kamel, "Particle Swarm Optimization with Varying Bounds", in *IEEE Congress on Evolutionary Computation,* Sep. 2007, pp. 4757 - 4761.