

A Comparison of Particle Swarm Optimization and Gradient Descent in Training Wavelet Neural Network to Predict DGPS Corrections

Mohammad Divband

Abstract—The performance of the basic GPS system has been augmented by the technique of Differential GPS (DGPS) for military as well as civilian uses. Performance evaluation of a DGPS system requires the availability of DGPS corrections as functions of time. In many parts of the world, a lack of base stations and other infrastructure makes it impossible to have the desired quality and quantity of data. Thus, it is useful to develop a system, which can generate GPS measurements for an arbitrary number of truth points. In this paper, Wavelet Neural Network (WNN) is used to online predict the corrections for Selective Availability (S/A) on and off. Gradient Descent (GD) and Particle Swarm Optimization (PSO) are used to train and optimize the weights of WNN. Experimental results for the errors real-time prediction show the feasibility and effectiveness of WNN-PSO. The results prove that the proposed WNN-PSO method has better accuracy in a low cost GPS receiver.

Index Terms—Wavelet Neural Network, Prediction, Particle Swarm Optimization, Gradient Descent, DGPS.

I. INTRODUCTION

The overall quality of precise point positioning results is dependent on the quality of the Global Positioning System (GPS) measurements and user processing software. Dual frequency, geodetic-quality GPS receivers are routinely used both in static and kinematic applications for high accuracy point positioning. However, use of low-cost, single-frequency GPS receivers in similar applications creates a challenge because of how the ionosphere, multipath, and other measurement error sources are handled [1].

During past several years, the main problem in improving of the positioning measuring accuracy was Selective Availability (S/A) error. S/A was produced to degrade the achievable navigation accuracy when non-military single frequency GPS receivers are used. Although it is removed now, we investigated the system performance also under this limitation.

Other significant error sources for low cost receivers are signal delays from ionospheric and tropospheric effects, satellite clock drift, satellite orbital position errors, signal multi path, and noise generated within the receiver itself. Table I shows the common errors of GPS system in meters.

TABLE I. AVERAGE ERROR INTRODUCED PER SATELLITE

Error Sources	Error Value (m)
Receiver Noise	0.4
Troposphere	0.5
Signal Multi-path	0.6
Satellite Clocks	1.5
Orbit Errors	2.5
Ionosphere	5.0
Selective Availability	30

GPS accuracy can be improved over with Differential GPS (DGPS), where a reference station broadcasts corrections on common view satellites on a regular basis to the remote GPS receiver, which provides a corrected position output. A reference station calculates differential corrections for its own location and time then send the corrections to receivers, which are not far from it. Any interrupt of DGPS service will cause loss of navigation guidance, which has possibility of developing into a vehicle accident, particularly in the phase of precision approach and landing. Thus, achievement of corrections in any second is impossible for ordinary users [2].

There are two approaches to provide continuity performance of the DGPS corrections; one is to make the receivers hardware utilities more sophisticated and complicated. This solution could increase the accurate receivers cost. Consequently, non-military users would not benefit from low cost high-precision positioning. Another solution is to use software programs to improve the quality of positioning. In this paper, one of the soft computing techniques, improved Wavelet Neural Network, is used to predict the future corrections.

In order to improve the precision of the corrections forecasts, a Wavelet Neural Network (WNN) model, based on Particle Swarm Optimization (PSO), has been proposed. Corrections time series analysis requires mapping complex relationships between inputs and output, because the forecasted value is mapped as a function of patterns observed in the past. The DGPS corrections future value is represented by the previous data, as given in (1):

$$\hat{x}(k+1) = F(x(k), x(k-1), \dots, x(k-M+1)) \quad (1)$$

Proposed method validity is verified with experiments on collected real data. This paper is organized as follow. Section II describes Wavelet Neural Network with GD learning rule. In section III, a brief introduction of PSO, then the proposed method for DGPS corrections prediction using WNN based on PSO will be described. In section IV, the experimental

Manuscript received July 25, 2010.

Mohammad Divband is a student with the Computer Engineering Group, Iran University of Science and Technology, Behshahr, Iran (e-mail: M.Divband@b-iust.ac.ir).

results with real data are reported, before and after S/A error. Conclusions are presented in section V.

II. WAVELET NEURAL NETWORK STRUCTURE AND GRADIENT DESCENT LEARNING METHOD

Recently, a new kind of Neural Networks known as the Wavelet Neural Networks (WNNs) have been proposed, which combine feed-forward neural network with the wavelet theory. It can provide better performance in function learning than conventional feed forward neural networks [3], [4].

A. Structure of WNN and Forward Calculation

This WNN consists of three layers: an input layer, a hidden layer, and an output layer. The input layer has M nodes. The output layer also has only one neuron whose output is the signal represented by the weighted sum of several wavelets. The hidden layer is composed of a finite number of wavelets representing the signal.

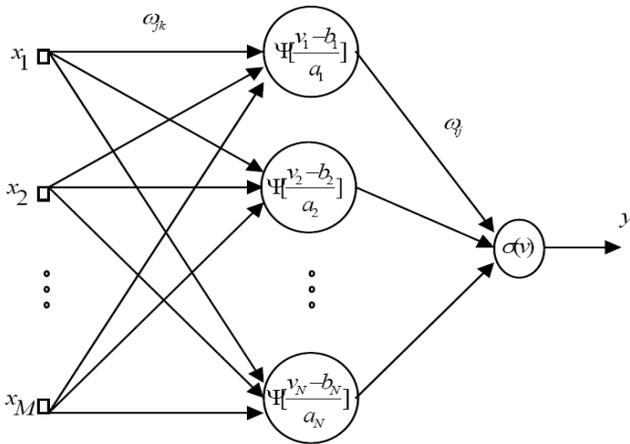


Fig. 1. Structure of a (M, N, 1) Wavelet Neural Network.

Consider a network consisting of a total of N neurons in hidden layer with M external input connections (Fig. 1). Let $x(n)$ denotes the M -by-1 external input vector applied to the network, $y(n)$ denotes the output of the network, $W_{jk}(n)$ presents the weight between the hidden unit j and input unit k , $W_{ij}(n)$ denotes the connection weight between the output unit i and hidden unit j , $a_j(n)$ and $b_j(n)$ present dilation and translation coefficients of wavlon in hidden layer at discrete time n , respectively.

The net internal activity of neuron j at time n , is given by:

$$v_j(n) = \sum_{k=0}^M w_{jk}(n) \cdot x_k(n) \quad (2)$$

Where $v_j(n)$ is the sum of inputs to the j -th hidden neuron, $x_k(n)$ is the k -th input at time n . The output of the j -th neuron is computed by passing $v_j(n)$ through the wavelets $\psi_{a,b}(\odot)$, obtaining:

$$\psi_{a,b}[v_j(n)] = \psi\left[\frac{v_j(n) - b_j(n)}{a_j(n)}\right] \quad (3)$$

The sum of inputs to the output neuron is obtained by:

$$v(n) = \sum_{j=0}^N w_{ij}(n) \cdot \psi_{a,b}[v_j(n)] \quad (4)$$

The output of the network is computed by passing $v_j(n)$ through the nonlinear function $\sigma(\odot)$, obtaining:

$$y(n) = \sigma[v(n)] \quad (5)$$

B. Gradient Descent learning rule

GD learning rule is central to much current work on learning in artificial NN. GD provides a computationally efficient method of changing the weights in a feed forward network, with differentiable activation function units, to learn a training set of input-output examples.

The instantaneous sum of squared error at time n as:

$$E(n) = \frac{1}{2} e^2(n) = \frac{1}{2} [y(n) - d(n)]^2 \quad (6)$$

Where $d(n)$ denotes the desired response of output at time n . To minimize of above cost function, the method of steepest descent is used. The weight between the hidden unit j and input unit k can be adjusted according to:

$$\begin{aligned} \Delta w_{jk}(n+1) &= -\eta \cdot \frac{\partial E(n)}{\partial w_{jk}(n)} + \mu \cdot \Delta w_{jk}(n) \\ &= \eta \cdot e(n) \cdot \sigma'[v(n)] \cdot w_{ij}(n) \\ &\quad \cdot \psi_{a,b}'[v_j(n)] \cdot \frac{x_k(n)}{a_j(n)} + \mu \cdot \Delta w_{jk}(n) \end{aligned} \quad (7)$$

Where, η is a learning rate. The connection weight between the output unit i and hidden unit j is updated as follow:

$$\begin{aligned} \Delta w_{ij}(n+1) &= -\eta \cdot \frac{\partial E(n)}{\partial w_{ij}(n)} + \mu \cdot \Delta w_{ij}(n) \\ &= \eta \cdot e(n) \cdot \sigma'[v(n)] \cdot w_{ij}(n) \cdot \psi_{a,b}'[v_j(n)] + \mu \cdot \Delta w_{ij}(n) \end{aligned} \quad (8)$$

The translation coefficient of the j -th wavlon in hidden layer can be adjusted according to:

$$\begin{aligned} \Delta b_j(n+1) &= -\eta \cdot \frac{\partial E(n)}{\partial b_j(n)} + \mu \cdot \Delta b_j(n) \\ &= -\eta \cdot e(n) \cdot \sigma'[v(n)] \cdot w_{ij}(n) \cdot \psi_{a,b}'[v_j(n)] \cdot \frac{1}{a_j(n)} \\ &\quad + \mu \cdot \Delta b_j(n) \end{aligned} \quad (9)$$

The dilation coefficient of the j -th wavlon in hidden layer is updated as follow:

$$\begin{aligned} \Delta a_j(n+1) &= -\eta \cdot \frac{\partial E(n)}{\partial a_j(n)} + \mu \cdot \Delta a_j(n) \\ &= -\eta \cdot e(n) \cdot \sigma'[v(n)] \cdot w_{ij}(n) \cdot \psi'_{a,b}[v_j(n)] \\ &\quad \cdot \frac{v_j(n) - b_j(n)}{a_j(n)^2} + \mu \cdot \Delta a_j(n) \end{aligned} \quad (10)$$

The wavelet function is ‘‘Gaussian-derivative’’ function as:

$$\psi(x) = -x \cdot e^{-\frac{1}{2}x^2} \quad (11)$$

The usual sigmoid function of used in this research is as follow [5], [6]:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (12)$$

III. PARTICLE SWARM OPTIMIZATION AND TRAINING WNN

A. Introduction to PSO

Particle Swarm Optimization (PSO), first introduced by Kennedy and Eberhart in the mid 1990s. PSO employs a population of possible solutions to identify promising regions of the search space. The population is called swarm and the members of the population are called particles. Each particle represents a possible solution to the optimizing problem at hand. During an iteration of the PSO, each particle accelerates independently in the direction of its own personal best solution found so far, as well as the direction of the global best solution discovered so far by any other particle. Therefore, if a particle finds a promising new solution, all other particles will move closer to it, exploring the solution space more thoroughly [7].

A swarm consists of a set of particles moving around the search space, each representing a potential solution (fitness). Each particle has a position vector ($\omega_i(t)$), a velocity vector ($v_i(t)$), the position at which the best fitness ($pbest_i$) encountered by the particle, and the index of the best particle ($gbest$) in the swarm [8].

In each generation, the velocity of each particle is updated to their best-encountered position and the best position encountered by any particle using (13):

$$\begin{aligned} v_i(t+1) &= wv_i(t) + c_1 r_1(t)(pbest_i - \omega_i(t)) \\ &\quad + c_2 r_2(t)(gbest - \omega_i(t)) \end{aligned} \quad (13)$$

The parameters c_1 and c_2 are called acceleration coefficients namely called self-cognitive and social parameter, respectively. $r_1(t)$ and $r_2(t)$ are random values, uniformly distributed between zero and one. The values of $r_1(t)$ and $r_2(t)$ are not same for every iteration. w is called inertia weight and is employed to control the impact of the previous history of velocities on the current one. Shi and Eberhart [9] have found a significant improvement in the performance of PSO with the linearly decreasing inertia weight over the generations, time-varying inertia weight that

is given in (14):

$$w = (w_1 - w_2) \cdot \left(\frac{\max\ iter - iter}{\max\ iter} \right) + w_2 \quad (14)$$

Where w_1 and w_2 represent the initial and final values of w , respectively, $\max\ iter$ is the maximum number of optimization steps and $iter$ represents the current iteration number. The position of each particle is updated every generation. This is done by adding the velocity vector to the position vector, as in (15):

$$\omega_i(t+1) = \omega_i(t) + v_i(t+1) \quad (15)$$

The algorithms output is the $gbest$ particle, which contains final trained weights and thresholds.

IV. SIMULATIONS AND RESULTS

Computer simulation was performed to evaluate the correction prediction performance using WNN both with GD and PSO algorithms. The choice of the algorithms parameters is also very important. In this paper, the proposed methods parameters selection was based on the test data. The parameters of the proposed algorithms are listed in Table II.

TABLE II. PARAMETERS VALUES OF GD AND PSO

Algorithm	Parameter name	Parameter value
GD	Number of Training Epochs	7
	Learning Factor Value	15
	Momentum	0
PSO	Swarm Size	120
	Self-recognition coefficient	2
	Social coefficient	2
	Inertia weight	0.9 → 0.4
	Number of Iterations	50

We tested both methods for one thousand times. Precise positioning needs X , Y , and Z , thus we executed the algorithms for these three time series. Fig. 2 to Fig. 5 show Ex , Ey , and Ez prediction errors (the difference between the predicted and real values) for 1000 test data.

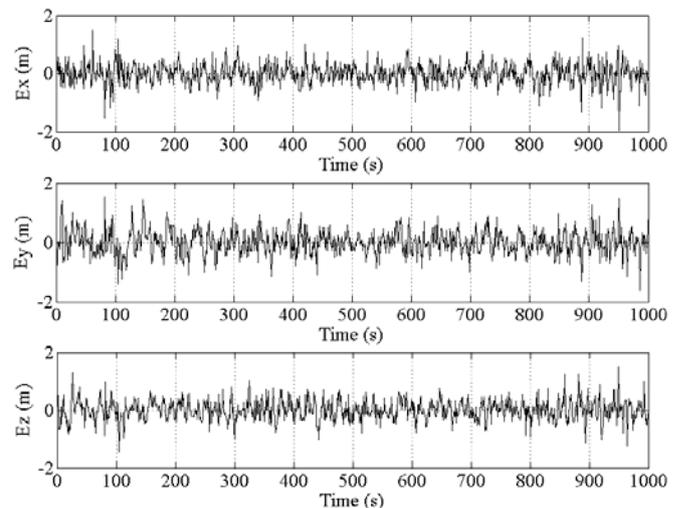


Fig. 2. 1000 Ex , Ey , and Ez prediction errors by using WNN-GD and S/A=off.

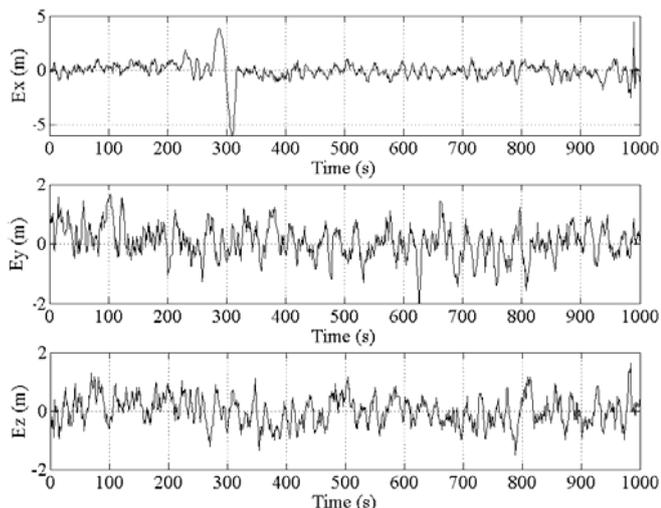


Fig. 3. 1000 Ex, Ey, and Ez prediction errors by using WNN-GD and S/A=on.

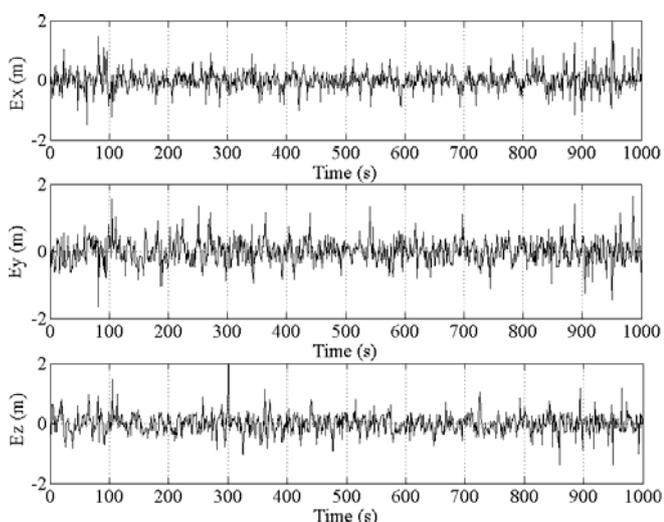


Fig. 4. 1000 Ex, Ey, and Ez prediction errors by using WNN-PSO and S/A=off.

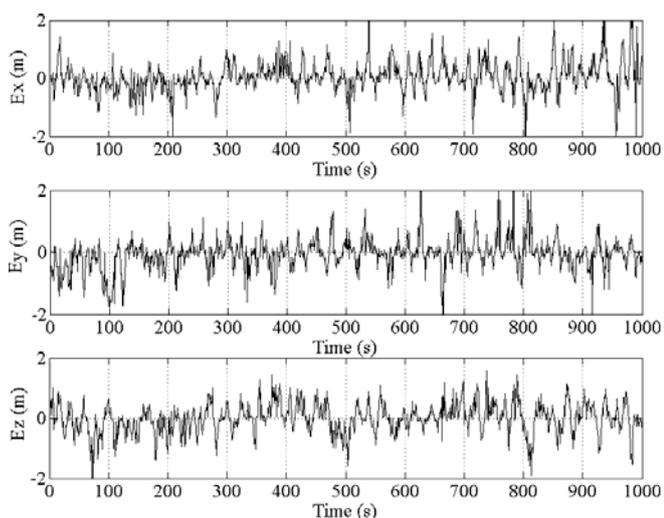


Fig. 5. 1000 Ex, Ey, and Ez prediction errors by using WNN-PSO and S/A=on.

Six statistical measures (Min, Max, Average, Variance, Standard Deviation, and Root Mean Square) are used to evaluate prediction results for 1000 test data. Tables III to VI, show prediction errors statistical significance characteristics.

TABLE III. RESULTS OF WNN-GD METHOD WITH S/A OFF.

Parameters	Ex	Ey	Ez
Min	-1.9665	-1.6186	-1.4473
Max	1.4934	1.5420	1.4976
Average	-0.0001	0.0104	0.0046
STD	0.0112	0.0128	0.0112
Variance	0.0001	0.0001	0.0001
RMS	0.3546	0.4055	0.3547

TABLE IV. RESULTS OF WNN-GD METHOD WITH S/A ON.

Parameters	Ex	Ey	Ez
Min	-6.0472	-2.1593	-1.4908
Max	4.4234	1.6632	1.6296
Average	-0.0456	0.0740	0.0057
STD	0.0284	0.0175	0.0153
Variance	0.0008	0.0003	0.0002
RMS	0.8991	0.5599	0.4858

TABLE V. RESULTS OF WNN-PSO METHOD WITH S/A OFF.

Parameters	Ex	Ey	Ez
Min	-1.4895	-1.6552	-1.3857
Max	2.0434	1.6334	2.3861
Average	0.0020	-0.0007	-0.0072
STD	0.3291	0.3655	0.3274
Variance	0.1083	0.1336	0.1071
RMS	0.3289	0.3654	0.3273

TABLE VI. RESULTS OF WNN-PSO METHOD WITH S/A ON.

Parameters	Ex	Ey	Ez
Min	-4.4234	-2.2396	-2.8039
Max	2.8414	3.2295	1.5808
Average	0.0510	-0.0630	-0.0108
STD	0.5693	0.5309	0.5078
Variance	0.3241	0.2818	0.2578
RMS	0.5713	0.5343	0.5076

As shown in Tables III to VI, accuracy in Tables V and VI are higher than that in Tables III and IV. To clearly compare the results, total RMS errors are reported in Table VII.

TABLE VII. COMPARISON OF PREDICTION ACCURACY BY USING GD AND PSO IN TRAINING WNN PREDICTOR.

Algorithm	Total RMS error in S/A off	Total RMS error in S/A on
GD	0.6449	1.1654
PSO	0.5906	0.9324

V. CONCLUSION

Learning with GD is normally slow due to the characteristics of the error surface on which the weights are navigating. PSO has been proposed for training in order to speed the convergence up, and to avoid local minima in the error surface. Some of the attractive features of the PSO include ease of implementation and the fact that no gradient information is required. In this paper, corrections prediction in single-frequency GPS receivers using WNN trained by a PSO algorithm is presented. The Conventional way of training WNN, GD was less effective compared to PSO. The results are analyzed. Experimental results for the errors real-time prediction showed the feasibility of the WNN, trained by the PSO algorithm, which was able to reduce RMS errors to 0.9324 meter with S/A on and 0.5906 meter with S/A off.

REFERENCES

- [1] T. Beran, D. Kim, and R. B. Langley, "High precision single-frequency GPS point positioning," *Proc. of ION GPS*, 2003, pp. 1192-1200.
- [2] J. Sang, K. Kubik, and L. Zhang, "Prediction of DGPS Corrections with Neural Networks," *Proc. of Intl. Conf. on Knowledge-Based Intelligent Electronic Systems*, Vol. 2, 1997, pp. 355-361.
- [3] R. Drossu and Z. Obradovic, "Rapid Design of Neural Networks for Time Series Prediction," *IEEE Computational Sci. and Eng.*, No. 2, Vol. 3, 1996, pp. 78-89.
- [4] Subanar and Suhartono, "New Procedures for Model Building in Wavelet Neural Networks for Forecasting non-Stationary Time Series," *Proc. of 5th Asian Mathematical Conf.*, Vol. 1, 2009, pp. 203-211.
- [5] Y. Chen, B. Yang, and J. Dong, "Time-series prediction using a local linear wavelet neural network," *Neurocomputing*, No. 6, Vol. 69, 2006, pp. 449-465.
- [6] M. R. Mosavi, "A wavelet based neural network for DGPS corrections prediction," *WSEAS Trans. on Systems*, 2004, pp. 3070-3075.
- [7] F. V. D. Bergh and A. P. Engelbrecht, "A Cooperative Approach to Particle Swarm Optimization," *IEEE Trans. on Evolutionary Computation*, No. 3, Vol. 8, 2004, pp. 225-239.
- [8] M. S. Arumugam, G. R. Murthy, M. V. C. Rao, and C. K. Loo, "A Novel Effective Particle Swarm Optimization Like Algorithm via Extrapolation Technique," *IEEE Conf. on Intelligent and Advanced Systems*, 2007, pp. 516-521.
- [9] Y. Shi and R. C. Eberhart, "Parameter Selection in Particle Swarm Optimization," *Evolutionary Programming VII, Lecture Notes in Computer Sci.* 1447, Springer, 1998, pp. 591-600.