Mohammad REZA MOSAVI[1]

Iran University of Science and Technology (1)

Error Reduction for GPS Accurate Timing in Power Systems using Kalman Filters and Neural Networks

Abstract. *The Global Positioning System (GPS) based time reference provides inexpensive but highly-accurate timing and synchronization capability and meets requirements in power system fault location, monitoring, and control. Precision satellite clocks and time measurements are the keys to the accuracy of GPS. A stand-alone civilian user enjoys an accuracy of 25 meters and 200 nanoseconds. Five methods, including two methods using Kalman Filter (KF), Recurrent Neural Network (RNN), Pi-Sigma Neural Network (PSNN) and Sigma-Pi Neural Network (SPNN), are proposed for error reduction of GPS receivers timing data. We use actual data to evaluate the performance of the proposed methods. An experimental test setup is designed and implemented for this purpose. Results using the five methods are discussed. The experimental results obtained from a Coarse Acquisition (C/A)-code single-frequency GPS receiver strongly support the potential of the method using PSNN to give high accurate timing. The GPS timing RMS error reduces to less than 38 nanoseconds.*

Streszczenie. *Opisano zastosowanie system pozycjonowania GPS lokalizacji uszkodzeń i monitorowania sieci przesyłowej. System cywilny GPS oferuje dokładność rzędu 25 m i 200 nanosekund. Opracowano szereg metod poprawy dokładności, wykorzystujących filtry Kalmana i sieci neuronowe. Zredukowano błąd taktowania do około czterdziestu nanosekund. (**Metody zmniejszenia błędów systemu GPS przy wykorzystaniu filtrów Kalmana i sieci neuronowych**)*

Keywords: Error Reduction, GPS Receivers, Timing Data, Power Systems, Kalman Filter, Recurrent Neural Network, Pi-Sigma Neural Network, Sigma-Pi Neural Network
Słowa kluczowe: GPS, filtry Kalmana, sieci neuronowe.

NOMENCLATURE

RMS	Root Mean Square
SA	Selective Availability
KF	Kalman Filter
NN	Neural Network
PMU	Phasor Measuring Unit
PPS	Pulse Per Second
UTC	Universal Time Coordinated
MSE	Mean Squared Error
AR	Auto-Regressive
UTOD	Universal Time of Day
RNN	Recurrent Neural Network
MLP	Multi Layer Perceptron
BP	Back Propagation
HONN	High-Order Neural Network
PSNN	Pi-Sigma Neural Network
SPNN	Sigma-Pi Neural Network
SSP	Sub Second Portion
t_i	The $i-th$ element of the input pattern
y	The output of output neuron
$\sigma(.)$	The activation function of output neuron as $\sigma(x) = 1/(1+e^{-x})$
w_{0j}	The threshold applied to the $j-th$ unit of hidden layer
Δw_{0j}	The adjusted value of the threshold w_{0j}
η	The learning-rate parameter
μ	The momentum value

Introduction

The GPS is a passive (one-way) satellite-based navigation system. It provides three-dimensional position and precise time transfer to users worldwide, twenty-four hours a day, in any weather and at virtually any location on the earth with a sufficiently clear view of the sky. One of the keys to obtaining high-accuracy position and time keeping with GPS is the use of redundant atomic clocks onboard GPS satellites and in the GPS control segment. These clocks provide the means for estimating satellite orbits and satellite clock correction parameters. These quantities are estimated on the ground by the control segment, results are transmitted to the satellites and then the data blocks containing the ephemeris model and clock correction terms are retransmitted to the users [1].

The GPS error sources can be grouped into three categories. The first set is due to control segment imperfections. The satellite ephemeris error is the difference between the actual position and velocity of a satellite and those predicted by the broadcast ephemeris model. This error is typically 1-2 m in the RMS sense. The satellite clock bias, the difference between the true clock and the satellite clock, also introduces about a 1-2 m RMS range error. In the past, civil users experienced an additional error due to SA, which intentionally dithered the clock to cause approximately a 22 m RMS error [2].

A second set of errors are introduced by the ionosphere and troposphere [3]. The ionosphere is a region of ionized gases which affect the speed of GPS signal propagation from a satellite to a receiver. The code phase measurements are delayed while the carrier phase measurements are advanced. Since this delay is inversely proportional to the signal frequency, dual frequency users can remove this error by themselves. Signal frequency users can reduce this delay by approximately 50% after utilizing the Klobuchar ionospheric model broadcast as part of the GPS navigation data. The resulting ranging error, proportional to the total electron content in the ionosphere, is about 1-5 m. The dry gases and water vapor composing the troposphere refract GPS signals and introduce an additional delay. The delay generally decreases with increasing elevation satellites. This tropospheric delay can be corrected using atmospheric models. If corrected based on average meteorological conditions, the resulting error is about 0.1-1 m [4].

The remaining errors are multipath and receiver noise [5]. Multipath errors are caused by the interfering signals reflected from surfaces. Since the code and carrier measurements are based on the sum of the direct and reflected signals, the ranging error depends on the strength of the reflected signal and the delay between direct and reflected signals. Multipath affects code measurements with a 1-5 m error and carrier measurements with a 1-5 cm RMS error. Adopting a multipath-limiting antenna, a narrow correlator receiver or carefully choosing an installation site for the antenna can reduce these errors. Finally, receiver

noise errors are due to thermal noise in the receiver front end, multi-access interference and signal quantization noise. The receiver noise introduces less then 0.5 m of code measurement error and about 1-2 mm of carrier phase measurement error. Table 1 shows GPS error sources.

Table 1. A summary of error size in GPS measurements [6]

Error sources	Error size
Satellite clock model	1-2 m (RMS)
Satellite ephemeris prediction	1-2 m (RMS)
Ionospheric delay	2-10 m in zenith direction
Tropospheric delay	2.3-2.5 m in zenith direction at sea level
Multipath	Code: 1-5 m Carrier:1-5 m
Receiver noise	Code: 0.5 m (RMS) Carrier: 1-2 mm (RMS)

Because of the above mentioned error sources in GPS measurements, modeling and prediction of GPS receivers timing errors is very important. In this paper, we estimate these errors using KFs and NNs. The KF in its various forms is clearly established as a fundamental tool for analyzing and solving a broad class of estimation problems. Its recursive structure avoids extensive calculations and data storage. Kalman filtering has undeniable advantages of algorithmic simplicity and relatively low computational intensity. Requirements for extremely accurate GPS measurements can only be achieved with third or fourth order filters. The key obstacle to the successful application of these higher order filters lies in the difficulty of choosing the Kalman gain coefficients. These coefficients govern the trade-off between the accuracy and the stability of the filter [7].

NNs have been proposed for time series prediction which involves prediction of the future patterns based on the patterns learnt from the past data. In these cases the network is designed to capture the pattern behaviour embedded in the past data. These ideas have been applied in several forecasting situations. Ideas based on time series prediction have also been exploited for identification of non-linear dynamical systems using NNs [8].

This paper presents five methods for GPS receivers timing errors estimation. The proposed methods validity is verified with experiments on actual data collection. This paper is organized as follows. Section 2 presents GPS applications in power systems. Section 3 describes the proposed prediction methods. Experimental results are reported with collected real data in section 4 and finally conclusions are presented in section 5.

Precise Timing Applications in Power Systems

Precise timing in power systems is one of the key technologies that will enable the development of new control systems and the monitoring required to maintain them. Some of these areas of potential development are described in the following paragraphs [9].

Sources of Synchronization

The GPS furnishes a common-access timing pulse, which is accurate to within 1 microsecond at any location on earth. For accurate acquisition of the timing pulse, only one of the satellites need be visible to the antenna. The experience with the availability and dependability of the GPS satellite transmission has been exceptionally good.

Phasor Measuring Units

PMUs using synchronization signals from the GPS satellite system have evolved into mature tools and are now being manufactured commercially. The GPS receiver provides the 1-PPS signal, and a time tag, which consist of the year, day, hour, minute, and second. The time could be the local time, or the UTC. The 1-PPS signal is usually divided by a phase-locked oscillator into the required number of pulses per second for sampling of the analog signals.

State Estimation

Modern electric utility centers use state estimators to monitor the state of the power system. The state estimator uses various measurements (such as complex powers and voltage and current magnitudes) received from different substations, and, through an iterative non-linear estimation procedure, calculates the power system state. The sate (vector) is a collection of all the positive sequence voltage phasors of network, and, from the time the first measurement is taken to the time when the state estimate is available, several seconds or minutes may have elapsed. Because of the time skew in the data acquisition process, as well as the time it takes to converge to a state estimate, the available state vector is at best an averaged quasi-steady-state description of the power system.

Improved Control

Power system control elements, such as generation excitation systems, HVDC terminals, variable series capacitors, SVCs, etc., use local feedback to achieve the control objective. With synchronized phasor measurements being brought to the controller location, it becomes possible to provide direct feedback from the angular difference between the two systems. Studies of this nature have shown that improved control performance is achieved when a model-based controller is replaced by one based upon feedback provided by the phasor measurement system.

Quasi-Traveling Wave Schemes

Quasi-traveling wave schemes compare only the relative phase of the charge in impedance at the inception of fault at the local end with a signal representing the relative change at the remote end. When a fault occurs, the instantaneous voltage will usually fall and the instantaneous current will rise; either quantity may be positive or negative at that time. The relative change between the two represents the change in impedance and the direction of the fault. The relay is triggered by a rate of change in the voltage and current and sends a directional signal. Trip decision times are short but must allow for transmission time of the carrier system, relative end-to-end phasing of the voltage/current is not normally critical.

Proposed Estimation Methods

Non-parametric approach of time series prediction is based on the assumptions that future behavior of a time series can be represented by a functional relationship of its previous observations. If $p\tau$ previous input data are given at the $n-th$ time step, for a time series data $x(n)$, the τ-step-ahead value $x(n+\tau)$ can be expressed as [8]:

$$x(n+\tau) = g[x(n), x(n-\tau), \cdots, x(n-(p-1)\tau)] \tag{1}$$

where $g(.)$ denotes a function that represents input-output relationship of non-parametric time series prediction process. The positive integer p is called embedding dimension and τ is time delay. Future value of a time series can be predicted by an output of a linear or non-linear function $g(.)$ of $p\tau$ previous input data.

Time series prediction methods can be classified into either one-step-ahead prediction or short-term prediction depending on the fact that predicted values are again used as input values. In one-step-ahead prediction, the predicted

value of future data $\hat{x}(n+\tau)$ is expressed by its previous p inputs with time delay τ of the data sequence as in equation (2):

$$\hat{x}(n+\tau) = g[x(n), x(n-\tau), \cdots, x(n-(p-1)\tau)] \tag{2}$$

In short-term or long-term prediction of time series data, predicted values of the data are again used as inputs for prediction of future data. Equation (3) shows short-term or long-term prediction of time series. The predicted value of future data $\hat{x}(n+\tau)$ is expressed according to the data previously predicted $\hat{x}(n), \hat{x}(n-\tau), \cdots, \hat{x}(n-(p-1)\tau)$. Therefore, long-term prediction of chaotic time series based on the data previously predicted is a difficult task since small initial error can cause large error accumulation effects in future values.

$$\hat{x}(n+\tau) = g[\hat{x}(n), \hat{x}(n-\tau), \cdots, \hat{x}(n-(p-1)\tau)] \tag{3}$$

The adequate embedding dimension p is determined for a specific time delay τ. For given τ, error performance measures are calculated for the training data and for the validation data. Error performance measures are defined as MSE calculated from the difference between the one-step-ahead prediction results and real data. For a given time series data, MSE are computed as one increases possible embedding dimension values for a fixed time delay. This process is repeated for training data and for validation data. The adequate embedding dimension corresponds to the embedding dimension whose error measure is minimized both for training and validation data. In this paper, five prediction methods will be discussed.

Modeling using Kalman Filter in Model I

The operation of a KF can be divided into two steps. In the first step the filter computes a prediction value of the state vector $\hat{X}_{m+1}^{-}$ and the respective covariance matrix P_{m+1}^{-}. It uses the following prediction equations [10]:

$$\hat{X}_{m+1}^{-} = A_m \hat{X}_m^{+} \tag{4}$$

$$P_{m+1}^{-} = A_m P_m^{+} A_m^T + Q_m \tag{5}$$

where A_m is the system matrix, which represents a given model and Q_m is the covariance matrix of the driving noise, representing the stochastic influences upon the state and the modeling uncertainties. In the second step the filter computes filtered state estimates and the corresponding covariance matrix. The filter equations are:

$$\hat{X}_m^{+} = \hat{X}_m^{-} + K_m(y_m - C_m \hat{X}_m^{-}) \tag{6}$$

$$P_m^{+} = P_m^{-} - K_m C_m P_m^{-} \tag{7}$$

where K_m is called Kalman gain and is obtained by the following equation:

$$K_m = P_m^{-} C_m^T (C_m P_m^{-} C_m^T + R_m)^{-1} \tag{8}$$

where y_m is the observation vector, C_m is the observation matrix and $R_m = E\{v_m v_m^T\}$ is the covariance of the Gaussian measurement noise v_m with $E\{v_m\} = 0$.

A KF requires the system model to be in state-space form. In this modeling, the transition matrix can be obtained using the time-varying AR model. AR is a popular model which is used in a discrete-time stochastic process. A time-varying AR model of order $AR(p)$ is mathematically described by [10]:

$$x(m) = -\sum_{i=1}^{i=p} a_i(m) x(m-i) + e(m) \tag{9}$$

where $x(m-i)$ is the output of the model and $e(m)$ is the noise value at time m for $m = 1,2,\cdots$. $a_i(m)$ for $i = 1,2,\cdots,p$ is the sets of parameters which describes the model structure. The parameter matrix $\theta(m) = \left[-a_1(m) \ -a_2(m) \ \cdots \ -a_p(m)\right]^T$ may be estimated using the least-squares method as:

$$\theta = (F^T F)^{-1} F^T X \tag{10}$$

where for $m = l, l+1, \cdots, N$:

$$X = \begin{bmatrix} x(l) \\ x(l+1) \\ \vdots \\ x(N) \end{bmatrix}, \ F = \begin{bmatrix} x(l-1) & x(l-2) & \cdots & x(l-p) \\ x(l) & x(l-1) & \cdots & x(l-p+1) \\ \vdots & \vdots & \vdots & \vdots \\ x(N) & x(N-1) & \cdots & x(N-p+1) \end{bmatrix} \tag{11}$$

Once the model parameters $a_i(m)$ are known, the calculation of $x(m)$ for an arbitrary m can be accomplished by [10]:

$$x(m) = -\sum_{i=1}^{i=p} a_{p-i+1}(m) x(m-i) \tag{12}$$

We can represent the above equations in state-space form using the following:

$$\begin{aligned} x_1(m) &= x(m-p) \\ x_1(m+1) = x_2(m) &= x(m-p+1) \\ x_2(m+1) = x_3(m) &= x(m-p+2) \\ &\vdots \\ x_p(m+1) = x_{p+1}(m) &= x(m) \end{aligned} \tag{13}$$

Finally we are led to the following canonical controllable state-space representation:

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix}_{m+1} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -a_1 & -a_2 & -a_3 & \cdots & -a_p \end{bmatrix}_m \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix}_m \tag{14}$$

The observation matrix is chosen as:

(15) $$C=\begin{bmatrix}0 & 0 & 0 & \cdots & 1\end{bmatrix}$$

Also, $\hat{P}_0^-$ and $\hat{X}_0^-$ are initialized as the following equations:

(16) $$\hat{P}_0^- = 30^2\begin{bmatrix}1 & 0 & 0 & \cdots & 0\\ 0 & 1 & 0 & \cdots & 0\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ 0 & 0 & 0 & \cdots & 1\end{bmatrix}$$

(17) $$\hat{X}_0^- = \begin{bmatrix}x(N-p)\\ x(N-p+1)\\ \vdots\\ x(N-1)\end{bmatrix}_{n\times 1}$$

GPS time is a continuous measurement of time from an epoch started at January 6, 1980 at midnight (0 hours, 0 minutes, 0 seconds) UTC. GPS time is often stated in a number of weeks and seconds from the GPS time epoch. GPS time does not introduce leap seconds and so is ahead of UTC by an integer number of seconds (10 seconds as of 1 July 1994, 11 seconds at 1 January 1996). GPS time is steered by the master control site to be within one microsecond (less leap seconds) of UTC. The GPS navigation message contains parameters that allow the GPS user to compute an estimate of the current GPS-UTC sub-microsecond difference as well as the number of leap seconds introduced into UTC since the GPS epoch. The GPS receivers timing error $x(n)$ is difference between the two sequence UTOD at time n, i.e, $x(n)=UTOD(n)-UTOD(n-1)$.

A subset of available actual data is used to construct the proposed KF parameters in this method. They were optimized by considering a trade-off between GPS receivers timing errors estimation accuracy and CPU time of algorithm. After optimization, p and N are 3 and 5, respectively.

Modeling using Kalman Filter in Method II

The positioning and timing errors sources in GPS are common. In this method, the GPS receivers positioning errors are modeled first. The next, the system matrix is obtained from them. Finally, GPS receivers timing errors are estimated using KF. In method II, the GPS receiver position is known and we estimate its timing errors epoch by epoch. Fig. 1 shows the flow chart of modeling using KF in method II.

A subset of available actual data is used to construct KF order in this method. A three KF order can be used as optimal order for GPS receivers timing errors prediction.

Modeling using Recurrent Neural Network

RNNs are NNs with one or more feedback connections that can be of local or global nature. Feedback allows the RNNs to acquire state representations, making them appropriate devices for different dynamic applications such as forecasting or modeling non-linear systems. In RNNs both feedforward and feedback (recurrent) connections between neurons are allowed. As with ordinary MLPs, recurrent MLPs can perform any non-linear mapping, but the difference is that the response to an input from a RNN is now based on all previous inputs, as these are used in feedback connections. Nonetheless, the RNN is a dynamic system, with the activations of the neurons with feedback connection being the state of the system. However, training RNN to perform a certain task is known to be a very difficult problem [11]. Fig. 2 shows a RNN structure.

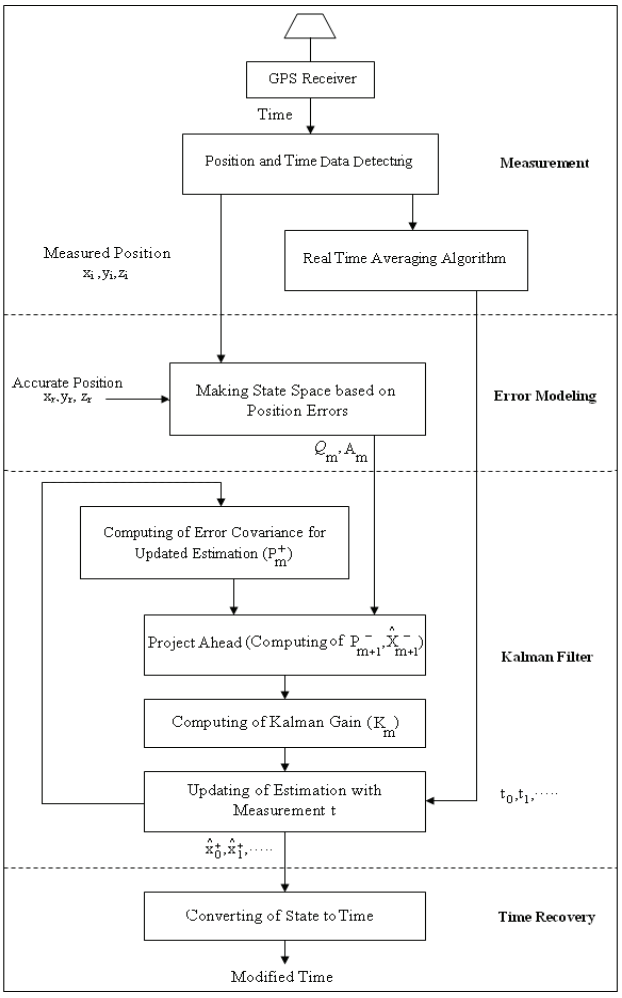

Fig. 1. Flow chart of modeling using KF in method II

In Fig. 2, $S_j(m)$ is the sum of inputs to the $j-th$ recurrent neuron and $X_j(m)$ is the output of the $j-th$ recurrent neuron. W^I, W^D and W^O are input, recurrent and output weights vectors, respectively. A summary of the learning algorithm for Fig. 2 is provided in the following [12]:

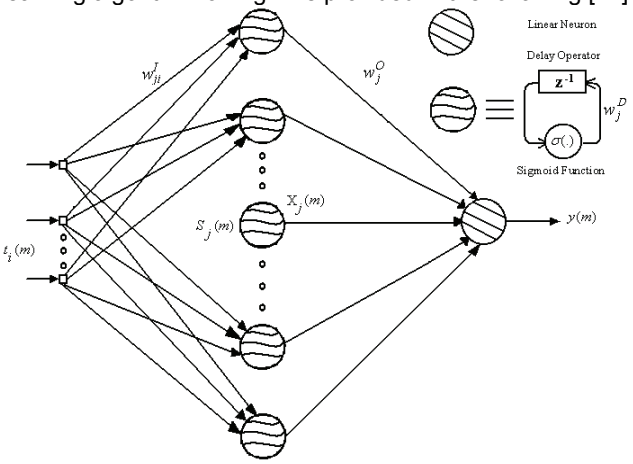

Fig. 2. RNN architecture with $(p,q,1)$ structure

Step 1: Weights Vector Initialization

Set all of the weights of the network to small random numbers that are uniformly distributed.

Step 2: Forward Computation

$$S_j(m) = \sum_{i=1}^{i=p} w_{ji}^I(m) t_i(m) + w_j^D(m) X_j(m-1) \tag{18}$$

$$X_j(m) = \sigma_j[S_j(m)] \tag{19}$$

$$y(m) = \sum_{j=1}^{j=q} w_j^O(m) X_j(m) \tag{20}$$

Step 3: Learning Process

$$\Delta w_j^O(m) = \eta[y(m) - d(m)] X_j(m) \tag{21}$$

where $d(m)$ is the desired response.

$$\begin{aligned} &\Delta w_j^D(m) = \eta[y(m) - d(m)] w_j^O(m) \frac{\partial X_j(m)}{\partial w_j^D(m)} \\ &\frac{\partial X_j(m)}{\partial w_j^D(m)} = X_j(m)[1 - X_j(m)][X_j(m-1) \\ &+ w_j^D(m) \frac{\partial X_j(m-1)}{\partial w_j^D(m)}] \\ &\frac{\partial X_j(0)}{\partial w_j^D(0)} = 0 \end{aligned} \tag{22}$$

$$\begin{aligned} &\Delta w_{ji}^I(m) = \eta[y(m) - d(m)] w_j^O(m) \frac{\partial X_j(m)}{\partial w_{ji}^I(m)} \\ &\frac{\partial X_j(m)}{\partial w_{ji}^I(m)} = X_j(m)[1 - X_j(m)][t_i(m) \\ &+ w_j^D(m) \frac{\partial X_j(m-1)}{\partial w_{ji}^I(m)}] \\ &\frac{\partial X_j(0)}{\partial w_{ji}^I(0)} = 0 \end{aligned} \tag{23}$$

Step 4: Iteration

Increment time m by one unit and go back to step 2.

Standard supervised BP learning methodology is followed in these experiments with RNNs. A subset of available actual data is used to construct training samples for the network. Training of a RNN involves obtaining optimal values for the learning rate, estimating the number of hidden layers and the number of nodes in each layer. The overall error is tracked until a minima is obtained by altering the fore-mentioned parameters. A trained network which has learned the sequential information in the training set can then be used in predicting GPS receivers timing errors. A 3-2-1 architecture with 3 inputs, one hidden layer of 2 nodes and one output is used.

Modeling using Pi-Sigma Neural Network

MLPs have been successfully used in time series prediction, however due to their multiple layer structure; they utilize computationally expensive training algorithms (such as the BP error) and can get stuck in local minima. In an attempt to overcome the problems associated with use of MLPs, HONNs have been employed with great success. HONNs make use of non-linear interactions between the inputs, thus functionally expanding the input space into another space, where linear separability, or reduction in the dimension of the non-linearity is possible. However, HONNs suffer from the combinatorial explosion of the high-order terms and demonstrate slow learning, when the order of the network becomes excessively high. A simple yet efficient alternative to HONNs is the PSNNs. PSNNs are constructed of linear summing units with the output layer being a single product unit with a non-linear transfer function. The weights from the summing units to the product units are fixed at unity, which implies that the summing units layer is not hidden. The degree of a PSNN equals the number of summing units in the first layer [13]. Fig. 3 shows a PSNN architecture.

The inputs are weighted and fed to a layer of q linear summing units, where q is the desired order of the NN. w_{ji} is an adjustable weight from input t_i to the $j-th$ summing unit, Δw_{ji} is the adjusted value of the weight w_{ji}, s_j the output of the $j-th$ summing unit of hidden layer and r the net internal activity level of output neuron.

To achieve a desirable set of synaptic weights to a pre-defined network architecture, a training process in needed. A training process is generally based on an optimization scheme to adjust the network parameters (mainly, the weights) in relation to a set of input-to-output to be matched by the NN model (supervised learning scheme). The BP algorithm based on a gradient descent technique has been widely applied for general NN training. A BP employs a two-pass weighted learning algorithm known as the generalized delta rule. In a forward pass through the network, an error is detected; the measured error is then propagated backward through the network while weights are adjusted to reduce the overall error. This iterative process that the network goes through in reducing the overall error is known as gradient descent. The training equations are provided in the following [13]:

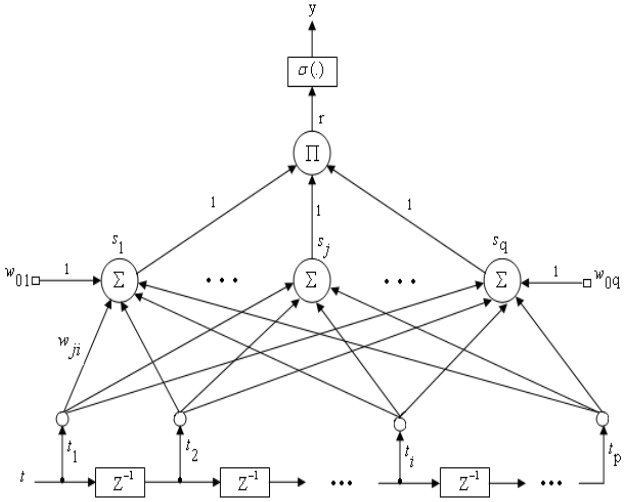

Fig. 3. PSNN architecture with $(p,q,1)$ structure

1: Weights Vector Initialization

Set all of the synaptic weights and threshold of the networks to small random numbers that are uniformly distributed.

Step 2: Forward Computation

$$s_j(m) = \sum_{i=1}^{i=p} w_{ji}(m) t_i(m) + w_{0j}(m) \;;\; j = 1 \;\; to \;\; q \tag{24}$$

(25) $$r(m) = \prod_{j=1}^{j=q} s_j(m)$$

(26) $$y(m) = \delta[r(m)] = \delta\left(\prod_{j=1}^{j=q} s_j(m)\right)$$

Step 3: Learning Process

(27) $$\Delta w_{0j}(m+1) = \eta[y(m) - d(m)]y(m)[1 - y(m)] \times [\prod_{l=1, l\neq j}^{l=q} s_l(m)] + \mu\Delta w_{0j}(m)$$

(28) $$\Delta w_{ji}(m+1) = \eta[y(m) - d(m)]y(m)[1 - y(m)] \times [\prod_{l=1, l\neq j}^{l=q} s_l(m)]t_i(m) + \mu\Delta w_{ji}(m)$$

or:

(29) $$\Delta w_{ji}(m+1) = [\Delta w_{0j}(m+1) - \mu\Delta w_{0j}(m)]t_i(m) + \mu\Delta w_{ji}(m)$$

Step 4: Iteration

Increment time m by one unit and go back to step 2.

An optimal PSNN architecture with 5 inputs, one hidden layer of 3 nodes and one output is used predicting GPS receivers timing errors.

Modeling using Sigma-Pi Neural Network

SPNNs have been shown to be able to uniformly approximate any continuous functions on a compact set in a multi-dimensional input space. SPNN has a very similar structure to the PSNN. Their difference is that hidden layer output is the product of the input terms and the network output is the sum of these products. It also has a single layer of adaptive weights, but in this network the adaptive weights are in the second layer. Fig. 4 shows a SPNN architecture. r_j is the output of the $j-th$ product unit of hidden layer, s is the net internal activity level of output neuron, w_j is the connecting weight between the $j-th$ product unit of hidden layer and the input of output neuron and Δw_j is the adjusted value of the weight w_j.

The BP algorithm based on a gradient descent technique is used for SPNN training. The training equations are provided in the following [13]:

Step 1: Weights Vector Initialization

Set all of the synaptic weights and threshold of the networks to small random numbers that are uniformly distributed.

Step 2: Forward Computation

(30) $$r_j(m) = \prod_{i=1}^{i=p} t_i(m) + w_{0j}(m) \;;\; j = 1 \;\; to \;\; q$$

(31) $$s(m) = \sum_{j=1}^{j=q} w_j(m) r_j(m)$$

(32) $$y(m) = \delta[s(m)] = \delta\left(\sum_{j=1}^{j=q} w_j(m) r_j(m)\right)$$

Step 3: Learning Process

(33) $$\Delta w_{0j}(m+1) = \eta[y(m) - d(m)]y(m) \times [1 - y(m)]w_j(m) + \mu\Delta w_{0j}(m)$$

(34) $$\Delta w_j(m+1) = \eta[y(m) - d(m)]y(m) \times [1 - y(m)]r_j(m) + \mu\Delta w_j(m)$$

or:

(35) $$\Delta w_j(m+1) = [\Delta w_{0j}(m+1) - \mu\Delta w_{0j}(m)] \times \frac{r_j(m)}{w_j(m)} + \mu\Delta w_j(m)$$

Step 4: Iteration

Increment time m by one unit and go back to step 2.

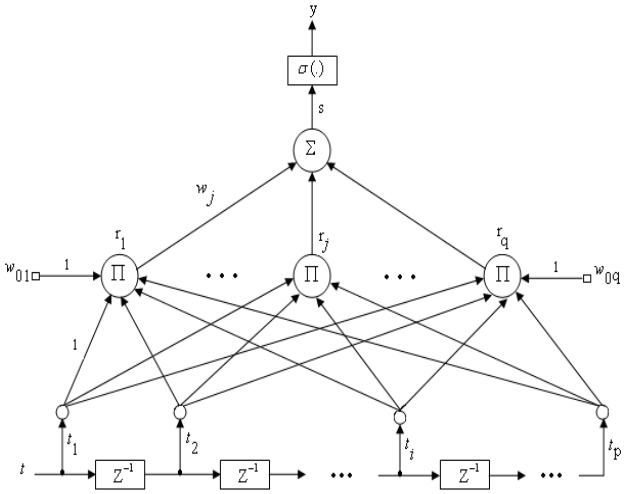

Fig. 4. SPNN architecture with $(p, q, 1)$ structure

An optimal SPNN architecture with 4 inputs, one hidden layer of 3 nodes and one output is used predicting GPS receivers timing errors.

Experimental Results

To evaluate the performance of the five methods proposed, data sets were collected on the building of the Control and Fuzzy Logic Research Lab at Iran University of Science and Technology. The experimental system consists of a low-cost Rockwell GPS OEM receiver [14], a low-cost active GPS antenna and a Pentium PC for data processing of each proposed prediction method. The representation of data, the training files and the test files were the same for all five experiments. Fig. 5 shows block diagram of data collection and processing system of designed and implemented in this research. Figures 6 to 10 show $SSP-UTOD$ predictions for 200 test using the five methods.

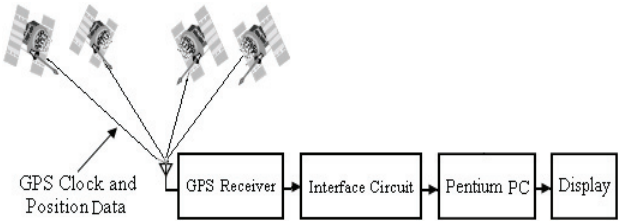

Fig. 5. Data collection and processing system

For simulations of the proposed methods, an application program was written using “Microsoft Visual Basic”. In order to evaluate the performance of the five models, the RMS measure was used to quantify of closeness between predicted and observed values [15,16]. RMS value is computed using:

$$RMS = \sqrt{\frac{1}{M}\sum_{i=1}^{i=M}[dt_{\Pr edicted}(i) - dt_{\mathrm{Re}\,al}(i)]^2} \tag{36}$$

where M is number of tests. The optimal selection of proposed methods parameters was based on the experimental results. Increasing the KF order, the number of hidden layer neurons and also training patterns improves the proposed methods performance. Increasing the KF order, the number of hidden layer neurons and training patterns increases the memory for software implementation and also the structure complexity for hardware implementation. Therefore, a trade-off in selecting the KF order, the number of hidden layer neurons and training patterns between CPU time and accuracy of methods is required.

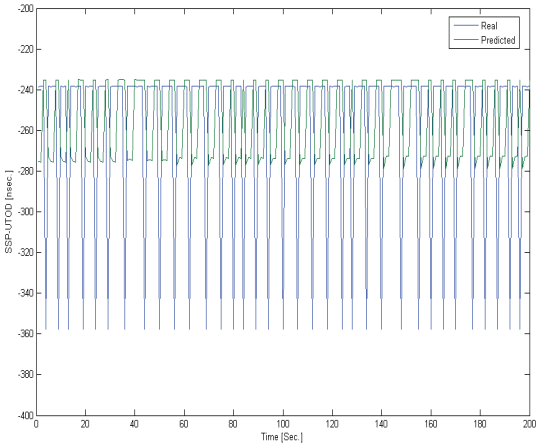

Fig. 6. Results of 200 $SSP-UTOD$ predictions using proposed KF in method I

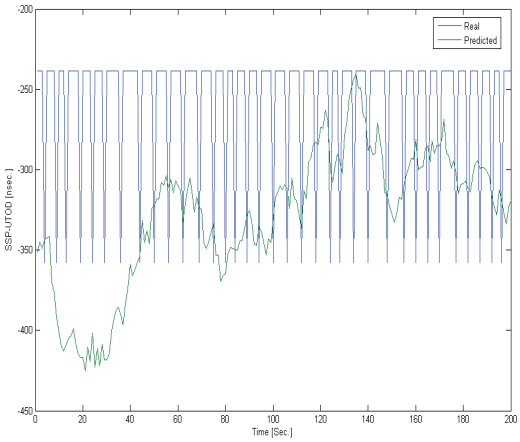

Fig. 7. Results of 200 $SSP-UTOD$ predictions using proposed KF in method II

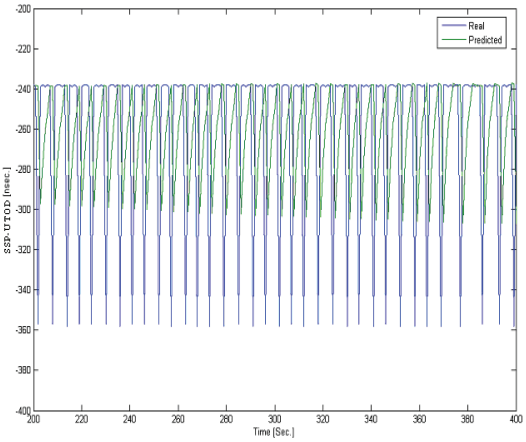

Fig. 8. Results of 200 $SSP-UTOD$ predictions using proposed RNN

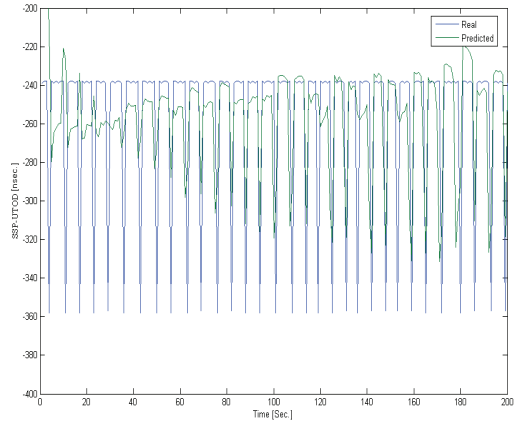

Fig. 9. Results of 200 $SSP-UTOD$ predictions using proposed PSNN

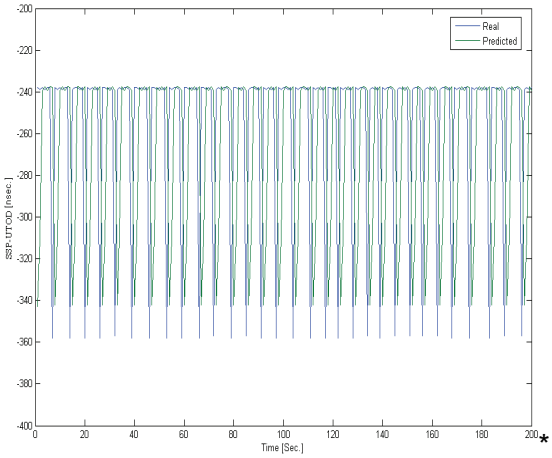

Fig. 10. Results of 200 $SSP-UTOD$ predictions using proposed SPNN

Table 2 shows statistical significance characteristics of prediction errors for 500 timing test data using the five proposed methods.

Table 2 demonstrates the PSNN has better accuracy for GPS timing errors prediction, since RMS prediction error in PSNN model is lower than other models.

There are a few papers that predicts GPS receivers timing errors using NNs. Advantages of the SPNN and PSNN proposed in this paper are that they are simple, low cost, and easy to design. They have the low structure complexity for hardware implementation. They also require low memory for software implementation.

Table 2. Prediction Error of The Proposed Methods in Nanoseconds

Parameters	KF in method I	KF in method II	RNN	PSNN	SPNN
Max	124.8	223.6	120.9	146.6	127.8
Min	-46.6	-84.7	-119.0	-100.9	-24.7
Average	3.2	27.8	-2.6	0.87	0.95
Variance	6.8	9.6	6.8	2.9	4.0
Standard deviation	2.6	3.1	2.6	1.7	2.0
RMS	**58.6**	**68.6**	**57.2**	**37.3**	**50.0**

Conclusion

Accurate timing using GPS can revolutionize the field of monitoring, protection, and control of power systems. It is with great excitement that we look for other applications, not yet thought of, that can advance the state of the art in electric power engineering. The past few years have witnessed increasing interest in synchronized accurate timing and how they may be used for various power system applications. The development of new types of computer-based hardware and the completion of the GPS of satellites provide the components needed for true synchronized monitoring systems. GPS time synchronization enables the accurate time tag of each recorded data sample to better than 1 microsecond accuracy.

This paper presented five methods which were designed and implemented to predict GPS receivers timing errors. The KFs and NNs based prediction have been successfully conducted. Two types of KF and three types of NN (including RNN, PSNN and SPNN) predictors were focused on. The performances have been explored and discussed. An experimental test setup was designed and implemented for this purpose. The experimental results obtained from a C/A-code single-frequency GPS receiver strongly support the potential of PSNN to provide best performance. Accuracy of the obtained results was very good; so that GPS timing RMS error reduced to less than 38 nanoseconds.

REFERENCES

[1] B. W. Parkinson, J. J. Spilker Jr, P. Axelrad and P. Enge, Global Positioning System: Theory and Applications, *The American Institute of Aeronautics and Astronautics*, 1996.

[2] K. L. V. Dyke, The World after SA: Benefits to GPS Integrity, *IEEE Conference on Position, Location and Navigation*, pp.387-394, 2000.

[3] M. R. Mosavi, Neural Networks-based Single-Frequency GPS Receivers Ionospheric Time-Delay Modeling, *Asian Journal of Geoinformatics,* Vol.9, No.4, pp.35-40, 2009.

[4] O. Øvstedal, Absolute Positioning with Single-Frequency GPS Receivers, *Journal of GPS Solutions, Vol.5, No.4,* pp.33-44, 2002.

[5] M. R. Mosavi, Estimation of Pseudo-Range DGPS Corrections using Neural Networks Trained by Evolutionary Algorithms, *Journal of Review of Electrical Engineering,* Vol.5, No.6, 2010.

[6] K. D. McDonald, The Modernization of GPS: Plans, New Capabilities and the Future Relationship to Galileo, *Journal of Global Positioning System, Vol.1, No.1,* pp.1-17, 2002.

[7] M. R. Mosavi, A Comparative Study between Performance of Recurrent Neural Network and Kalman Filter for DGPS Corrections Prediction, *IEEE Conference on Signal Processing (ICSP 2004), Vol.1,* pp.356-359, 2004.

[8] M. S. Kim and S. G. Kong, Parallel-Structure Fuzzy Systems for Time Series Prediction, *Journal of Fuzzy Systems, Vol.3, No.1,* pp.331-340, 2001.

[9] A. G. Phadke, Synchronized Phasor Measurements in Power Systems, *IEEE Computer Applications in Power,* pp.11-15, 1993.

[10] M. R. Mosavi, Comparing DGPS Corrections Prediction using Neural Network, Fuzzy Neural Network and Kalman Filter, *Journal of GPS Solutions, Vol.10, No.2,* pp.97-107, 2006.

[11] F. D. Marques, L. F. R Souza, D. C. Rebolho, A. S. Caporali, E. M. Belo and R. L. Ortolan, Application of Time-Delay Neural and Recurrent Neural Networks for the Identification of a Hingeless Helicopter Blade Flapping and Torsion Motions, *Journal of the Brazilian Society of Mechanical Sciences and Engineering, Vol.27, No.2,* pp.97-103, 2005.

[12] J. Sang, K. Kubik and L. Zhang, Prediction of DGPS Corrections with Neural Networks, *IEEE Conference on Knowledge-based Intelligent Electronic Systems,* pp.355-361, 1997.

[13] M. R. Mosavi, A Practical Approach for Accurate Positioning with L1 GPS Receivers using Neural Networks, *Journal of Intelligent and Fuzzy Systems, Vol.17, No.2,* pp.159-171, 2006.

[14] MicroTracker LP Designer's Guide, *Rockwell International Corporation, GPS-22,* 1995.

[15] A. Indriyatmoko, T. Kang, Y. J. Lee, G. I. Jee, Y. B. Cho and J. Kim, Artificial Neural Networks for Predicting DGPS Carrier Phase and Pseudo-Range Correction, *Journal of GPS Solutions, Vol.12, No.4,* pp.237-247, 2008.

[16] I. Sadinezhad and M. Joorabian, An Adaptive Precise One End Fault Location in Transmission Lines Based on Hybrid Complex Least Error Squares Algorithm and Adaptive Artificial Neural Networks, *Journal of Review of Electrical Engineering,* Vol.3, No.5, pp.803-810, 2008.

Author: *Mohammad-Reza Mosavi received his B.S., M.S., and Ph.D. degrees in Electronic Engineering from Department of Electrical Engineering, Iran University of Science and Technology (IUST), Tehran, Iran in 1997, 1998, and 2004, respectively. He is currently faculty member of Department of Electrical Engineering of IUST as associate professor. He is the author of about 110 scientific publications on journals and international conferences. His research interests include Artificial Intelligent Systems, Global Positioning Systems, Geographic Information Systems and Remote Sensing. Email: M_Mosavi@iust.ac.ir*