SSC04-VIII-5

Neural Network Based Orbit Propagation for Small Satellite Missions

Liu Fang
Regional Operations & Engineering Division
Singapore Telecommunications Limited
500 Rifle Range Road, #02-00
Bukit Timah Satellite Earth Station, Singapore 588397
liufang@singtel.com

N. Nagarajan
School of Electrical and Electronic Engineering
Nanyang Technological University
Nanyang Ave, Singapore 639798; (65) 6790 6860
enara@ntu.edu.sg

ABSTRACT: The Global Positioning System (GPS) receivers are being widely utilized for low earth navigation in space. These receivers provide the orbital information in real time while the attitude sensors provide real time attitude data. For full autonomy of a satellite, any failures which may occur due to different sources, such as GPS receiver, attitude sensors are to be detected autonomously and suitable remedial actions need to be taken to continue the satellite operations. Towards achieving autonomy, GPS receivers are widely used onboard the satellite for orbit information. In case of a failure or outages of the GPS receiver data, an Artificial Neural Network (ANN) based back-up scheme is proposed in this research to handle these failures in orbit determination. ANN has the advantage of no requirement of system modeling. They are employed in this research to detect attitude and orbit sensor failures in view of their good performance and flexibility in implementing onboard a LEO small satellite.

INTRODUCTION

An Artificial Neural Network (ANN) is an information-processing system that has certain performance characteristics in common with biological neuron networks.[1] When the network is stimulated by the environment, it undergoes changes as a result of stimulation. The network responds to the environment in a new way after the occurrence of the change. It is more suitable for a complex system. ANN is utilized in many fields including detecting sensor failures and recovering the lost measurements from a group of redundant sensors.[2-4] It can be proposed as an estimator for replicating sensor data taking advantage of the properties such as applicability to nonlinear systems parallel distributed processing and on-line learning ability. The architecture and the learning methods affect the ANN's performances.[2-5]

Once a satellite is injected into the orbit, its orbital parameters are used for further prediction of orbital position and velocity as a function of time, which are in turn used for various mission operations like satellite tracking, imaging operations, payload data processing, orbital maneuvers etc. While one desires to have a perfect Keplerian orbit, in reality the orbital path changes due to various perturbation forces like atmospheric drag, solar radiation pressure, Earth's oblateness etc and this leads to the requirement of frequent updating of the orbital parameters. Presently, the most popular form of the orbital parameters is the Two-Line-Element (TLE) followed by NORAD. With the increasing interest in small satellite constellations, the TLEs are attractive for mission management, as these parameters are regularly updated and made available on Internet. As the number of small satellites is growing, there is also a growing interest in onboard autonomy.

As a step towards onboard autonomy, it is essential to generate orbit information onboard the satellite, to carry out the operations mentioned earlier. While a Global Positioning System (GPS) receiver is an obvious solution to this, a back-up strategy is to have an orbit predictor based on the TLE set. As a back-up orbit propagator, this research work addresses the feasibility of using a Neural Network based orbit predictor for handling any failure in the output of a GPS receiver.

In the present research work, the orbital information like mean-motion (η), inclination (*i*), epoch mean-anomaly (M_o), mean rate of motion ($\dot{\eta}$) from the Two Line Element (TLE) set are used as inputs and the high accuracy orbital position information obtained from the model-based propagator (SGP4) is used to train an Artificial Neural Network (ANN). This process is performed on a ground computer. The trained network can then be used to find the position of a satellite, in terms of latitude and longitude, as a function of time. Thus a simple neural network, instead of complex spherical trigonometry equations, is employed to generate the geographical information of a satellite at any given time. The neuron weights obtained from training could be up-linked to the satellite On Board Computer (OBC), wherein an exact replica of the neural network will be implemented in C code. Other forms of implementation include realization of neural network using FPGA in case of large number of neurons. The onboard neural network then calculates the current position of the satellite based on the onboard clock. The paper concludes by proposing an onboard strategy whereby the output of a neural network based orbit propagator can be used to formulate a failure detection scheme for a GPS receiver and thus serve as a back-up orbit propagator in case of contingencies.

ORBIT PREDICTION

The Two Line Element (TLE) set essentially describes the orbital information at a given epoch.[6] This epoch information is used by orbit prediction software like STK, Winorbit to predict the satellite position in Earth Centered Earth Fixed (ECEF) frame at any given instant of time. Figure 1 shows the latitude, longitude and altitude data thus generated from the TLE set for the Low Earth Orbit (LEO) satellite ADEOS whose orbital inclination is about 98.4 deg. It can be observed from Figure 1 that the latitude (ϕ) and altitude (h) patterns repeat for every orbit. The longitude (λ) history has a repeating pattern superimposed on a secular shift of longitude ($\Delta\lambda$) between successive orbits. After removing the secular shift ($\Delta\lambda_{avg}$) the differential longitude ($\delta\lambda$) (longitude with respect to the recent ascending node crossing longitude) shows a repeating pattern.

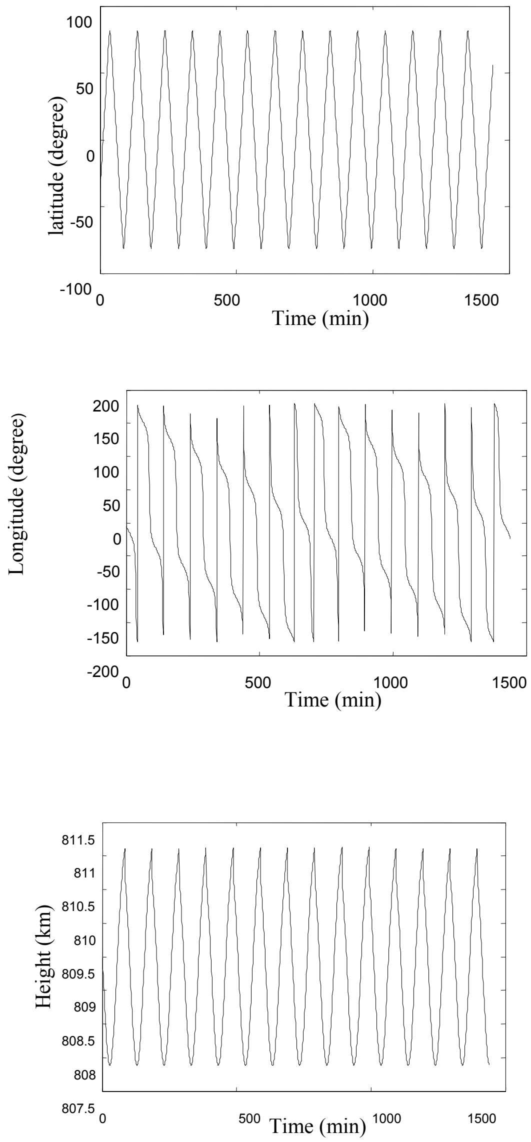

Figure 1. Predicted Satellite Position

Thus, the orbital position at any time t is,

$$\begin{aligned} \phi_t &= \phi(T+\tau) \\ \lambda_t &= \lambda_0 - N \bullet \Delta\lambda_{avg} + \delta\lambda(T+\tau) \\ h_\tau &= h(T+\tau) \end{aligned} \qquad (1)$$

where T is the orbital period and τ is the time relative to the recent ascending node crossing and $\Delta\lambda_{avg}$ is the longitude shift per orbit. The signatures of ϕ, $\delta\lambda$ account for short periodic effects of orbital perturbation that are modeled by complex trigonometric functions. In this work, ϕ, $\delta\lambda$, and h are used to train simple neural networks which are then used to retrieve the longitude and latitude given by (1). Thus, a simple neural network is formed instead of using the complex spherical trigonometry equations for orbit prediction. Due to the presence of atmospheric drag effects, the trained network will be accurate only for one or two days, after which the network may have to be trained with new data.

The neuron weights (w_i) and biases (b_i), together with the constants Q (including Mean Orbital Period T_p longitude at epoch λ_0, longitude shift per orbit $\Delta\lambda_{avg}$, epoch time t_0, minimum and maximum values of time, latitude, longitude and altitude for data normalization - $\tau_{\min}$, $\tau_{\max}$, $\phi_{\min}$, $\phi_{\max}$, $h_{\min}$ $h_{\max}$, $\delta\lambda_{\min}$ and $\delta\lambda_{\max}$), obtained during training are the parameters to be uploaded to the satellite's On Board Computer (OBC), wherein an exact replica of the neural network will be implemented. The onboard neural network then calculates the current position of the satellite based on the onboard time The different steps are outlined in Figure 2.

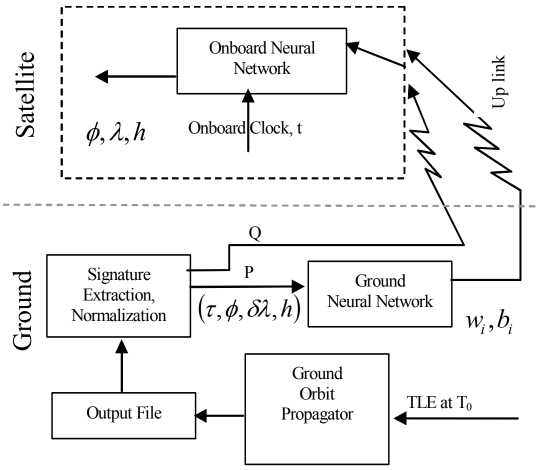

Figure 2. Proposed ANN scheme for orbit prediction

NEURAL NETWORK DESIGN APPROACH

The primary advantage of a neural network is that, it is made of inter-connected neurons represented by weight and bias parameters and the output of the network is a linear combination of these weight and bias parameters. It is shown in literature that a simple neutral network of one hidden layer is sufficient to represent a sinusoid.[7] Using this as a guiding factor, this research investigates the design and simulation of a neural network which would be trained by the data generated as explained in the previous section. The data set consisting of ϕ, $\delta\lambda$ and h is extracted from the orbit file and is used for training the network. The other parameters like $\Delta\lambda_{avg}$, T, which are also extracted from the orbit file, are used later in generating ϕ, λ and h from the trained neural network.

Three neural networks are employed in this research one each for latitude, longitude and height. The neural network model consists of three layers of neurons (input, hidden and output) connected to each other by connection weights. Each neuron has an activation value. The connections between neurons have associated weights. In simulation, the neural network computation is divided into two phases: learning phase and recall phase.

During the learning phase, the topology and weights of the network are determined from a labeled set of samples using a training rule known as Bayesian Regularization Backpropagation algorithm. With this algorithm, the weight and bias values are updated according to Levenberg- Marquardt optimization. It minimizes a combination of squared errors and weights, and then determines the correct combination so as to produce a network that generalizes the training set well by reducing the Sum Square Error (SSE). As a first step the data points are normalized and then employed to train the neural networks separately. Different neuron numbers decide the structures of the networks and affect the performances. The relations between the neuron numbers and the performance of the system will be shown later, together with simulation results.

In the recall phase (i.e. the simulation phase of a neural network), the activation values of the neurons from the output layer are computed according to the weight values computed in the learning phase. The output data of the neural networks are restored as the prediction values. Figure 3 shows the flow diagram of this process (net1 for latitude, net2 for longitude, and net3 for height). The various blocks and the interconnections shown in the figure, form the block 'Onboard Neural Network' of Figure 2.

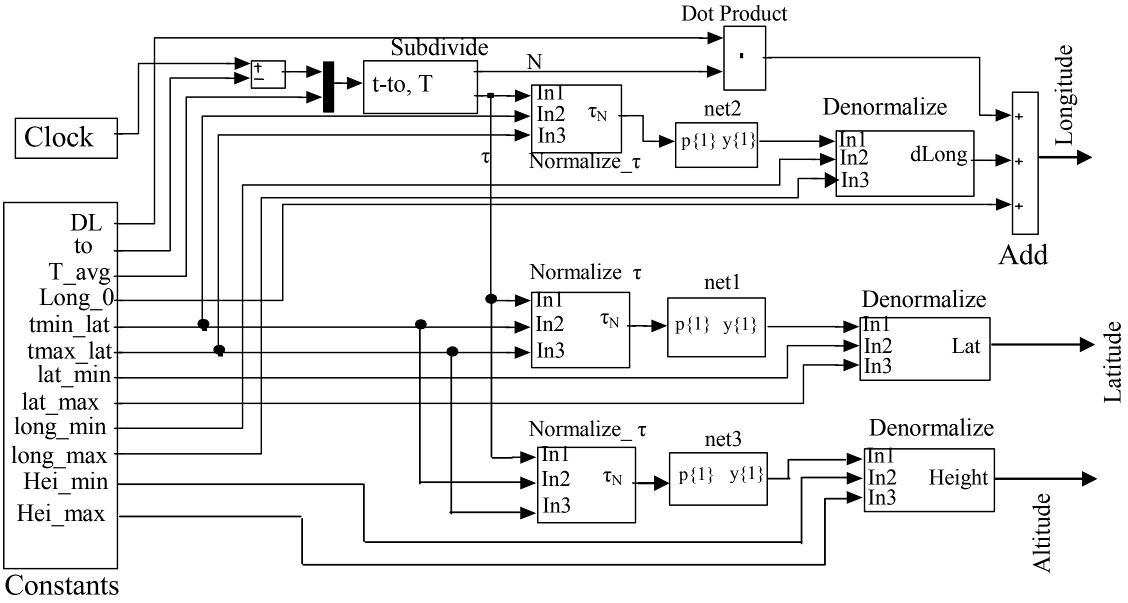

Figure 3. Diagram of the recall phase

RESULTS OF NEURAL NETWORKS

In order to analyze the size of the neural network, TLE data from six satellites orbiting with different inclination are utilized to train and simulate the network system. The orbital information of the first 24 hours is used to train the network and the data of the next 24 hours is used to assess the network performance. For a given satellite's orbit, training is done with different sizes of network, and the one that results in a SSE < 1e-4 with reasonably small number of neurons (no more than 10) is chosen as the appropriate network size. Table.1 shows the performance of neural networks for latitude and longitude predictions for various orbital inclinations. As a back-up system, the performances are fairly good. For a three-layer network, if there are N neurons, the total number of parameters (bias and weights) to be trained are 3N+1. In our simulations, we have utilized the matlab function TRAINBR which outputs the number of the effective parameters p (< 3N+1) that are useful in the training. This helps to assess the actual size of the neural network. Table.2 shows the performance of the neural network for height prediction. The accuracy of 2 km is also enough for a back-up strategy.

FAILURE DETECTION SCHEME

The failure detection scheme is based on the comparison between the position vectors calculated from the ANN and GPS receiver. The position vector in ECEF is given by,

Satellite Name	Inclination (degree)	Net ID	Neuron Number (N)	SSE	Number of parameters (p)	Minimum Error (deg)	Maximum Error (deg)	STD of Errors (deg)
ADEOS	98.4284	Net1	10	3.687e-5	26	-0.1634	0.1857	0.0538
		Net2	7	9.981e-6	21	-0.2492	0.2315	0.0660
OKEAN4	82.5406	Net1	10	3.634e-5	26	-0.1561	0.1929	0.0604
		Net2	9	1.414e-5	24	-0.4961	0.1677	0.0775
KO23	66.0808	Net1	8	2.068e-5	20	-0.1153	0.1267	0.0590
		Net2	8	9.563e-6	21	-0.0847	0.2730	0.0650
ISS	51.5901	Net1	7	2.482e-5	17	-0.1272	0.1158	0.0394
		Net2	5	7.306e-6	13	-0.1037	0.3461	0.0754
ROCSAT1	34.9826	Net1	6	4.574e-5	14	-0.0768	0.1427	0.0420
		Net2	5	2.708e-6	13	-0.1615	0.1375	0.0613
PO34	28.4627	Net1	5	9.002e-5	14	-0.1118	0.0832	0.0392
		Net2	5	2.853e-6	14	-0.1335	0.0584	0.0364
Net1: Latitude network; Net2: Longitude Network ; p < 3N+1 as decided by the Matlab training function TRAINBR								

Table 1. Performance of neural networks for latitude and longitude predictions

Satellite Name	Inclination (degree)	Neuron Number (N)	Minimum Error (km)	Maximum Error (km)	STD of Errors (km)
ADEOS	98.4284	7	1.0000	1.0000	0.2389
OKEAN4	82.5406	4	-2.1975	2.2075	0.9958
KO23	66.0808	3	-1.2518e-7	2.7511e-7	3.8515e-8
ISS	51.5901	6	-1.4430	1.0000	0.5296
ROCSAT1	34.9826	3	-2.1640	2.1216	0.7170
PO34	28.4627	3	-2.1566	2.1294	1.0313

Table 2. Performance of neural network for height prediction

$$\mathbf{V}_p = (r_e + h) * \begin{bmatrix} \cos\phi \cdot \cos\lambda \\ \cos\phi \cdot \sin\lambda \\ \sin\phi \end{bmatrix} \qquad (2)$$

where

$\mathbf{V_p}$ is the position vector, r_e the radius of the Earth, (6378km) and ϕ, λ, h are the latitude, longitude, altitude respectively.

In order to simulate the output of an onboard GPS receiver, the latitude, longitude and altitude values generated by Winorbit software are used as measurements by adding a normal measurement noise (variance: 0.01 degree^2 for latitude and longitude, 1e-6 km^2 for altitude) from 1st to 1400th sample representing the normal behavior of the receiver. An increased noise (variance: 1 degree^2 for latitude and longitude, 1e-4 km^2 for height) is added from 500th to 700th sample to represent the noise behavior and finally a total failure is represented by "no signal" from 1000th to 1200th sample. Such a group of simulated measurements is used to validate the failure detection scheme explained in the following section. Figure 4 shows the simulated measurements. A group of back-up orbit values as generated by the neural network is shown in Figure 5.

To compare these two groups of data, the vectors $\mathbf{V_p}$ are calculated by (2). Then the norm of the difference shown in Figure 6 is used to detect failure.

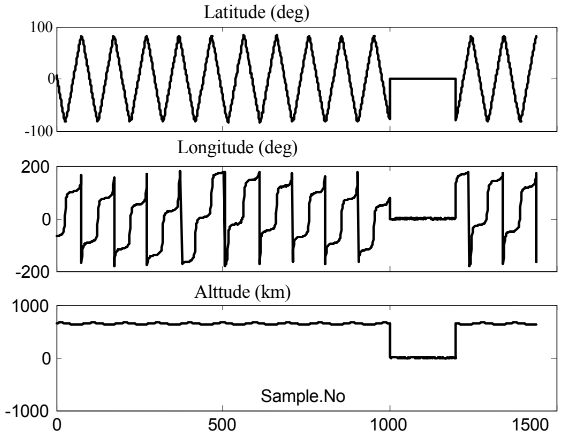

Figure 4. Simulated GPS receiver measurements

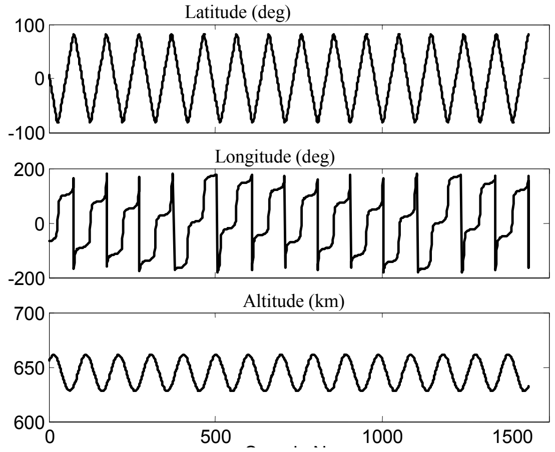

Figure 5. Back-up orbit data

Another parameter which can be used to assist in reducing the false alarms is the difference in the norm of the vectors. This is also shown in Figure 7.

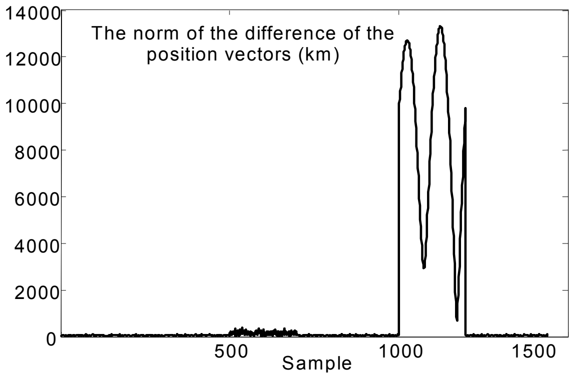

Figure 6. Norm of the difference of the position vectors

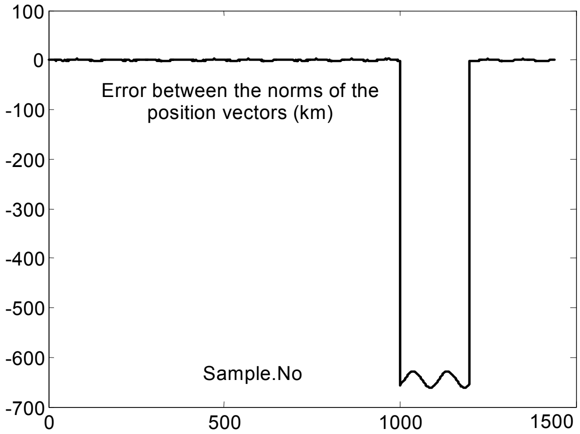

Figure 7. Errors between norms of the position vectors

A failure detection strategy for the GPS receiver by combining with the output of an ANN based orbit predictor is shown in Figure.8.

In the simulation cases, the variance of the error calculated in a small window (say a window size of 4 samples) and the error are used together to identify the two modes of failure ('noisy state' and 'total failure' state). Figure 9 shows the identification of failure modes. It can be seen that all the failure modes are detected and identified correctly (no false alarm) almost rapidly (in about 3 samples).

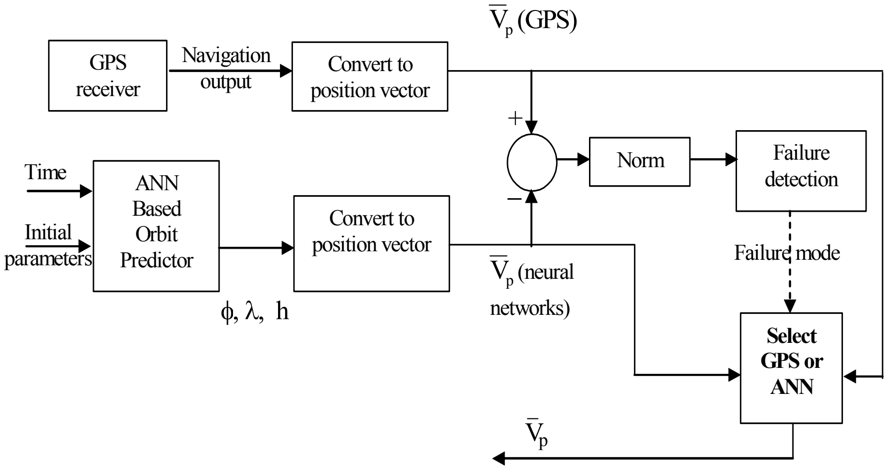

Figure 8. Failure detection scheme using ANN

Based on the final output of the selection switch, Figure 10 shows that, the failure detection strategy effectively isolates the output of the faulty receiver.

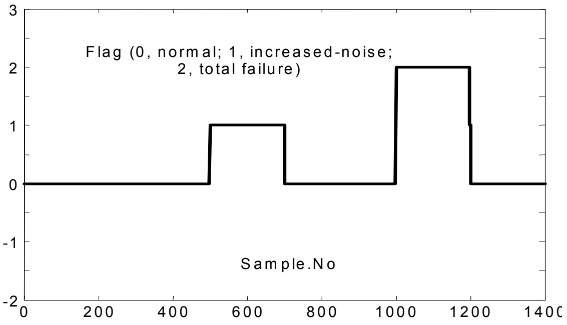

Figure 9 Detection of failure modes

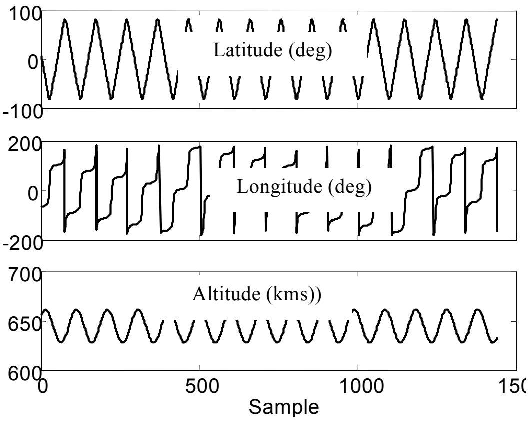

Figure.10 Final outputs of the failure detection system

CONCLUSION

Artificial Neural Network (ANN) is shown to be quite simple effective for orbit prediction, as compared to the nonlinear model involving trigonometric functions. The scheme also has the advantages mentioned in that all the failures can be detected quickly, and the false alarm rate is also lower (in this scheme, no false alarm occurs).[3] It can fulfill the tasks of detecting the failures and recover the measurements by using only a virtual sensor.[4]

Simulations show that such a network can function as a back-up system (with the standard deviations of the errors less than 0.08 degrees in latitude and longitude for all orbital inclinations and about 1km for GPS based navigation. The size of the neural network is also small (no more than 10 neurons in the hidden layer). The predictor is combined with the failure detection strategy to establish an orbit failure detection scheme. It is proved by simulation to be successful in failure detection and modes identification. The time lags are shorter than 3 samples and no false alarm occurs. The system can give out the orbit prediction with good performance.

References

[1] G. C. Page, J. B. Gomm, and D. Williams: *Application of neural networks to modeling and control*, Champman & Hall, London, UK, 1993.

[2] M. R. Napolitano, D. A. Windon, J. L. Casanova, M. Innocenti, and G. Silverstri, "Kalman filters and neural-network schemes for sensor validation in flight control systems," *IEEE Transactions on control systems technology*, vol. 6, No. 5, September 1998.

[3] T. W. Long, E. L. Hanzevack, and W. L. Bynum, "Sensor fusion and failure detection using virtual sensors," *IEEE American Control Conference, Proceedings of the 1999*, Vol. 4, pp. 2417-2421, 1999.

[4] T. H. Guo and J. Nurre, "Sensor failure detection and recovery by neural networks," *Neural Networks, 1991., IJCNN-91-Seattle International Joint Conference on* , Vol: 1 , pp. 221 –226, 1991.

[5] K. Hornik, M. Stinchcombe, and H. White, "Multiplayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359-366, 1989.

[6] Two Line Element's website, Address: http://celestrack.com/.

[7] G. C. Page, J. B. Gomm, and D. Williams, *Application of neural networks to modeling and control*, Champman & Hall, London, UK, 1993.