Learning Characteristics of a Space-Time Neural Network As a Tether "Skiprope Observer"

N93-22176

Robert N. Lea and James A. Villarreal
NASA / Lyndon B. Johnson Space Center
Houston, Texas 77058

Yashvant Jani
Togai InfraLogic Inc.
Houston, Texas 77058

Charles Copeland
Loral Space Information Systems
Houston, Texas 77058

Abstract :

The Software Technology Laboratory at the Johnson Space Center is testing a Space Time Neural Network (STNN) for observing tether oscillations present during retrieval of a tethered satellite. Proper identification of tether oscillations, known as "skiprope" motion, is vital to safe retrieval of the tethered satellite. Our studies indicate that STNN has certain learning characteristics that must be understood properly to utilize this type of neural network for the tethered satellite problem. We present our findings on the learning characteristics including a learning rate versus momentum performance table.

1.0 Introduction

NASA and the Italian Space Agency plan to fly the Tethered Satellite System (TSS) aboard the Space Shuttle in July, 1992. The mission, lasting approximately 40 hours, will deploy a 500 kg satellite upward (away from the earth) [1, 2] to a length of 20 km, perform scientific experiments while on-station, and retrieve the satellite safely. Throughout the deployment, experimentation, and retrieval, the satellite will remain attached to the Orbiter by a thin tether through which current passes, providing power to experiments on-board the satellite. In addition to the scientific experiments on-board the satellite, the dynamics of the TSS itself will be studied. The TSS dynamics are complex and non-linear due to the mass as well as spring-like characteristics of the tether. When the tether is modeled as a massless spring, it typically exhibits longitudinal and librational oscillations [2]. However, when the tether is modeled as beads connected via springs as shown in fig. 1, the dynamics of TSS includes longitudinal, librational and transverse circular oscillations or so-called "skiprope" phenomenon. These circular oscillations are generally induced when current pulsing through the tether interacts with the Earth's magnetic field [3, 4]. The center bead typically displaces the most from the center line. Thus, the "skiprope" can be viewed (fig. 2) by plotting a trajectory of the mid-point of the tether as it is retrieved slowly from the Onstation-2 phase in a high fidelity simulation test case. Detection and control of the various tether modes, including the 'skiprope' effect, is essential for a successful mission. Since there are no sensors that can directly provide a measure of skiprope oscillations, indirect methods like Time Domain Skiprope Observer [4] and Frequency Domain Skiprope Observer [3] are being developed for the mission. We are investigating a Space Time Neural Network (STNN) based skiprope observer.

The STNN is basically an extension to a standard backpropagation network [5,6,7] in which the single interconnection weight between two processing elements is replaced with a number of Finite Impulse Response (FIR) filters [8]. The use of adaptable, adjustable filters as interconnection weights provides a distributed temporal memory that facilitates the recognition of temporal sequences inherent in a complex dynamic system such as the TSS. We have performed experiments in detecting various parameters of skiprope motion using an STNN.

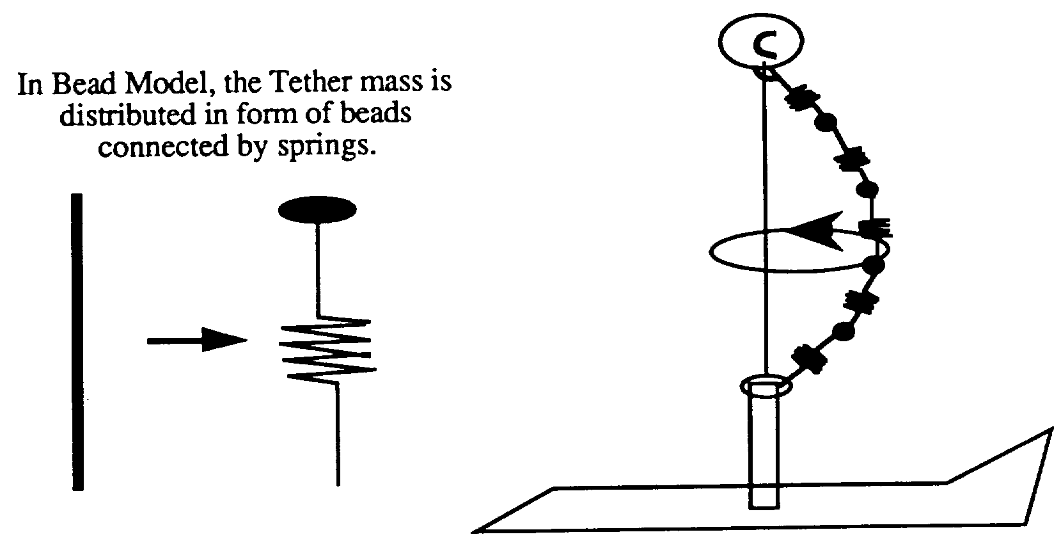


Fig. 1 When tether is modelled as beads, the transverse circular oscillations known as "skiprope" are induced during retrieval.

Extensive studies using high fidelity simulations have shown that the tethered satellite exhibits characteristic rate oscillations in the presence of skiprope motion as shown in figure 3. Since these rate oscillations are measured by the satellite's on-board rate gyros, the measured rates can be used as inputs to a skiprope detection system along with other measured parameters such as tension and length [9]. We have trained an STNN using data logged from a high fidelity Orbital Operations Simulator (OOS) [10] which models the behavior of the TSS. The parameters used in network training include satellite roll, pitch, and yaw rates, sensed tension and length of the tether, and the position of the mid-point of the tether during skiprope motion. In this paper, we first describe the STNN architecture in section 2. The STNN configuration used for skiprope observation is described in section 3 along with training and test data generated by the simulation test cases. Learning characteristics are discussed in section 4, and conclusions are summarized in section 5.

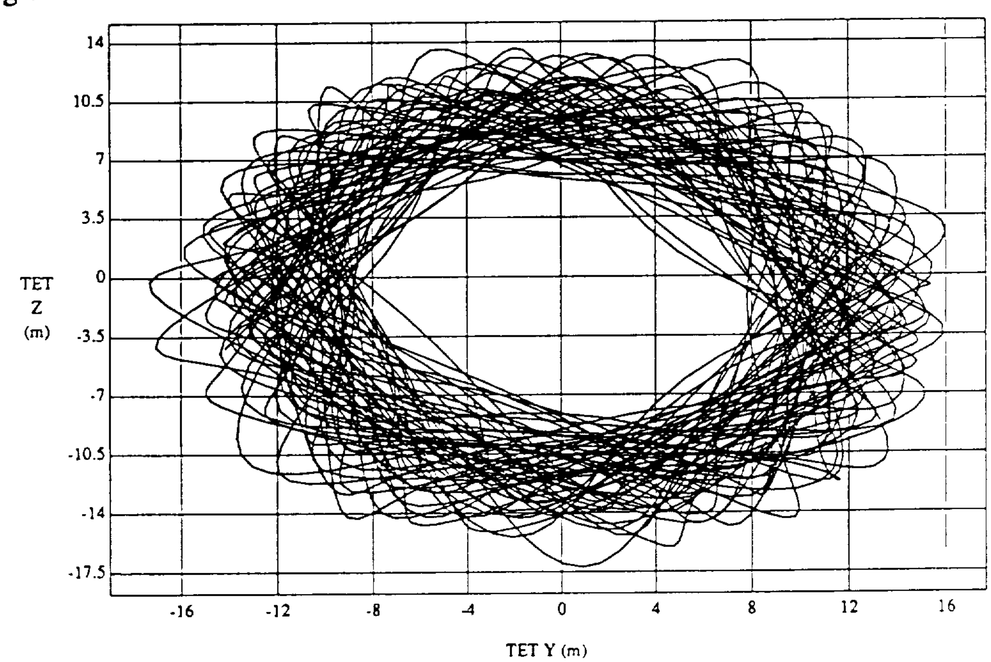


Figure 2 - Trajectory of tether mid-point during "skiprope".

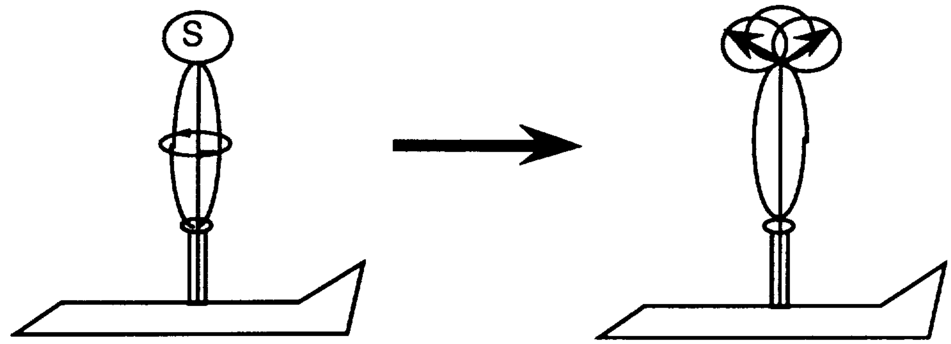


Figure 3 - Tether "skip rope" effect leads to highly characteristic satellite attitude oscillations which can be used to detect the magnitude and phase of the skiprope

2.0 STNN Architecture

The STNN architecture [8] allows the dimension of time to be added to the strong spatial modelling capabilities found in neural networks. The time dimension can be added to the standard processing element used in conventional neural networks by replacing the synaptic weights between two processing elements with an adaptable-adjustable filter as shown in figure 4.

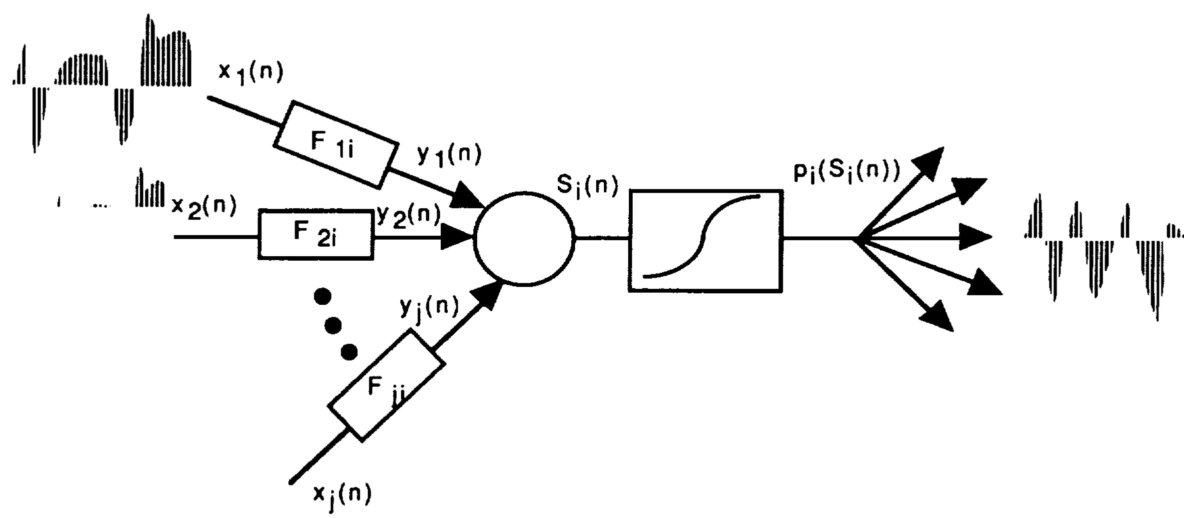


Figure 4 - A pictorial representation of the Space-Time processing element.

Instead of a single synaptic weight with which the standard backpropagation neural network represented the association between two individual processing elements, there are now several weights representing not only spatial association, but also temporal dependencies. In this case, the synaptic weights are the coefficients to the adaptable digital filters:

$$y(n) = \sum_{k=0}^{N} b_k x(n-k) + \sum_{m=1}^{M} a_m y(n-m) \qquad (1)$$

Here the x and y time sampled sequences are the input and output respectively of the filter and a_m's and b_k's are the coefficients of the filter. Thus, if there are j parameters going into a neuron, the y_j

are input into the neuron, where each y_j is a filtered value of the x_j using n time series samples as shown in fig. 4. The x_j's are the real input from an external source. Thus, the STNN is learning a temporal dependency of the input parameters.

A space-time neural network includes at least two layers of filter elements fully interconnected and buffered by sigmoid transfer nodes at the intermediate and output layers as shown in figure 5. A sigmoid transfer function is not used at the input. Forward propagation involves presenting a separate sequence dependent vector to each input, propagating those signals throughout the intermediate layers until the signal reaches the output processing elements. In adjusting the weighting structure to minimize the error for static networks, such as the standard backpropagation, the solution is straightforward. However, adjusting the weighting structure in a space-time network is more complex because not only must present contributions be accounted for but contributions from past history must also be considered. Therefore, the problem is that of specifying the appropriate error signal at each time and thereby the appropriate weight adjustment of each coefficient governing past histories to influence the present set of responses. A detailed discussion of the algorithm can be found in the reference [8].

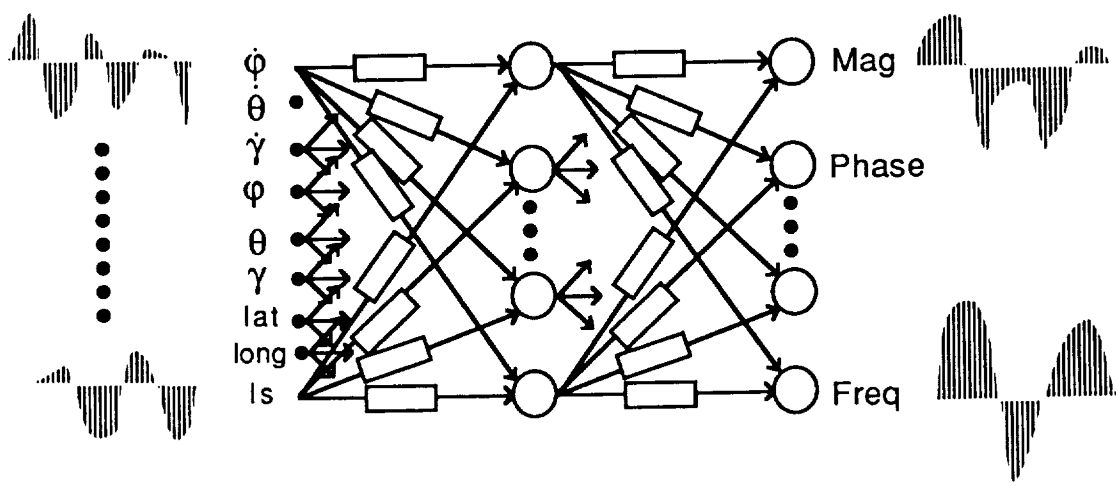


Figure 5 - A depiction of a STNN architecture showing the distribution of complex signals in the input space.

3.0 STNN Configuration and Test/Training Data

Several different simulation runs were used to gather data for STNN training. The simulation runs are consistent with the requirement that the skiprope observer must be capable of performing during various combinations of current flow through the tether and satellite spin. For example, one simulation represents a case in which current flows through the tether continuously, and the satellite is in yaw hold. Another simulation represents the case in which current flows through the tether only during the on-station phase, and the satellite is in yaw hold. A third simulation represents continuous current flow, and satellite spin at 4.2 degrees/second. These three scenarios will form the basis for STNN skiprope observer training and testing, and are consistent with simulations that are used for testing the Time-Domain Skiprope Observer (TDSO) [4] which will be flown on TSS-1.

Ultimately, the network should utilize only roll rate, pitch rate, yaw rate, sensed tension and sensed length since these are the only directly measurable parameters. However, we have conducted experiments using derived parameters such as roll, pitch, and yaw position in addition to rates with no significant improvement. The biggest challenge to network training so far has been to learn the phase mapping. Several different network configurations have yielded good results in predicting skiprope amplitude, but we have not been as lucky with skiprope phase. Since the ultimate goal is to provide the crew with accurate measurements of skiprope amplitude and phase to support the yaw maneuver, the skiprope observer should learn to predict amplitude and phase based on the available inputs. However, decisions concerning the yaw maneuver can be based on the x and y coordinates of the mid-point of the skiprope motion as well. Therefore, the basic network configuration consists of 6 inputs (roll rate, pitch rate, yaw rate, sensed tension, x(t), and y(t)) and 2 outputs (x (t+1) and y(t+1)). Notice that we are training on the current x and y position and predicting x and y position for the next time step. In previous experiments we focussed on finding the optimum network configuration in terms of numbers of hidden units and numbers of zeros from layer to layer. Through experimentation, we settled on 30 hidden units and 30 zeros from the input layer to the hidden layer, and 30 zeros from the hidden layer to the output layer, although slight deviations in these parameters have little or no effect in network performance. In this paper we concentrate primarily on the effects of learning rate and momentum on the overall generalization of the Space-Time Neural Network.

4.0 Learning Characteristics

A well known characteristic of backpropagation networks, or networks derived from backpropagation, is that in order to achieve reasonable generalization, the network must learn the training data. Experiments have indicated that, like standard backpropagation, the learning characteristics of STNN are such that if the training data is not learned, generalization will not occur. These and other learning characteristics dictate that a particular sequence of steps be followed in the training and testing of STNN. The following general steps were used as guidelines throughout the STNN testing.Please note that the use of the word "momentum" in this report refers to a term in the learning algorithm that represents a fraction of the previous weight change rather than any physical properties of the TSS.

1. Train and test - evaluate learnability of training data.
2. Adjust network as necessary (set learning rate and momentum in updating of interconnection weights).
3. If network is unable to obtain sufficient convergence on training data, test individual parameters one at a time. Eliminate un-mappable parameters and start over.
4. If reasonable convergence is realized on training data, divide the data set into a training set and a separate test set.
5. When reasonable performance is achieved on the separate test data, then go for multi-test case generalization.

Step 2 above generally involves trying different combinations of learning rate and momentum in the interconnection weight update formulas. Table 1 illustrates the test case matrix we have identified in order to test the effects of different combinations of learning rate and momentum.

The results that follow are from training and testing using data from the simulation which includes current pulsing and satellite spin, which is considered the most difficult case. Following our general training and testing steps listed above, we verified that the STNN was able to learn the training data using a learning rate of 0.05, and momentum set to 0.9. We trained and tested on all 3500 Input/Output pairs and achieved a MAX error of 0.08 and RMS error of 0.02 at 140 cycles. Since the network will be trained off-line before being placed in the operational environment, we must determine how well the network will perform when presented with data that it has not previously seen. Therefore, to test the generalization ability of STNN, we train on only the first and last 400 input/output pairs from the full 3500, and test separately on the middle 2700 input/output pairs while trying various combinations of learning rate and momentum with the following results. First, with a momentum of 0.9, we tried learning rates of 0.05, 0.2, and 0.7 (test cases #4-6 in Table 1). Test case #4 resulted in MAX error = 0.43, and RMS error = 0.04 at cycle 100. Figure 6 shows the error plot for test case #4 up to 500 cycles. Figures 7a and 7b show a portion of the x and y predictions from test case #4. Test case #5 resulted in MAX error = 0.43 and RMS error = 0.04 at cycle 480. Figure 8 shows that the network prediction of y in test case 5 is similar to that of test case #4. Increasing the learning rate to 0.7 in test case #6 results in the network never reaching errors as low as in the previous two test cases (at least not within 500 cycles) and overall performance is similarly degraded as is seen in figures 9a and 9b. Next we set momentum to 0.2 and try learning rates of 0.05, 0.2, and 0.7 (test cases #1-3 in Table 1). Test case #1 yielded MAX error = 0.44, and RMS error = 0.05 at 100 cycles, as is shown in figure 10a. Figure 10b shows that the network's prediction of x in this test case is not quite as accurate as test cases #4 and #5. As we increase learning rate from 0.05 to 0.2, performance degrades significantly as is shown in figure 11a. The error graph in figure 11b shows that no learning occurred in test case #2, as RMS error never dropped significantly below 0.5, and MAX error remained near 0.8. Similar results occurred in test case #3 as we increased the learning rate from 0.2 to 0.7. The overall test errors are summarized in Table II.

Table 1 - Learning Rate Versus Momentum in STNN Weight Update Formulas

Test Case	Momentum in weight update	Learning Rate
1	0.2	0.05
2	0.2	0.2
3	0.2	0.7
4	0.9	0.05
5	0.9	0.2
6	0.9	0.7
7	0.95	0.05
8	0.98	0.05

Table II - Number of Training Cycles to Reach Lowest Test Errors.

Test Case	Max Error	RMS Error	Number of Cycles
1	0.44	0.05	100
2	0.78	0.49	280
3	0.8	0.5	480
4	0.43	0.04	100
5	0.43	0.04	480
6	0.5	0.09	400
7	0.41	0.04	480
8	0.41	0.04	480

5.0 Conclusions

Through experimentation, we have gained insight into the learning characteristics of STNN in terms of learning rate and momentum parameters. In particular, we find that the skiprope observer problem requires high momentum and very low learning rate. In test case 4 we have seen that the RMS error drops to 4 % within only 100 cycles of learning. We further verified this by performing two test cases (#7 and #8) with high momentum and low learning rate. It should be noted that the max error is reduced in both cases.

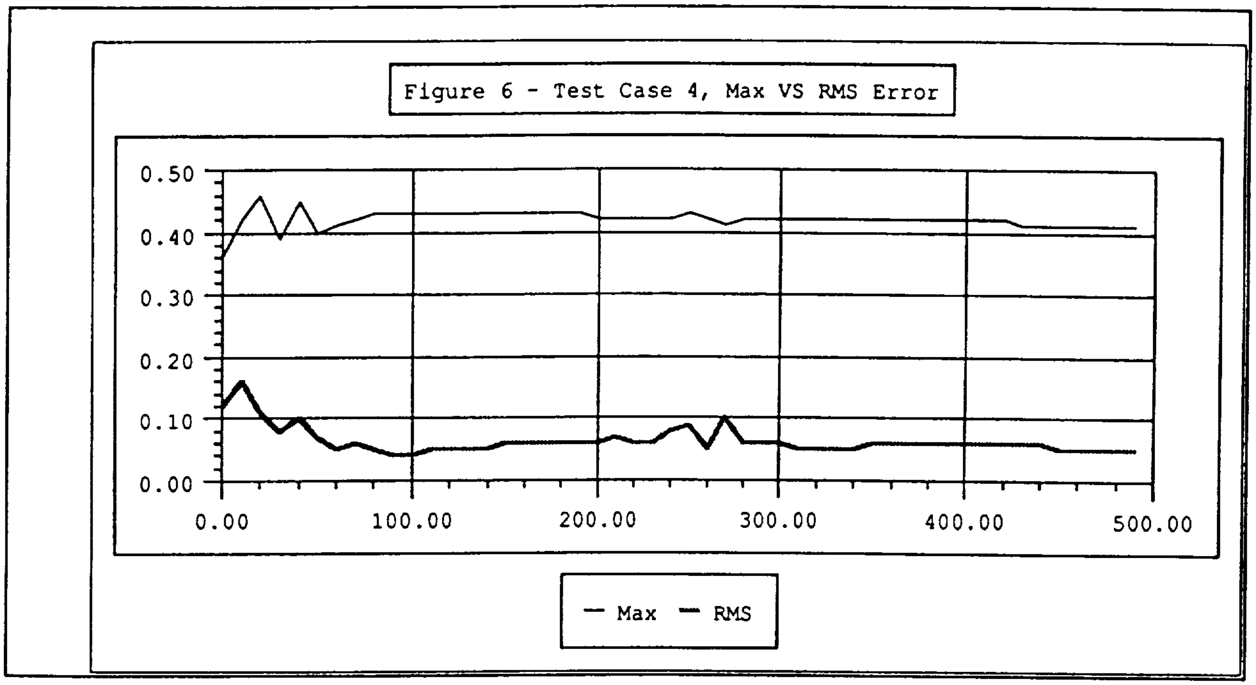


Figure 6 - Test Case 4, Max VS RMS Error
0.50
0.40
0.30
0.20
0.10
0.00
0.00
100.00
200.00
300.00
400.00
500.00
Max
RMS

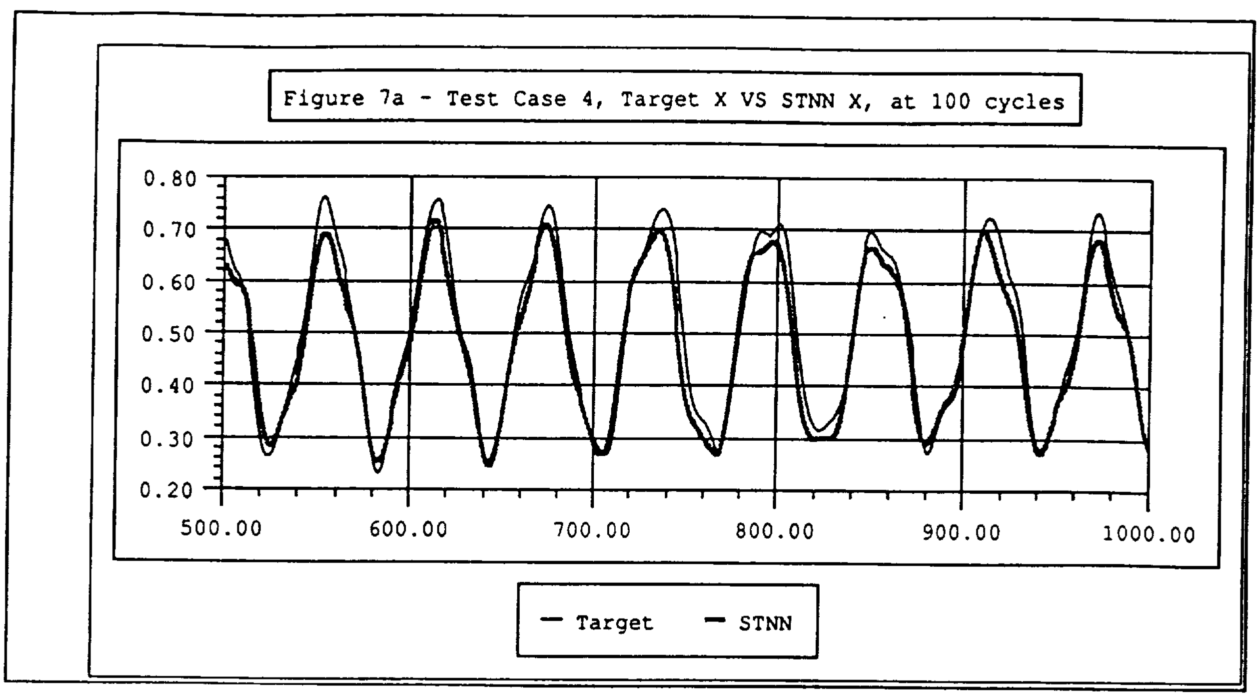


Figure 7a - Test Case 4, Target X VS STNN X, at 100 cycles
0.80
0.70
0.60
0.50
0.40
0.30
0.20
500.00
600.00
700.00
800.00
900.00
1000.00
Target
STNN

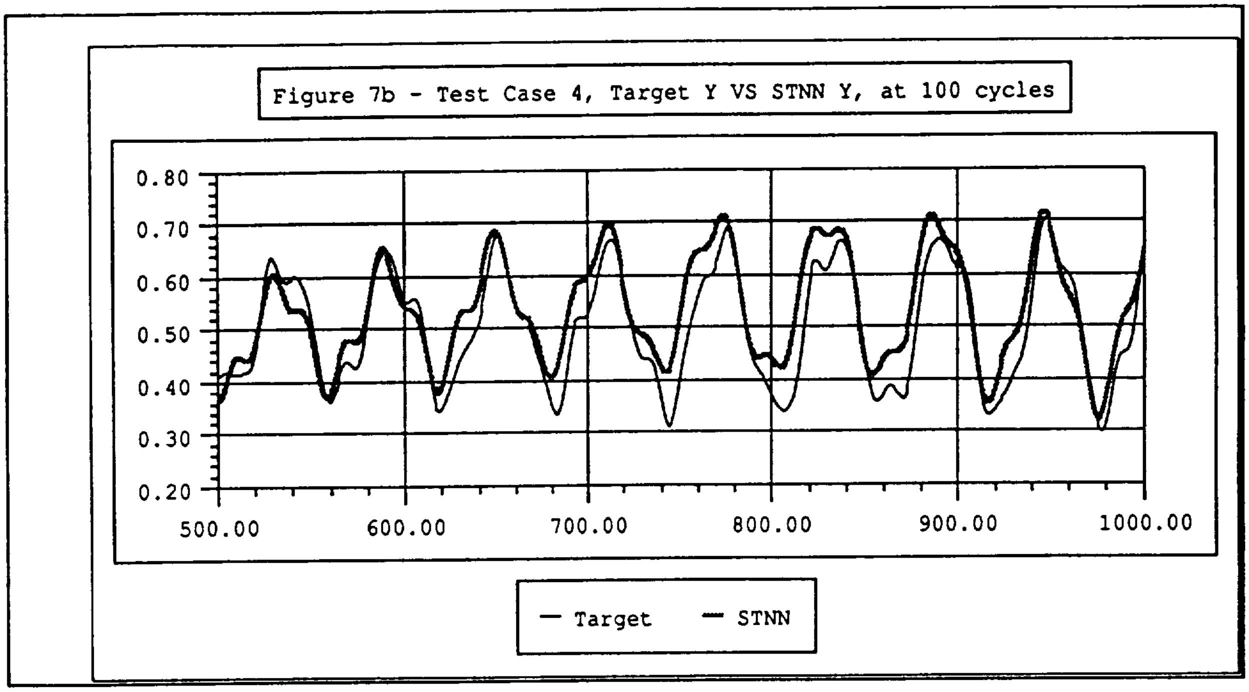


Figure 7b - Test Case 4, Target Y VS STNN Y, at 100 cycles
0.80
0.70
0.60
0.50
0.40
0.30
0.20
500.00
600.00
700.00
800.00
900.00
1000.00
Target
STNN

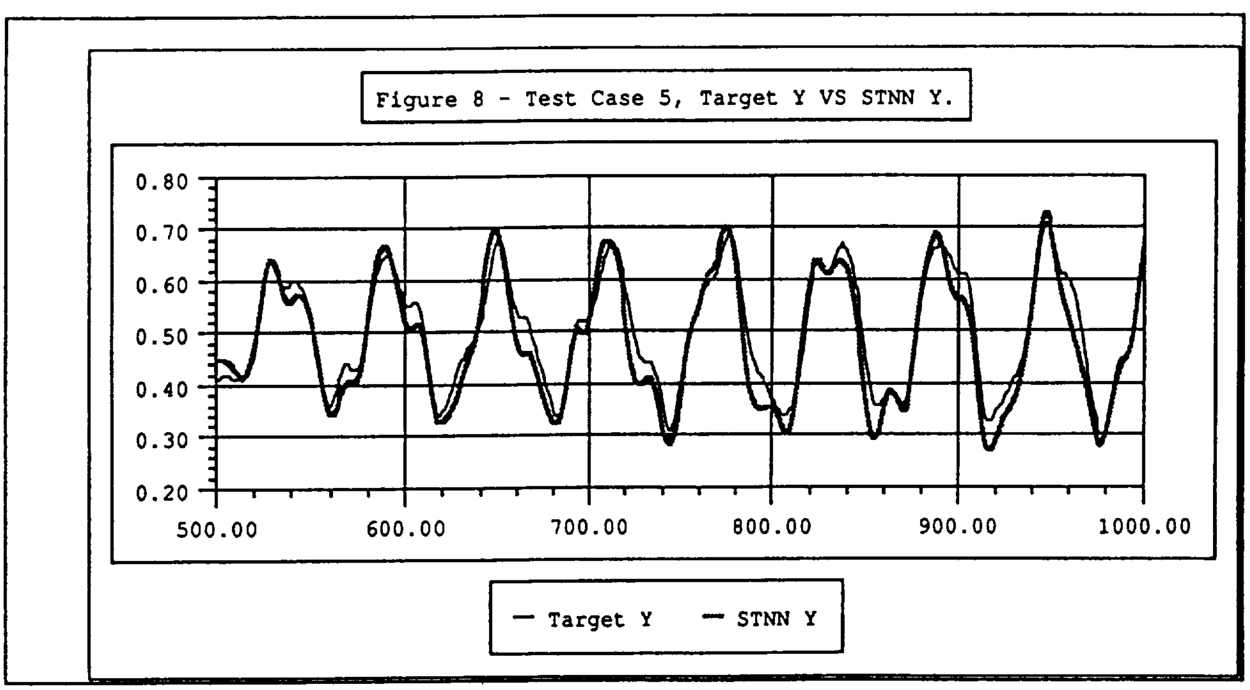


Figure 8 - Test Case 5, Target Y VS STNN Y.
0.80
0.70
0.60
0.50
0.40
0.30
0.20
500.00
600.00
700.00
800.00
900.00
1000.00
Target Y
STNN Y

Figure 9a - Test Case 6, Max VS RMS Error

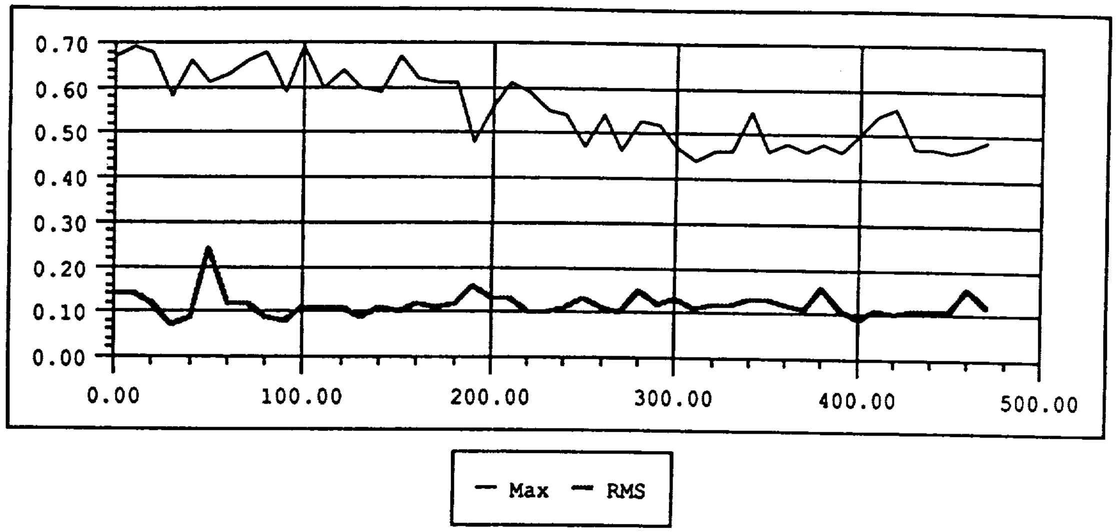
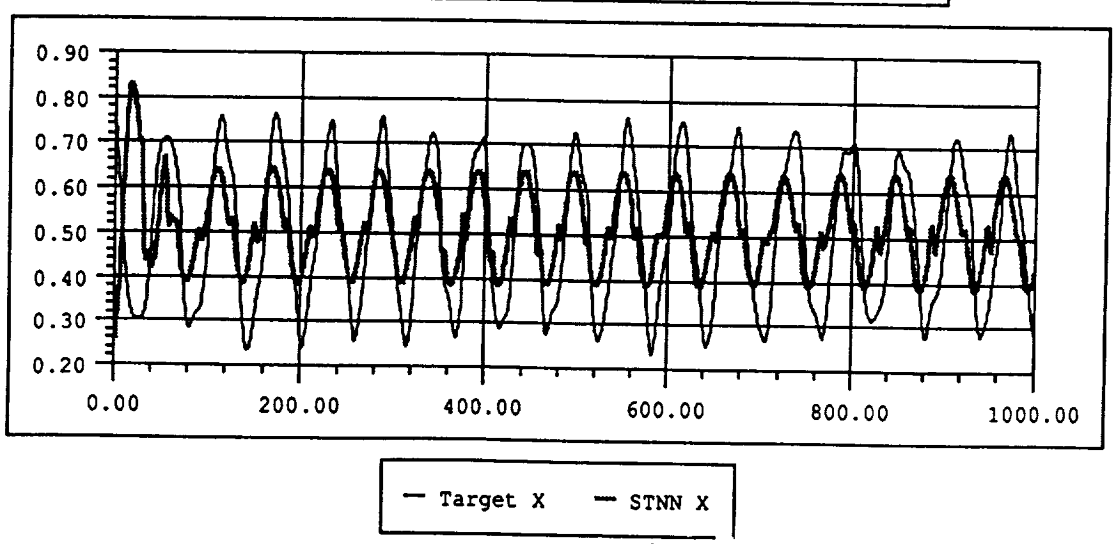


Figure 9b - Test Case 6, Target X VS STNN X, 400 cycles



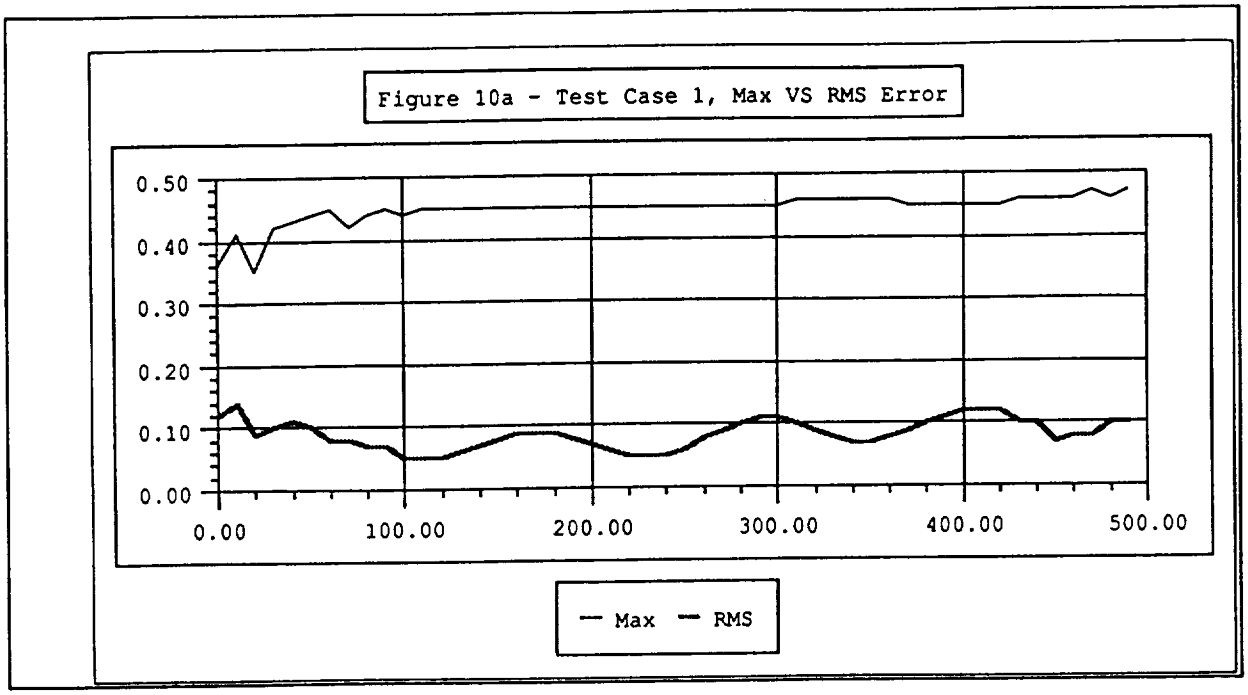


Figure 10a - Test Case 1, Max VS RMS Error
0.50
0.40
0.30
0.20
0.10
0.00
0.00
100.00
200.00
300.00
400.00
500.00
Max
RMS

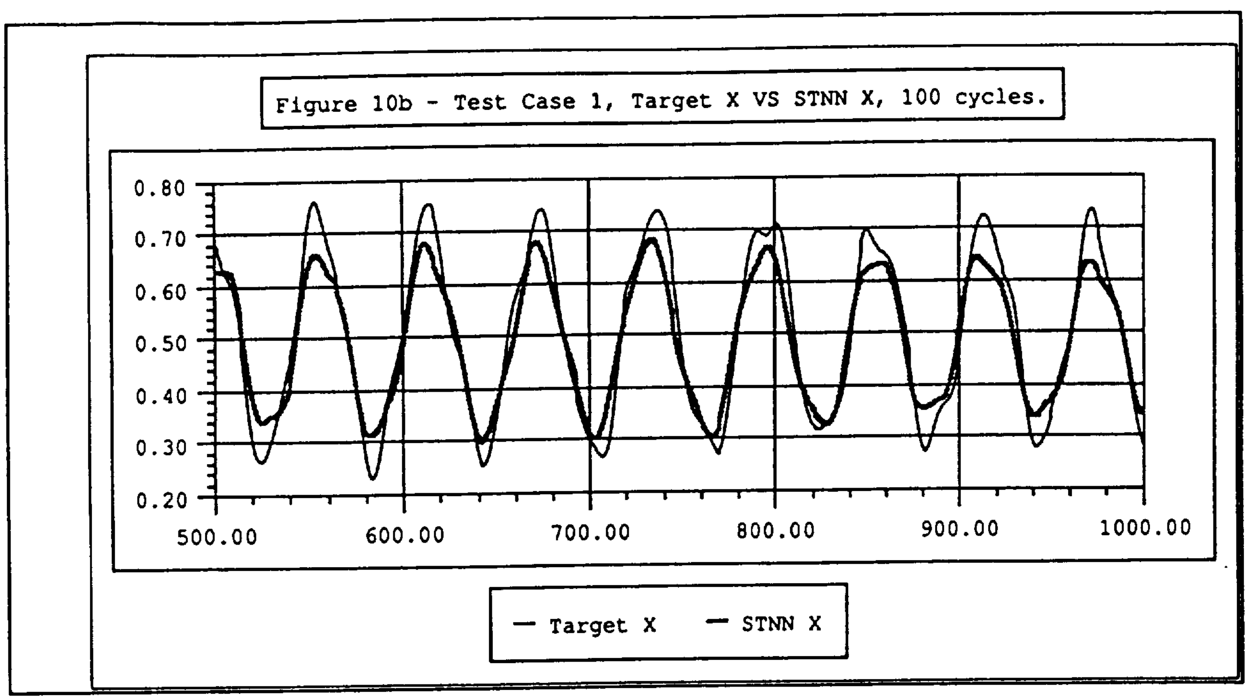


Figure 10b - Test Case 1, Target X VS STNN X, 100 cycles.
0.80
0.70
0.60
0.50
0.40
0.30
0.20
500.00
600.00
700.00
800.00
900.00
1000.00
Target X
STNN X

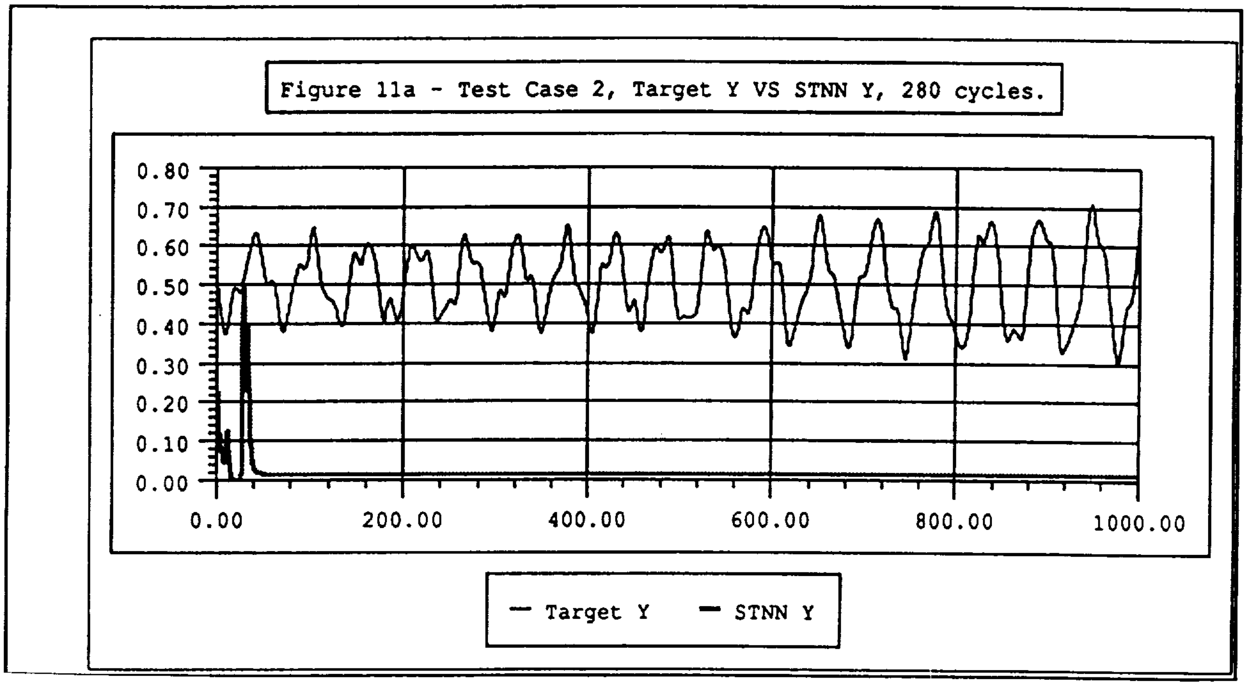


Figure 11a - Test Case 2, Target Y VS STNN Y, 280 cycles.
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
0.00
200.00
400.00
600.00
800.00
1000.00
Target Y
STNN Y

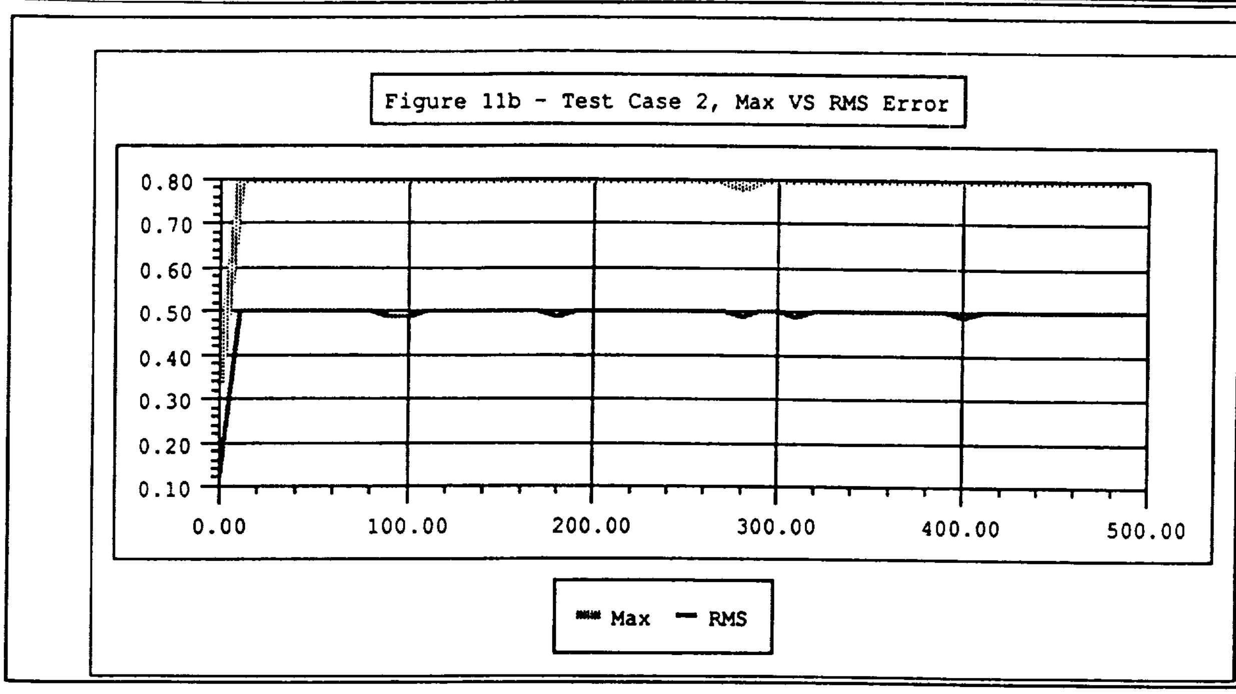


Figure 11b - Test Case 2, Max VS RMS Error
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
100.00
200.00
300.00
400.00
500.00
Max
RMS

C-4

Based on our earlier results, we conclude that the STNN is slow in learning sharp discontinuities like those encountered in phase behavior. The value of the phase goes from 180 to -180 abruptly when the circle is complete. When we changed to the x- and y- component form (rather than amplitude and phase), the STNN based skiprope observer performed much better in predicting x and y coordinates of the mid-point of the tether.

We will have an opportunity to perform a side-by-side comparison of the STNN based skiprope observer and the TDSO using simulation data. Next, we will test the STNN based skiprope observer with the post mission data after the TSS-1 flight.

References :

1. Coledan, S. : "Tether Satellite Advances", Space News, vol. 2, no. 15, p. 8, 1991.
2. Powers, C.B., Shea, C., and McMahan, T. : "The First Mission of the Tethered Satellite System", A special brochure developed by the Tethered Satellite System Project Office, NASA/Marshall Space Flight Center, Huntsville, Alabama, U.S. GPO 1992-324-999, 1992.
3. Ioup, G.E., Ioup, J.W., Rodrigue, S.M., Amini, A.M., Rayborn, G.H., and Carroll, S. : "Frequency Domain Skiprope Observer", Skiprope Containment Status Meeting held at Denver, Sep. 10-11, 1991. (Research supported by NASA Contract NA8-38841)
4. Glowczwski, R. : " Time Domain Skiprope Observer Overview", Skiprope Containment Status Meeting held at Martin Marietta, Denver, Sept. 10-11, 1991.
5. Kosko, B. : "Neural Networks and Fuzzy Systems", Prentice-Hall, New Jersey, 1992.
6. J.A.Freeman and D.M.Skapura : "Neural Networks Algorithms, Applications and Programming Techniques", Addison-Wesley Publishing Company, Reading, MA. 1991.
7. Wasserman, P.D. : "Neural Computing Theory and Practice" Van Nostrand Reinhold, New York, 1989.
8. Villarreal, J.A., and Shelton, R.O. : "A Space-Time Neural Network", International Journal of Approximate Reasoning , 6(2), 133-149, 1992.
9. Lea, R.N., Villarreal, J.A., Jani, Y., and Copeland, C. : "Application of Space Time Neural Networks to detect Tether Skiprope Phenomenon in Space Operations", Proceedings of the AIAA GN&C Conference held at Hilton Head, South Carolina, August 10-12, 1992.
10. Edwards, H. C., and Bailey, R. : "The Orbital Operations Simulator User's Guide", LinCom corporation, ref. LM85-1001-01, June 87.