Technical Notes

TECHNICAL NOTES are short manuscripts describing new developments or important results of a preliminary nature. These Notes should not exceed 2500 words (where a figure or table counts as 200 words). Following informal review by the Editors, they may be published within a few months of the date of receipt. Style requirements are the same as for regular contributions (see inside back cover).

Testing of Inflatable-Structure Shape Control Using Genetic Algorithms and Neural Networks

Fujun Peng*
Tongji University,
200092 Shanghai, People's Republic of China
and
Yan-Ru Hu† and Alfred Ng‡
Canadian Space Agency,
St.-Hubert, Quebec J3Y 8Y9, Canada

DOI: 10.2514/1.20795

Nomenclature

m	=	number of neurons in the output layer
n	=	number of neurons in the input layer
p	=	number of neurons in the hidden layer
v_{ij}, w_{jk}	=	neural network weights
X_i	=	neural network inputs
Y_k	=	neural network outputs
Z_j	=	activation function outputs
β_j, ρ_k	=	neural network thresholds
γ_j	=	combination function outputs

I. Introduction

INFLATABLE structures have attracted much interest in the space community due to their unique advantages in achieving low mass and high packaging efficiency [1,2]. We are currently working on an in-house research and development project of a membrane synthetic aperture radar antenna and solar array (see Fig. 1). It is expected that the membrane will be subjected to flatness problems during its lifetime in orbit, due to the thermal variation in space. A purely passive control method may not be sufficient to keep the membrane flat. Hence, an active control system is proposed to adjust the tensions according to the thermal variation. Actuators are installed in series with the links, such that the tensions stretching the membrane can be adjusted. A genetic algorithm and neural network (GA–NN) scheme is proposed for modeling and tension optimization. The neural network model of the membrane is established as a mapping from the boundary stretching tensions and space environment to membrane flatness. After the neural network training is completed, the membrane flatness can be estimated by inputting the measured stretching tensions and space environment data to the neural network model. Based on the neural network model, the genetic algorithm is applied to search for the optimal tensions that minimize the membrane wrinkles [3]. Experimental results demonstrate the effectiveness of the proposed scheme.

Presented as Paper 2055 at the 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference, Austin, TX, 18–21 April 2005; received 28 October 2005; revision received 18 February 2007; accepted for publication 18 February 2007.

*Professor, College of Aerospace Engineering and Mechanics, 1239 Siping Road. Member AIAA.

†Research Scientist, Control & Analysis Group, Spacecraft Engineering Directorate, 6767 Route de L'Aeroport.

‡Manager, Control & Analysis Group, Spacecraft Engineering Directorate, 6767 Route de L'Aeroport.

II. Genetic Algorithm and Neural Network

To implement the search for the optimal tensions for the membrane, all of the parameters (here, they are the amplitudes of tensions) to be optimized are first mapped (coded) into a chromosome. Each parameter corresponds to one particular portion of the chromosome. Then the following steps are executed to search for the best solutions (see Fig. 2):

1) According to the structure of this chromosome, an initial population is generated, with a group of individuals created randomly. Each individual corresponds to a possible solution of the given task.

2) The individuals are evaluated and each of them is given a score based on how well they perform at the given task.

3) If any individual is found meeting the given requirement, the optimal solutions can then be obtained directly by decoding the optimal individuals. Otherwise, the genetic algorithm will go to the next cycle to generate a new population.

4) The individuals are selected and paired based on their fitness: the higher the fitness, the higher the chance of being selected.

5) Using a crossover probability, cross over the individuals to form new offspring.

6) Using a mutation probability, mutate new offspring at each locus (position in an individual).

7) Place new offspring in a new population. The new individuals generally perform better at the given task than the parent individuals.

8) Loop; go to step 2.

To estimate membrane flatness, a mapping needs to be established from the space environment and boundary tensions to membrane flatness. Such a mapping relationship can be treated as a static problem, therefore, a static feedforward neural network is a good choice. Here, we use one hidden-layer multilayer perceptron (MLP), which was proven to be able to approximate any continuous nonlinear function to an arbitrary degree of accuracy if sufficient hidden-layer neurons are used [4,5]. The structure of the MLP network is depicted in Fig. 3.

The basic operation of a hidden-layer neuron involves performing two functions: a combination function and an activation function (logistic sigmoid is used here), which is illustrated in Fig. 4.

$$\gamma_j = \sum_{i=1}^{n} v_{ij} X_i \tag{1}$$

$$Z_j = 1/[1 + \exp(-\gamma_j - \beta_j)], \qquad j = 1, \cdots, p \tag{2}$$

Similarly, the output of the kth neuron in the output layer can be written as

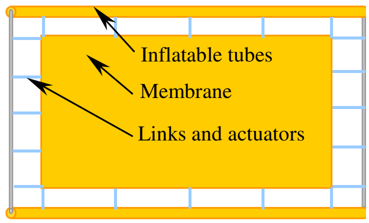

Fig. 1 Concept of the inflatable structure.

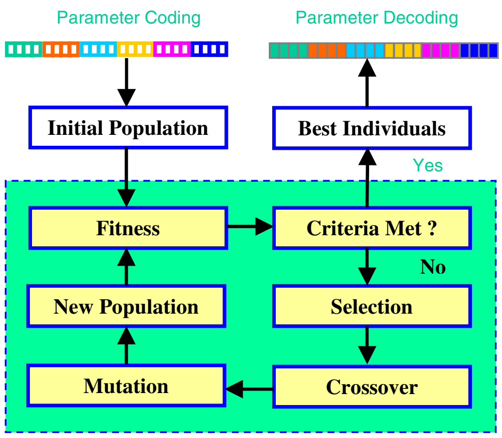

Fig. 2 Block diagram of the genetic algorithm.

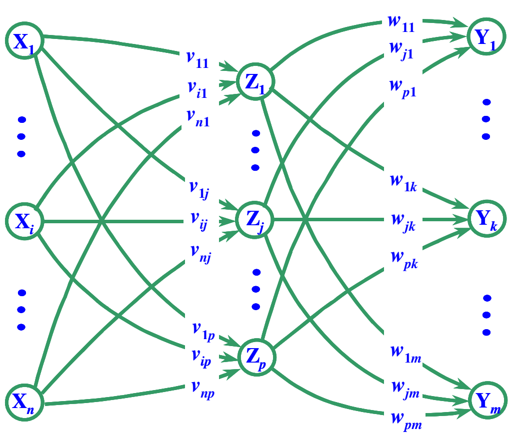

Fig. 3 Depiction of an MLP neural network.

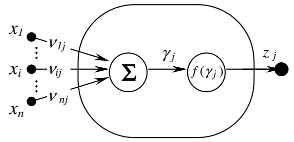

Fig. 4 Basic operation of an artificial neuron.

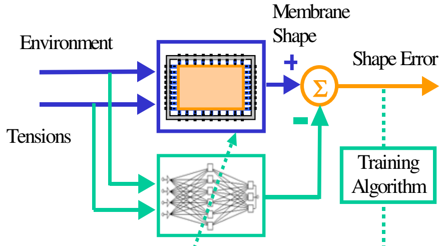

Fig. 5 Block diagram of neural network supervised training.

$$Y_k = f_{\text{output}}\left(\sum_{j=1}^{p} w_{jk} Z_j + \rho_k\right), \qquad k = 1, \cdots, m \tag{3}$$

The weights v_{ij} and w_{jk} and thresholds β_j and ρ_k are unknown parameters and should be determined by a supervised training. The block diagram of the supervised training is shown in Fig. 5. The same environment and tension data are input to the real inflatable structure and the neural network, and a training algorithm adjusts neural network parameters to minimize the difference of their outputs. After the training is completed, the MLP network can be used to estimate the membrane flatness. An estimation can be obtained by inputting the measured environment data and boundary tensions into Eqs. (1–3).

III. Experimental Setup and Control System Implementation

The structure to be used for the neural network validation is shown in Fig. 6. The membrane is a 200×300 mm rectangular Kapton membrane, which is stressed by eight discrete links installed between the membrane boundaries and the aluminum frame. A local thermal-load source is placed under the membrane (not visible in Fig. 6). The membrane flatness is dependent on the local thermal load and the tension combinations. To adjust tension combinations, four shape-memory alloy wire actuators (0.25 mm in diameter and 100 mm in length) are installed on links 1 to 4. To acquire the values of tensions, strain gages are glued onto small and thin aluminum strips, which are then installed on links 1 to 4. To avoid symmetric properties, two springs are intentionally installed on links 5 and 8, whereas links 6 and 7 are rigid.

The shape-memory alloy (SMA) wire actuator cannot be used directly for exerting tensions, due to its poor stability and controllability. It is hard to identify a reliable relationship between strain-output and electrical-input current, and a steady strain output is not achievable even by applying a fixed current. So a feedback SMA controller was developed to ensure the required tension could be achieved with good accuracy [6].

A vision system is developed to measure the membrane flatness. Test results show that the vision system works very well. It takes only 0.1 s to obtain 330 points of 3-D coordinates, with the accuracy on the order of 0.05 mm [7]. The membrane-flatness data obtained from this vision system will be used for network training. When the control system is performing control, the vision system will not feed back the measured data to the control system. Instead, it just monitors the control effectiveness.

The entire control system consists of three modules, which are organized in five loops, as illustrated in Fig. 7. Loop 1 receives parameters from the graphical user interface (GUI) and then triggers loop 2 and loop 3 with proper parameters. Loop 2 executes the vision system independently with a high rate until it is stopped by a trigger. As mentioned earlier, the vision system is only used here to monitor the membrane flatness, and the obtained result is compared with the estimated values to evaluate the effectiveness of the proposed GA–NN scheme. Loop 3 triggers the executions of the GA–NN module

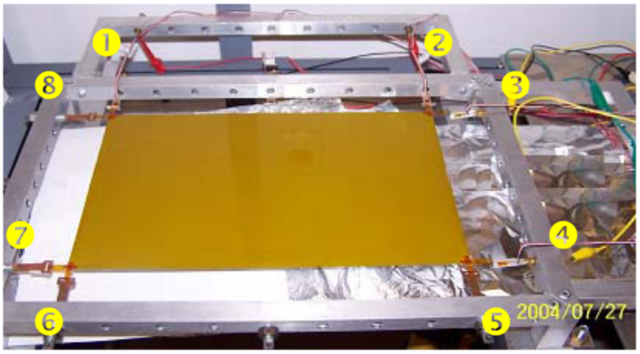

Fig. 6 Membrane structure used for neural network validation.

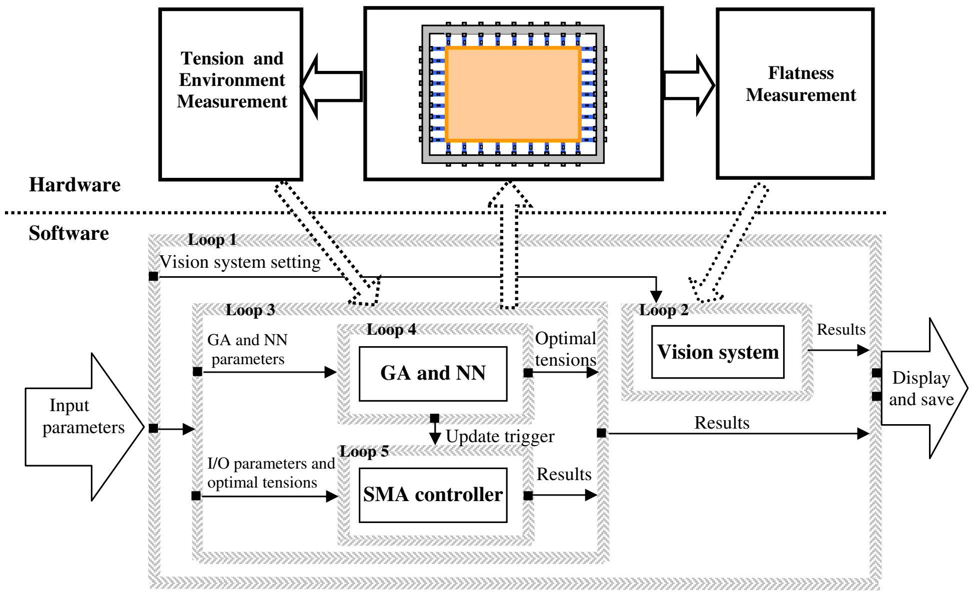

Fig. 7 Block diagram of the control system implementation.

and SMA controller. The GA–NN module can work in two modes: monitoring mode and optimization mode. Most of the time, GA–NN module works in the monitoring mode, that is, it monitors the environment data in real time and estimates the membrane flatness using current environment data and tensions. If the estimated membrane flatness is acceptable, it will remain in this mode. However, if the environment changes too much and the estimated value shows the membrane is not flat enough, the GA–NN module will automatically switch to the optimization mode, that is, it searches for a new optimal-tension combination that can reestablish a flat-membrane state. After the new optimal-tension combination is obtained, the GA–NN module will send it to loop 3 and at the same time trigger the SMA controller to stop, and then GA–NN terminates automatically. After the SMA controller stops, loop 3 will immediately go to the next cycle and transfer the new optimal-tension combination to the SMA controller. The control system is implemented using LabView, Matlab, and Automation Manager. LabView code controls input parameters, results display, and data I/O. Matlab code implements the genetic algorithm and the neural network. Automation Manager realizes the vision system. LabView code is the master, which coordinates the whole system functioning by using the ActiveX technique.

IV. Neural Network Training

For the neural network to better approximate the mapping from environment temperature and tension combinations to the membrane flatness, the input data (the environment temperature and tension combinations) chosen for the neural network training should spread out in a data space in which a number of optimal-tension combinations should exist. After performing preliminary tests, we limit the values of training tensions between 1.32 and 2.64 N, and a total of nine values are uniformly chosen in this region. Here, we only record the tension values of actuators 1 to 4, denoted as T_1 to T_4, and nine values of them are 1.32, 1.49, 1.65, 1.82, 1.99, 2.15, 2.31, 2.48, and 2.64 N. The total number of tension combinations chosen for neural network training is $9 \times 9 \times 9 \times 9 = 6561$. As for the membrane-environment temperature, the heater under the membrane remains unchanged at 200°C. This constant-temperature disturbance produces dynamic wrinkles on the membrane, which is fluctuating all the time. A total of 1000 randomly generated tension combinations are also exerted to the membrane structure, and corresponding values of the membrane flatness are recorded. These data will be used as testing data to validate the trained neural network.

The neural network used here is an MLP network with one hidden layer having logistic sigmoid neurons. It has five inputs, one for

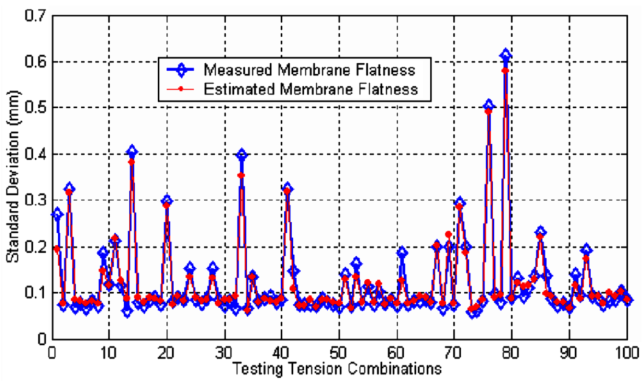

Fig. 8 Measured and estimated membrane flatness corresponding to randomly generated testing data.

environment temperature and the other four for tensions. In the output layer, there is only one neuron with a linear activation function; this neuron corresponds to the membrane flatness. With the hidden-layer neuron number fixed at 10 (the optimal number), the network training is performed 20 times with different training epochs. The obtained minimum testing mean square error is 2.8864×10^{-4} mm^2, and the corresponding training error is 3.18744×10^{-4} mm^2.

Figure 8 shows the estimated and measured values of the membrane flatness produced by the randomly generated testing tension combinations, instead of the training data set. Basically, the estimated values agree quite well with the measured values. When the membrane flatness is poor, the estimated errors are larger. In the cases of good flatness, however, the estimated flatness is very close to measured values. This characteristic is quite similar to the training cases. In practical applications, our concern is only the good flatness cases, in which neural networks offers good approximations. Therefore, the obtained neural network is sufficient and robust for use in the membrane-flatness estimation and control.

V. Test of Membrane-Flatness Control

After the optimal trained neural network is obtained, tests of membrane-shape control are performed. The initial tensions are all set at 1.32 N. For the genetic algorithm, there are eight individuals in one population, and four of them are updated at one time. Other genetic algorithm settings are gray coding, stochastic universal

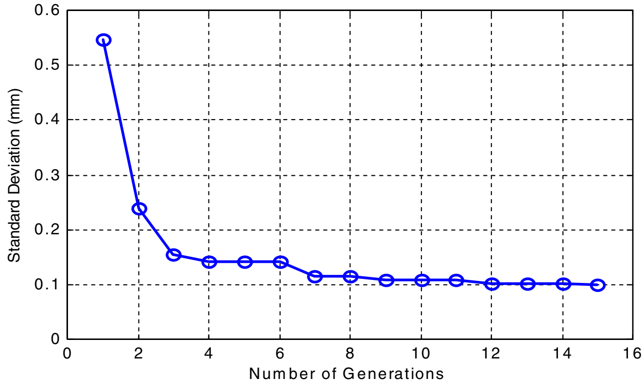

Fig. 9 Decrease of estimated membrane flatness.

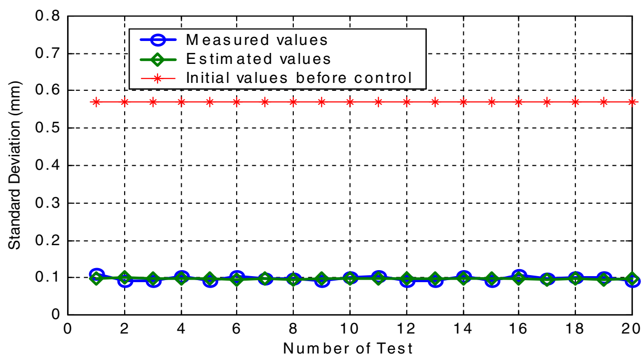

Fig. 10 Obtained membrane flatness of the 20 tests.

sampling, multipoint crossover, and discrete mutation. The membrane-flatness requirement is set at 0.1 mm (standard deviation). To test the membrane-flatness control system, the local heater under the membrane is turned on. When its temperature attains the required 200°C (the same as for the case of neural network training), the control system is activated. The genetic algorithm begins to search for the optimal-tension combination based on the obtained neural network. When the optimal-tension combination is found, it is transferred to the SMA controller and the latter exerts it through SMA actuators. After the real tensions attain the optimal values and the vision system completes the recording of the membrane flatness, the control system stops automatically. Figure 9 shows the convergence of the estimated membrane flatness, in which the 15 points marked with circles are the best values of the membrane flatness estimated in the 15 generations of optimization. To find this best tension combination, a total of 15 generations of 64 tension combinations were tested by the genetic algorithm. This convergence process takes less than 1 s for our case, in which the genetic algorithm is coded with Matlab.

Tests are performed 19 more times, and each time, the membrane-flatness requirement remains the same at 0.1 mm. The obtained results of the membrane flatness in 20 tests are shown in Fig. 10. One can see that for all 20 tests, the genetic algorithm successfully finds the optimal-tension combination. Also, the estimated membrane flatness agrees very well with the real measured values, and both are very close to the membrane-flatness requirement. These results demonstrate that the neural network scheme is effective in the estimation of the membrane flatness.

VI. Conclusions

This paper presents a genetic algorithm and neural network scheme for the active shape control of inflatable structures. Experimental tests are performed on a small membrane structure, and the results demonstrate the effectiveness of the proposed neural network scheme in the shape estimation of this type of structure. Test results also indicate that the genetic algorithm can find the optimal-tension combinations effectively and quickly. More tests are ongoing, with varying local thermal loads applied.

References

[1] Jenkins, C. H. M. (ed.), *Gossamer Spacecraft: Membrane and Inflatable Structures Technology for Space Applications*, Progress in Astronautics and Aeronautics, Vol. 191, AIAA, Reston, VA, 2001.

[2] Cadogan, D., and Grahne, M., "Inflatable Space Structures: A New Paradigm for Space Structure Design," Proceedings of the International Astronautical Congress, Melbourne, Australia, International Astronautical Federation Paper 98-I.1.02, 1998.

[3] Chambers, L. (ed.), *The Practical Handbook of Genetic Algorithms: Applications*, 2nd ed., CRC Press, Boca Raton, FL, 2000.

[4] Cybenko, G., "Approximations by Superposition of a Sigmoidal Function," *Mathematics of Control, Signals, and Systems*, Vol. 2, No. 4, 1989, pp. 303–314.

[5] Funahashi, K., "On the Approximate Realization of Continuous Mappings by Neural Networks," *Neural Networks*, Vol. 2, No. 3, 1989, pp. 183–192.

[6] Peng, F., Jiang, X. X., Hu, Y.-R., and Ng, A., "Application of Shape Memory Alloy Actuators in Active Shape Control of Inflatable Space Structures," 2005 IEEE Aerospace Conference, Inst. of Electrical and Electronics Engineers, Aerospace and Electronic Systems Society Paper 1236, 2005.

[7] Peng, F., Hu, Y.-R., and Ng, A., "Measurement of Membrane Flatness for Active Control of Inflatable Structures," *7th CanSmart Meeting International Workshop Smart Materials & Structures*, CanSmart Group, Kingston, Ontario, Canada, 2004, pp. 17–26.

C. Pierre
Associate Editor