AIAA-98-1778

A FEASIBILITY STUDY OF SYNTHESIZING SUBSTRUCTURES MODELED WITH COMPUTATIONAL NEURAL NETWORKS

John T. Wang[*], Jerrold M. Housner[†], and Z. Peter Szewczyk[‡]

NASA Langley Research Center

Hampton, Virginia 23681-0001

Abstract

This paper investigates the feasibility of synthesizing substructures modeled with computational neural networks. Substructures are modeled individually with computational neural networks and the response of the assembled structure is predicted by synthesizing the neural networks. A superposition approach is applied to synthesize models for statically determinate substructures while an interface displacement collocation approach is used to synthesize statically indeterminate substructure models. Beam and plate substructures along with components of a complicated Next Generation Space Telescope (NGST) model are used in this feasibility study. In this paper, the limitations and difficulties of synthesizing substructures modeled with neural networks are also discussed.

Introduction

Virtual product development and real time simulation are two key elements necessary for future immersive design environment [1] in which engineers will be able to create and modify their designs, and the effects of their modifications can be visualized immediately. The use of off-the-shelf components (substructures) in developing new products is faster and cheaper than creating new designs for all the components and parts. It is likely that many new aerospace structure developments may involve trading off one off-the-shelf component with another or relocating a component. Therefore, computational tools, which can facilitate the evaluation of the effects of these substructure design changes in real time, are highly desirable.

Although the finite element method is widely used for structural analysis of each component; creating new models, modifying existing models, and solving large systems of equations are all time consuming and cannot provide the computational speed required for the future immersive design environment. On the other hand, computational neural networks [2] can be trained using analytical or test data to model each component. The trained neural network model for each component (substructure) has a small number of neurons and the prediction of the component response is computationally efficient.

Currently, complete aerospace vehicles are being designed and manufactured by teams of companies. Substructure response cast into a compact neural network model can ease the transfer of design information from company to company while protecting each company's proprietary information.

For the aforementioned reasons, the capability of synthesizing substructures modeled with neural networks is critical for the success of implementing a future immersive design environment. Although the procedures of synthesizing physics-based models have been well developed [3,4], the procedures for synthesizing substructures modeled with neural networks can not be found in the literature. The only related reference is a paper written by the third author [5] during the course of this study.

The objective of this paper is to investigate the feasibility of synthesizing substructures modeled with computational neural networks. A superposition approach and an interface displacement

[*] Aerospace Engineer, CSB/SD, Member AIAA

[†] Branch Head, CSB/SD, Member AIAA

[‡] National Research Council Research Associate, Member AIAA

collocation approach are used for synthesizing substructures modeled with neural networks. Beam and plate substructures along with components of a Next Generation Space Telescope (NGST), are used to illustrate the neural network substructure coupling method. The limitations, difficulties, and future development are also discussed in this paper.

Feed-Forward Neural Networks

The inception of neural networks comes from the recognition that the human brain functions in an entirely different way from the conventional digital computer. The human brain may have more than 10 billion neurons that form a highly efficient parallel information processing network. A neural network is designed to mimic brain functions using massive interconnection processing units referred to as "neurons." A neural network may contain an input layer, an output layer, and a few hidden layers. A typical feed-forward neural network configuration used in this study is presented in the Appendix.

In the neural network model (Figure A in the Appendix), the strength of a connection between two neurons is referred to as "weight''. The data from input layer or hidden layers are multiplied by the weights associated with a next-layer neuron and sums together before processing by the neuron. Before a neural network can perform useful computations, its weights need to be adjusted through a training scheme. The back-propagation technique is the most popular training scheme in which the input data are propagated through the network (feed-forward) and the output data are calculated. The errors between the desired output and the calculated output are computed. Then a minimization procedure is used to adjust weights between two connection layers starting backward from the output layer to the input layer [6].

Back-propagation trained, multiple-layer feed-forward neural networks in MATLAB's Neural Networks Toolbox [7] are used for this study. In this work, the Levenberg-Marquard update rule [8], which combines the steepest descent search with the Gauss-Newton method is used. The weight update formula is :

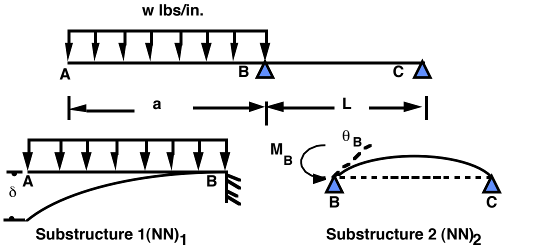
$$\Delta\mathbf{W} = (\mathbf{J}^{T}\mathbf{J} + \mu\mathbf{I})^{-1}\mathbf{J}^{T}\mathbf{e} \qquad (1)$$

where $\Delta\mathbf{W}$ is the iterative weight change, $\mathbf{J}$ is the Jacobian matrix of error derivatives with respect to the weights, μ is a scalar control parameter, $\mathbf{I}$ is a identity matrix, and the error vector is denoted by $\mathbf{e}$. If μ is large, the update formula approximates the gradient descent. If μ is small, the above formula becomes the Gauss-Newton Method. The Gauss-Newton method is faster and more accurate near an error minimum. To shift towards the Gauss-Newton method quickly, μ is decreased after each successful iteration step and increased only when a step increases the error.

Beam Problems

To explore the neural network synthesizing process, a statically determinate beam and a statically indeterminate beam are analyzed. Based on a classical analysis, the statically determinate beam is represented by two statically determinate substructures as shown in Figure 1, a uniformly loaded cantilever beam **AB** and a simply supported beam **BC**. Computational neural networks, $\mathbf{(NN)_1}$ and $\mathbf{(NN)_2}$, are used to model each substructure as shown in Figure 1. For statically determinate substructures, substructure interface loads (i.e. the bending moment in this case) from $\mathbf{(NN)_1}$ are directly used as an input to $\mathbf{(NN)_2}$ to obtain the interface displacements (i.e. the rotation in this case). The total tip deflection at point **A** is obtained by the superposition method, i.e.

$$\delta_{\mathbf{A}} = \delta + \mathbf{a} \times \theta_{\mathbf{B}} \qquad (2)$$



$$\begin{Bmatrix} \mathbf{w} \\ \mathbf{a} \end{Bmatrix} \rightarrow \mathbf{(NN)_1} \rightarrow \begin{Bmatrix} \mathbf{M_B} \\ \delta \end{Bmatrix} \qquad \{\mathbf{M_B}\} \rightarrow \mathbf{(NN)_2} \rightarrow \{\theta_{\mathbf{B}}\}$$

Figure 1. Statically determinate beam and its substructure neural networks.

A response surface of the tip deflection as a function of the distributed load (**w**) and the cantilever length (**a**), predicted by the neural networks synthesizing procedure and the analytical solution, are shown in Figure 2. The tip deflection error of the neural network prediction is plotted in Figure

2(c). The neural network coupling procedure produces good results with small amount of errors.

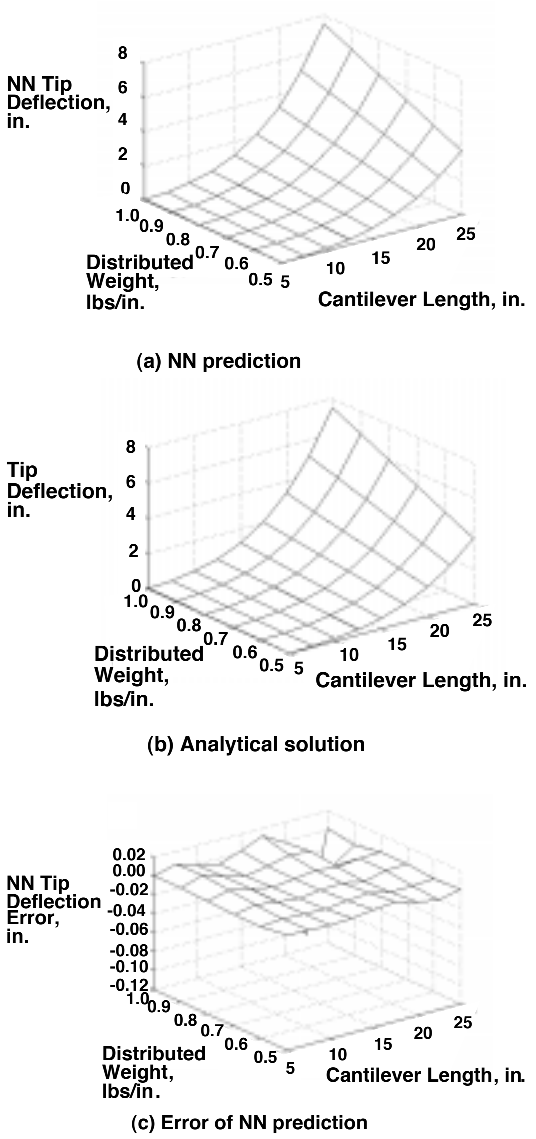


Figure 2. Comparison of neural network predictions with analytical solutions.

The statically indeterminate beam problem is shown in Figure 3. The beam is split into two statically indeterminate substructures as shown in Figures 4a and 4b. The interface loads (**F** and **M**) between the two substructures cannot be statically determined. If the range of interface loads and moments are given, substructures 1 and 2 can be represented by neural nets $(\mathbf{NN})_1$ and $(\mathbf{NN})_2$, respectively. The input to $(\mathbf{NN})_1$ is the $\mathbf{P}_1$ location (**a**), interface force (**F**), and moment (**M**). The input to $(\mathbf{NN})_2$ is the interface force, the interface moment, and the load $\mathbf{P}_2$ location (**b**). The output of both networks is displacements at location **x**.

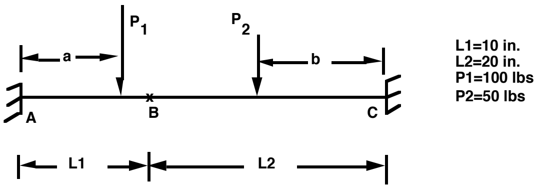
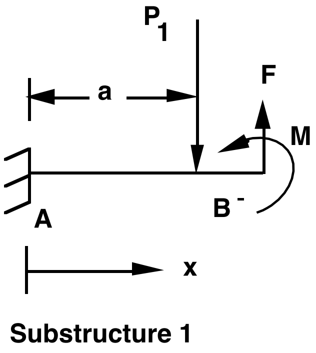
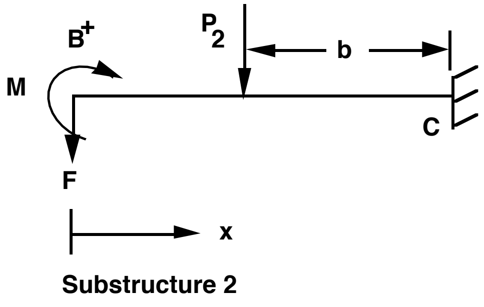


Figure 3. Statically indeterminate beam.



$$\begin{Bmatrix} \mathbf{a} \\ \mathbf{F} \\ \mathbf{M} \\ \mathbf{x} \end{Bmatrix} \rightarrow (\mathbf{NN})_1 \rightarrow \begin{Bmatrix} \delta \\ \theta \end{Bmatrix}_1$$

Figure 4a. Substructure 1 modeled with neural network $(\mathbf{NN})_1$.

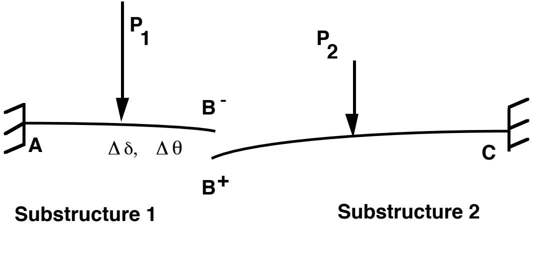


$$\begin{Bmatrix} \mathbf{b} \\ \mathbf{F} \\ \mathbf{M} \\ \mathbf{x} \end{Bmatrix} \rightarrow (\mathbf{NN})_2 \rightarrow \begin{Bmatrix} \delta \\ \theta \end{Bmatrix}_2$$

Figure 4b. Substructure 2 modeled with neural network $(\mathbf{NN})_2$.

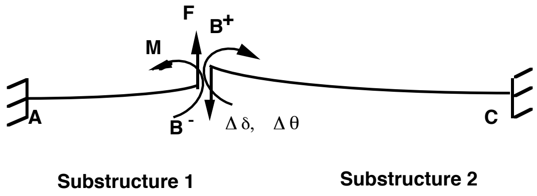
When these two substructure neural nets are synthesized, the interface forces at point **B** need to be determined from the enforcement of compatibility requirements. Figures 5 and 6 illustrate an interface displacement collocation approach in which the compatibility of displacements along the interface of adjacent substructures is established when the differential displacements induced by the applied forces are eliminated by the differential displacements induced by the interface forces. Two additional neural networks, $(\mathbf{NN})_1$* and $(\mathbf{NN})_2$*, are used to determine the interface forces. Differential displacements between point B^- and B^+ for applied loads $\mathbf{P_1}$ and $\mathbf{P_2}$ at various locations are computed. Neural network $(\mathbf{NN})_1$* is trained using load locations, $\begin{Bmatrix}\mathbf{a}\\ \mathbf{b}\end{Bmatrix}$, as input and differential displacements, $\begin{Bmatrix}\Delta\delta\\ \Delta\theta\end{Bmatrix}_{\mathbf{1*}}$, as output. After training, $(\mathbf{NN})_1$* has the capability to predict the differential displacements for applied loads at arbitrary locations within the training range.

Neural network $(\mathbf{NN})_2$* is trained to solve an inverse problem, so substructure interface forces can be determined from given differential displacements. To generate training data of $(\mathbf{NN})_2$, various pairs of forces, $\begin{Bmatrix}\mathbf{F}\\ \mathbf{M}\end{Bmatrix}$, are applied at the tips (point $\mathbf{B^+}$ and $\mathbf{B^-}$) of each substructure and differential displacements, $\begin{Bmatrix}\Delta\delta\\ \Delta\theta\end{Bmatrix}_{\mathbf{2*}}$, are computed. The neural network is trained using differential displacements as input and interface forces as output.



$$\begin{Bmatrix}\mathbf{a}\\ \mathbf{b}\end{Bmatrix} \rightarrow (\mathbf{NN})_1{}^* \rightarrow \begin{Bmatrix}\Delta\delta\\ \Delta\theta\end{Bmatrix}_{\mathbf{1*}}$$

Figure 5. Differential displacements between Point B^- and B^+ due to applied loads.



$$\begin{Bmatrix}\Delta\delta\\ \Delta\theta\end{Bmatrix}_{\mathbf{2*}} \rightarrow (\mathbf{NN})_2{}^* \rightarrow \begin{Bmatrix}\mathbf{F}\\ \mathbf{M}\end{Bmatrix}$$

Figure 6. Differential displacements between Point B^- and B^+ due to interface loads.

The neural network $(\mathbf{NN})_2$* is used to predict the interface forces required to close the differential displacements generated by network $(\mathbf{NN})_1$* for any loading locations, $\begin{Bmatrix}\mathbf{a}\\ \mathbf{b}\end{Bmatrix}$, as shown in Equation 3.

$$\begin{Bmatrix}\mathbf{a}\\ \mathbf{b}\end{Bmatrix} \rightarrow (\mathbf{NN})_1{}^* \rightarrow \begin{Bmatrix}\Delta\delta\\ \Delta\theta\end{Bmatrix}_{\mathbf{1*}} \rightarrow \begin{Bmatrix}-\Delta\delta\\ -\Delta\theta\end{Bmatrix}_{\mathbf{1*}} \rightarrow (\mathbf{NN})_2{}^* \rightarrow \begin{Bmatrix}\mathbf{F}\\ \mathbf{M}\end{Bmatrix} \qquad (3)$$

The interface forces predicted by $(\mathbf{NN})_1$* and $(\mathbf{NN})_2$* are compared with closed form solutions [9] as shown in Table 1. The differences of $(\mathbf{NN})_2$* predictions and the closed form solutions are within three percent. Once the interface forces are determined, the structural response is determined using neural networks $(\mathbf{NN})_1$ and $(\mathbf{NN})_2$.

Application of Superposition Approach to Rigidly Connected Substructures

To illustrate that the procedure of synthesizing neural network models of substructures can be used to predict the response of a large scale aerospace structure, synthesis of two NGST substructures, each modeled with a neural network, is performed. A complete finite element model of the NGST is shown in Figure 7. In this study, the optical telescope assembly and the science and spacecraft module are assumed to be rigidly connected. Concentrated loads are applied at various locations along the tube of the optical telescope assembly and the top end of the spacecraft module is assumed to be fixed. The neural network models of the substructures are integrated and used to predict the response of the full structure. Static finite element analysis results of the full model are also generated for comparison.

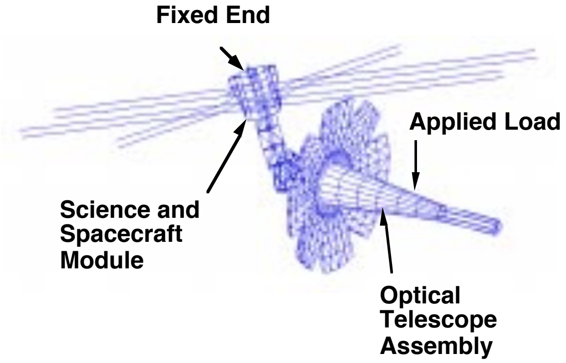


Figure 7. NGST finite element model.

Substructures of the NGST are modeled with two neural networks, $(NN)_1$ and $(NN)_2$. Procedures for creating these networks are discussed in the following sections.

$(NN)_1$: The Optical Telescope Assembly

Finite element analyses are performed on the optical-telescope-assembly substructure to generate training data. The load is applied to the tube and the interface hub is assumed to be fixed as shown in Figure 8. The $(NN)_1$ input is the applied load and the location of the applied load. The output is reaction forces at the interface hub and displacements of the secondary mirror relative to the interface hub as shown in Equation 4.

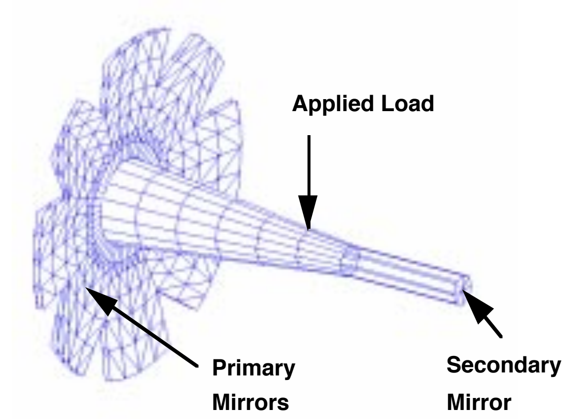


Figure 8. The optical telescope assembly.

$$\begin{Bmatrix}\text{Applied Load}\\ \text{Location}\end{Bmatrix} \rightarrow (NN)_1 \rightarrow \begin{Bmatrix}\text{Interface Hub Loads}\\ \text{Relative Displacements of}\\ \text{Secondary Mirror}\end{Bmatrix} \tag{4}$$

$(NN)_2$: The Science and Spacecraft Module

The finite element model of the science and spacecraft module is analyzed with interface hub loads in an estimated range and the top end of the spacecraft module is fixed as shown in Figure 9. Finite element results are used to train the neural net. The $(NN)_2$ input is the interface hub loads and the output is the interface hub displacements as shown in Equation 5.

$$\begin{aligned}&\{\text{Interface Hub Loads}\}\\ &\rightarrow (NN)_2 \rightarrow \{\text{Interface Hub Displacements}\}\end{aligned} \tag{5}$$

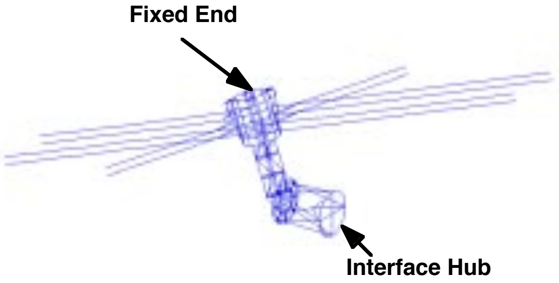


Figure 9. The science and spacecraft module.

Output of neural network $(NN)_1$ is used as input to neural network $(NN)_2$ for predicting the interface hub displacements. The NGST's second mirror deflection is predicted by the superposition approach. The results obtained from coupling these two neural networks are shown in Figure 10. Neural network results of the secondary mirror deflection are shown as open circles. The x symbols are the finite element analysis results of the whole NGST model. Note the solid lines are the least square fittings of the finite element results. Reasonably accurate predictions from the coupling of substructure neural networks are evident from Figure 10.

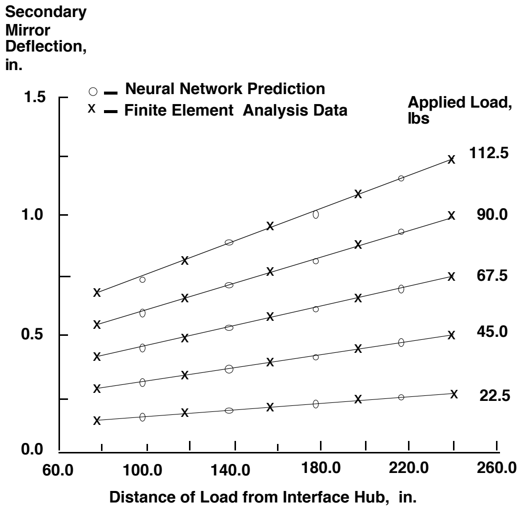


Figure 10. Comparison of the finite element model results and the neural networks results.

Application of the Interface Displacement Collocation Approach for Multipoint Connected Substructures

The beam problems and the rigidly connected NGST substructures are connected at a single point only. For multipoint connection problems, training sets and sizes of neural nets are significantly increased due to the increase of design variables. An aluminum plate shown in Figure 11 is used to illustrate a multiple point connection problem. Two plate substructures, shown in Figures 12a and 12b, are connected at three points (**A**, **B**, and **C**). Substructure 1, which includes an open hole, is subjected to a moving line load of 100 lbs/in. Design parameter, **a**, is the location of the line load. Substructure 2 is subjected to a stationary line load of 50 lbs/in. at 5-in. from the right end. The dimensions of the plate are given in Figure 11.

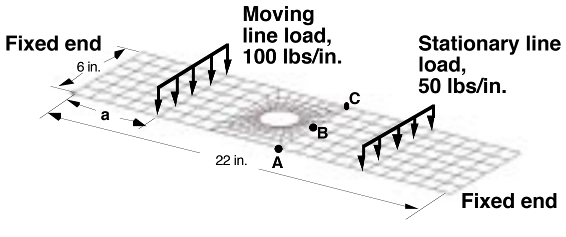
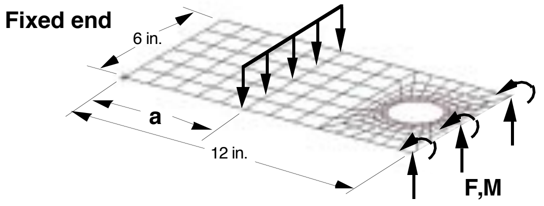

Fixed end
6 in.
a
Moving line load, 100 lbs/in.
C
B
A
22 in.
Stationary line load, 50 lbs/in.
Fixed end

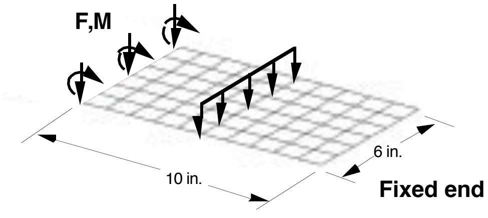
Figure 11. Assembly of aluminum plate substructures, 0.2 inches thick.

Two neural nets, $(\mathbf{NN})_1$ and $(\mathbf{NN})_2$, are trained to model the responses of substructures 1 and 2, respectively. Figure 12a shows that the input of the $(\mathbf{NN})_1$ includes the line load location, **a** , the interface force **F**, moment **M**, and finite-element nodal coordinates **x** and **y**. The output of $(\mathbf{NN})_1$ is the nodal displacements, $\begin{Bmatrix}\delta\\ \theta\end{Bmatrix}_1$. Figure 12b shows the input of $(\mathbf{NN})_2$ is the interface force **F**, moment **M**, and nodal coordinates **x** and **y**; the output of $(\mathbf{NN})_2$ is the nodal displacements, $\begin{Bmatrix}\delta\\ \theta\end{Bmatrix}_2$. These nets are used to predict the responses of each substructure once the interface forces are determined.



$$\begin{Bmatrix}\mathbf{a}\\ \mathbf{F}\\ \mathbf{M}\\ \mathbf{X}\\ \mathbf{Y}\end{Bmatrix} \rightarrow (\mathbf{NN})_1 \rightarrow \begin{Bmatrix}\delta\\ \theta\end{Bmatrix}_1$$

Figure 12a. Substructure 1 modeled with $(\mathbf{NN})_1$.



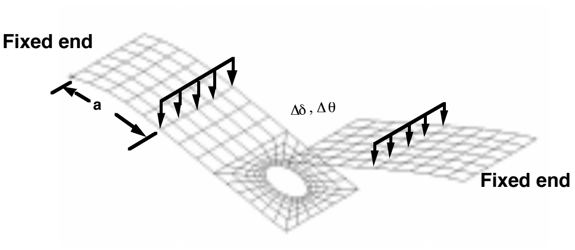
$$\begin{Bmatrix}\mathbf{F}\\ \mathbf{M}\\ \mathbf{X}\\ \mathbf{Y}\end{Bmatrix} \rightarrow (\mathbf{NN})_2 \rightarrow \begin{Bmatrix}\delta\\ \theta\end{Bmatrix}_2$$

Figure 12b. Substructure 2 modeled with $(\mathbf{NN})_2$.

The interface displacement collocation approach is applied to determine the interface forces. Two additional neural nets, **(NN)**$_1$* and **(NN)**$_2$*, are created to determine the interface forces and moments between these two substructures. The procedures used to create the nets are as follows.

Neural Net **(NN)**$_1$*

To generate training data, finite element analyses are performed on each substructure and the interface differential displacements are computed as shown in Figure 13. Note that the substructures are subjected to applied line loads only and there are no interface forces applied on each substructure. By moving the line load on Substructure 1, corresponding interface differential displacements can be computed. These data are used to train the neural network **(NN)**$_1$*.

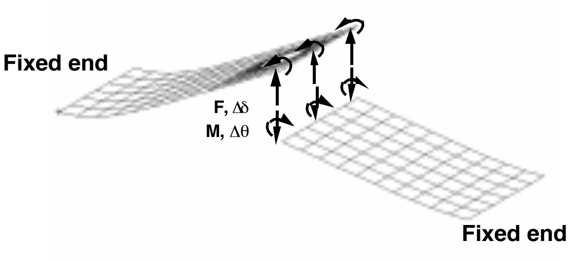


$$\{\mathbf{a}\} \rightarrow (\mathbf{NN})_{\mathbf{1}}{}^{*} \rightarrow \begin{Bmatrix}\Delta\delta \\ \Delta\theta\end{Bmatrix}_{\mathbf{1}^*}$$

Figure 13. Substructures subjected to applied loads.

Neural Net **(NN)**$_2$*

To couple the two plate substructures to form an assembled structure, the interface connecting points are collocated by applying interface forces to eliminate the differential displacements created by the line loads. This neural net establishes the relationship between the differential displacements and the interface forces as shown in Figure 14. To generate training data for **(NN)**$_2$*, forces and moments are applied at each of the three connecting points. Although these loads and moments are unknown, the range of each of these interface loads, however, can be estimated from a similar design or from several finite element analyses of the assembled structure.



$$\begin{Bmatrix}\Delta\delta \\ \Delta\theta\end{Bmatrix}_{\mathbf{2}^*} \rightarrow (\mathbf{NN})_{\mathbf{2}}{}^{*} \rightarrow \begin{Bmatrix}\mathbf{F} \\ \mathbf{M}\end{Bmatrix}$$

Figure 14. Substructures subjected to interface forces.

The estimated ranges of each force and each moment are divided by N data points which are uniformly distributed in the range of interest. If there are M connecting points and each connecting point has K connecting forces, the total number of loading combinations is N^{M*K}. We use N=8, M=2 (symmetry), and K=2 for this plate problem. A total of 4,096 load combinations are created to generate training data of **(NN)**$_2$*. The number of load combinations can be very large when the number of connecting points is increased. Fortunately, these loading cases are at the right hand side of the finite element system equations of each substructure and back substitution can be used to obtain the displacements at the interface of each substructure for computing the interface differential displacements. **(NN)**$_2$* is trained for predicting interface forces using interface differential displacements as input.

In the substructure synthesis process, **(NN)**$_1$* predicts the interface differential displacements, $\begin{Bmatrix}\Delta\delta \\ \Delta\theta\end{Bmatrix}_{\mathbf{1}^*}$, when the line load on substructure 1 moves to a new location. By inputting the negative interface differential displacements, $\begin{Bmatrix}-\Delta\delta \\ -\Delta\theta\end{Bmatrix}_{\mathbf{1}^*}$, to **(NN)**$_2$*, the interface forces are determined.

Results of interface forces and moments predicted by neural networks are compared with results from finite element analysis in which multipoint constraints are used to connecting substructures. Figures 15 and 16 show the comparisons of the neural network prediction of shear forces (**F**) and bending moments (**M**), respectively, at three connecting points along the interface and those predicted by finite element analysis of the assembled structures. Good correlation exists between the finite element results and the neural network predictions.

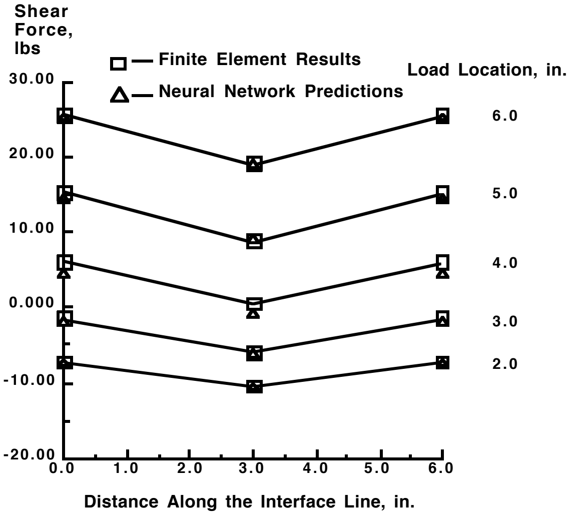


Figure 15. Comparison of interface forces predicted by neural network synthesis method and finite element analysis.

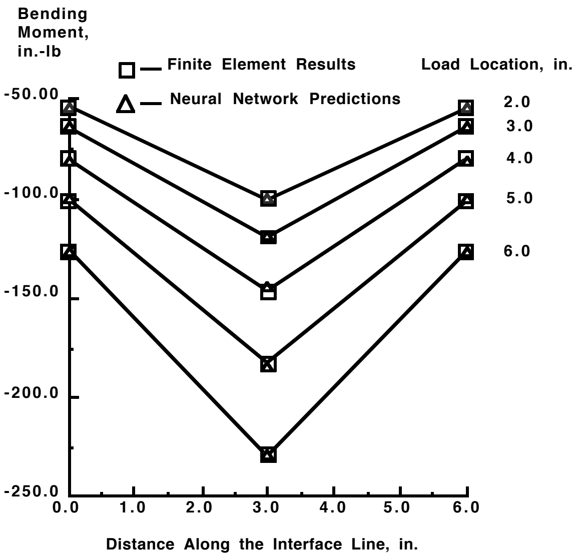


Figure 16. Comparison of interface moments predicted by neural network synthesis method and finite element analysis.

Once the interface forces and moments are determined, the responses of each substructure are instantaneously obtained from their representative neural nets, $(NN)_1$ and $(NN)_2$, respectively. The capability of instant response prediction by neural networks for examining the effect of design change in real time is very suitable for the future immersive "virtual reality" design environment. Figure 17 shows the deformed shape predicted by $(NN)_1$ and $(NN)_2$ for a moving line load located at 6-in. from the left end. The displacement compatibility at Point A, B and C is well maintained.

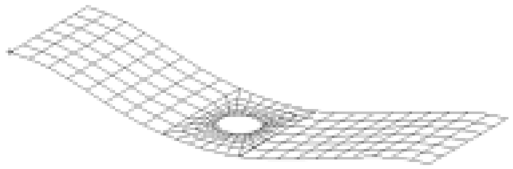


Figure 17. The deformed shape predicted by $(NN)_1$ and $(NN)_2$.

Concluding Remarks

This study found that it is feasible to predict an assembled structure's response from the synthesis of neural network substructure representations. Two approaches have been used in this study, a superposition approach and an interface displacement collocation approach. The superposition approach is only valid for statically determinate beam substructures and rigidly connected substructures. The interface displacement collocation approach is more general and can be used for statically indeterminate beam substructures and multipoint connected plate substructures. The responses predicted by the neural network synthesis method are in good agreement with the responses predicted by closed form analysis and finite element analysis.

Although generating the training data, creating neural networks for each substructure, and synthesizing them can be time consuming; these tasks, fortunately, only need to perform once and can be done beforehand. The response of an assembled structure can then be obtained almost instantaneously by synthesizing various neural network substructure representations. This allows the effect of design changes on structural response to be examined nearly in real time.

There are still many issues to be resolved before this substructure neural network synthesis method can be a viable future design tool. Among them are (1) how to reduce the large amount of training data required for multipoint connected substructures, (2) how to improve the training process when the size of a neural net becomes large, and (3) how to better determine the range of interface forces for training.

Appendix - Feed-forward Neural Network

Figure A shows the configuration of a feed-forward network with two hidden layers. The mathematical expression of this neural network can be written as

$$y3=F3(w3*F2(w2*F1(w1*x)))$$

where **x** is the input vector; **y1** and **y2** are the intermediate output vectors from the hidden layers; **y3** is the output vector; **w1**, **w2**, and **w3** are the weight matrices; * is a multiplication operator, **F1**, **F2**, and **F3** are transfer functions "neurons" of the first hidden layer, the second hidden layer, and the output layer. These transfer functions can be a Sigmoid transfer function,

$$F(x) = 1/(1+e^{-x})$$

or hyperbolic tangent function,

$$F(x) = \tanh(x).$$

For linear approximation, a linear transfer function may be used. It is recommended that the input data and output data be mapped into an interval [0.1,0.9] to speed up the training process.

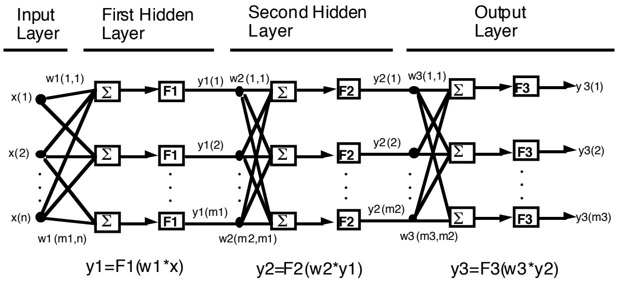


Figure A. A configuration of a typical feed-forward network.

References

1. Noor, A. K., Venneri, S. L., Housner, J. M., and Peterson, J. C., "A Virtual Environment for Intelligent Design," *Aerospace America*, April 1997.
2. Haykin, S., *Neural Networks*, Prentice-Hall, Inc. 1994, ISBN 0-02-352761-7.
3. Noor, A. K., Kamel, H. A., and Fulton, R. E., "Substructuring Techniques - status and projections," *Computers & Structures*, Vol. 8, No. 5, 1978, pp. 621-632.
4. Craig, R. R., Jr., and Bampton, M. C. C., "Coupling of Substructures for Dynamic Analyses, " *AIAA Journal*, Vol. 6, No. 7, July 1968, pp. 1313-1319.
5. Szewczy, Z. P., "Data-based Modeling for Interactive, Virtual Prototyping," AIAA paper 98-1011, Presented in *36th Aerospace Sciences Meeting and Exhibit*, Reno, January 12-15, 1998.
6. Rumelhart, D. E., Hinton, G. E., and Williams, R. J., "Learning Internal Representations by Error Propagation," D. Rumelhart and J. McClelland, editors. *Parallel Data Processing*, Vol. 1, Chapter 8, the M.I.T. Press, Cambridge, MA 1986, pp. 318-362.
7. Demuth, H., and Beale, M., *Neural Network Toolbox User's Guide*, The MathWorks, Inc., January 1994.
8. Hagen, M. T., and Menhaj, M., "Training Feedforward Networks with the Marquart algorithm," *IEEE Transitions on Neural Networks*, Vol. 5, No. 6, November 1994, pp. 989-993.
9. Roark, R. J., and Young, W. C., *Formulas for Stress and Strain*, 5th edition, McGraw Hill, Inc., 1975, ISBN 0-07-053031-9.

Table 1. Interface forces and moments predicted by neural network synthesis method and closed form solution.

Case	a (in.)	b(in.)	Force(lbs)	Force(lbs)	Moment(in.-lb)	Moment(in.-lb)
			(NN)	(Ref. 9)	(NN)	(Ref. 9)
1	4.25	8.5	-4.435	-4.315	67.225	68.738
2	4.75	9.5	-5.046	-5.139	88.688	87.116
3	5.25	10.5	-6.008	-5.972	106.083	107.953
4	5.75	11.5	-6.900	-6.796	134.879	131.332