MODEL REDUCTION FOR THE DYNAMICS AND CONTROL OF LARGE STRUCTURAL SYSTEMS VIA NEURAL NETWORK PROCESSING DIRECT NUMERICAL OPTIMIZATION[1]

Georges A. Bécus and Alistair K. Chan
Aerospace Engineering and Engineering Mechanics
University of Cincinnati
Cincinnati, OH

ABSTRACT

Three neural network processing approaches in a direct numerical optimization model reduction scheme are proposed and investigated.

INTRODUCTION

Large structural systems, such as large space structures, offer new challenges to both structural dynamicists and control engineers. One such challenge is that of dimensionality. Indeed these distributed parameter systems can be modeled either by infinite dimensional mathematical models (typically partial differential equations) or by high dimensional discrete models (typically finite element models) often exhibiting thousands of vibrational modes usually closely spaced and with little, if any, damping. Clearly, some form of model reduction is in order, especially for the control engineer who can actively control but a few of the modes using system identification based on a limited number of sensors. Inasmuch as the amount of "control spillover" (in which the control inputs excite the neglected dynamics) and/or "observation spillover" (where neglected dynamics affect system identification) is to a large extent determined by the choice of a particular reduced model (**RM**), the way in which this model reduction is carried out is often critical.

Different techniques to obtain **RM**'s have been proposed by various authors. While they are based on the same philosophy of retaining only those modes which play a significant role, they differ in the way the roles of the modes are quantified. Among these techniques we mention: (i) Modal Truncation; (ii) Balanced Controller Reduction; (iii) Component Cost Analysis; (iv) Optimal Projection Conditions; (v) Energy Based Model Reduction (also referred to as Modal Performance Tracking); (vi) Subsystem Balancing. (See [1] for references on methods (ii-iv), [2] and the references therein for method (v) and [3] for (vi).)

Model reduction can also be viewed as providing an answer to the question: What are the $m < n$ linear combinations of the $n \leq \infty$ states of the full model which best describe the behavior of the system? The various techniques only differ in the way "best" is defined. As such, model reduction is an optimization

[1] The work of both authors was supported in part by NASA-Lewis Research Center under Grant NAG3-1174.

problem. In fact, most model reduction schemes first attempt to find an analytical solution to the optimization problem, using necessary optimality conditions to obtain one or several equations to be satisfied by the solution and which can then be solved in an iterative numerical scheme. Viewed in this light, most currently available model reduction schemes suffer from three shortcomings: (i) they are restricted to optimality criteria for which a (partial) analytical solution to the optimization problem can be found, (ii) being based on necessary conditions, they cannot guarantee that the solution so obtained is the actual optimum sought, and (iii) the iterative numerical construction of the solution can be a formidable task. Recently, to alleviate the above shortcomings, we proposed to carry model reduction by direct numerical solution of the optimization problem [4]. In this paper we propose and investigate the use of neural network processing methods to carry out this direct optimization. First we review the direct numerical optimization approach proposed in [4].

DIRECT NUMERICAL OPTIMIZATION METHOD

Consider the n-th order linear time-invariant state space model of a large structural system

$$\dot{\mathbf{x}} = \mathbf{A}\,\mathbf{x} + \mathbf{B}\,\mathbf{u} \tag{1a}$$

$$\mathbf{y} = \mathbf{C}\,\mathbf{x}\;. \tag{1b}$$

Here $\mathbf{x}$, $\mathbf{u}$ and $\mathbf{y}$ are the n, r and p-dimensional state, input and output vectors respectively, $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are constant matrices of appropriate dimensions and the system is assumed to be completely controllable. Model reduction consists of finding a model of order m<n

$$\dot{\mathbf{x}}_{\mathbf{m}} = \mathbf{A}_{\mathbf{m}}\,\mathbf{x}_{\mathbf{m}} + \mathbf{B}_{\mathbf{m}}\,\mathbf{u} \tag{2a}$$

$$\mathbf{y}_{\mathbf{m}} = \mathbf{C}_{\mathbf{m}}\,\mathbf{x}_{\mathbf{m}}\;. \tag{2b}$$

Here $\mathbf{x}_{\mathbf{m}}$ and $\mathbf{y}_{\mathbf{m}}$ are m and p-dimensional state and output vectors, while $\mathbf{A}_{\mathbf{m}}$, $\mathbf{B}_{\mathbf{m}}$ and $\mathbf{C}_{\mathbf{m}}$ are constant matrices of appropriate dimensions, which "best approximates" the full order model (1a,b).

In this paper, as in [4], we restrict ourselves to model reduction schemes based on an integral-square-error performance index (in particular to the optimal projection method of Hyland and Bernstein), [1,5], but the methodology is applicable to other schemes as well. We are thus interested in determining matrices $\mathbf{A}_{\mathbf{m}}$, $\mathbf{B}_{\mathbf{m}}$ and $\mathbf{C}_{\mathbf{m}}$ which minimize

$$J(\mathbf{A}_{\mathbf{m}},\mathbf{B}_{\mathbf{m}},\mathbf{C}_{\mathbf{m}}) = \lim_{t \to \infty} \mathbf{E}[(\mathbf{y}-\mathbf{y}_{\mathbf{m}})^{\mathbf{T}}\mathbf{R}(\mathbf{y}-\mathbf{y}_{\mathbf{m}})] \tag{3}$$

when $\mathbf{u}$ is white noise with intensity $\mathbf{V}$. In (3) $\mathbf{E}[\;]$ denotes expected value and $\mathbf{R}$ is a positive definite weighting matrix.

Introducing the augmented system of order n+m

$$\dot{x}_a = A_a x_a + B_a u \tag{4a}$$
$$y_a = C_a x_a , \tag{4b}$$

where

$$x_a = \begin{bmatrix} x \\ x_m \end{bmatrix}, \quad y_a = y - y_m, \quad A_a = \begin{bmatrix} A & 0 \\ 0 & A_m \end{bmatrix}, \quad B_a = \begin{bmatrix} B \\ B_m \end{bmatrix}, \quad C_a = [C \;\; -C_m], \tag{5}$$

the optimality criterion (3) is written as

$$J(A_m, B_m, C_m) = \lim_{t \to \infty} E[y_a{}^T R\, y_a] = tr[Q_a R_a] \tag{6}$$

where Q_a is the positive semidefinite solution of the Lyapunov equation

$$0 = A_a Q_a + Q_a A_a^T + B_a V B_a^T \tag{7}$$

and

$$R_a = C_a R C_a^T \; . \tag{8}$$

The model reduction problem has been recast as the optimization problem:

$$\min \; tr[Q_a R_a] \tag{9}$$
$$\text{subject to} \qquad 0 = A_a Q_a + Q_a A_a^T + B_a V B_a^T \; . \tag{7}$$

Similar results hold for other integral-square-error performance indices (see [6] for example).

Introducing the partition

$$Q_a = \begin{bmatrix} Q_1 & Q_2 \\ Q_2^T & Q_m \end{bmatrix}, \tag{10}$$

compatible with partitions (5), the constraint (7) is decomposed as

$$0 = A\, Q_1 + Q_1 A^T + B\, V B^T \tag{11a}$$
$$0 = A\, Q_2 + Q_2 A_m^T + B\, V B_m^T \tag{11b}$$
$$0 = A_m Q_m + Q_m A_m^T + B_m V B_m^T \, . \tag{11c}$$

Note, from (11a), that Q_1 is completely determined from knowledge of the full model. Thus expanding the objective function in (9) and neglecting the constant term involving $\mathbf{Q}_1$, the optimization problem (9,7) is rewritten as

$$\min \ \{\mathrm{tr}[\mathbf{Q}_m \mathbf{C}_m^T \mathbf{R} \mathbf{C}_m] - 2\,\mathrm{tr}[\mathbf{Q}_2^T \mathbf{C}^T \mathbf{R} \mathbf{C}_m]\} \tag{12}$$

$$\text{subject to} \quad \mathbf{0} = \mathbf{A}\,\mathbf{Q}_2 + \mathbf{Q}_2\,\mathbf{A}_m^T + \mathbf{B}\,\mathbf{V}\,\mathbf{B}_m^T \tag{11b}$$

$$\mathbf{0} = \mathbf{A}_m\,\mathbf{Q}_m + \mathbf{Q}_m\,\mathbf{A}_m^T + \mathbf{B}_m\,\mathbf{V}\,\mathbf{B}_m^T \ . \tag{11c}$$

Note that all of the above manipulations were aimed at transforming the statement of the optimization problem and not at obtaining a (partial) solution. Thus, this approach does indeed alleviate the first two of the shortcomings mentioned earlier since it is not restricted to particular optimality criteria (although it was illustrated here for a particular one), and it is guaranteed to yield at least a local minimum. In addition, we can choose the numerical optimization scheme which is best adapted to the particular optimization problem which the RM must satisfy. In [4] some promising preliminary results for a classic and somewhat pathological example [5,7] and the use of a generalized reduced gradient algorithm [8] were presented. Here we investigate the feasibility of using neural network processing methods to solve the optimization problem (9,7) or (12,11b,c). Improving the computational efficiency for large problems through massive parallelization is the motivation for using these methods, thus alleviating the third shortcoming.

NEURAL NETWORK PROCESSING METHOD

The neural network processing method is an extension of the Hopfield neural network model [9] which has been successfully used to solve combinatorial optimization problems such as the Travelling Salesman problem. Developed by W. Jeffrey and R. Rosner to solve a class of ill posed inverse problems, the neural network processing method [10] is a reformulation of the Hopfield model. Our aim is to apply this methodology to the model reduction problem. We begin with some details of the method.

Consider a network, possibly modeled by analog electronic components, the energy $\mathbf{E}$ of which at any time can be expressed as a quadratic function of its state $\mathbf{x}$ as

$$\mathbf{E}(\mathbf{x}) = -\,\mathbf{x}\,\mathbf{W}\,\mathbf{x} + 2\mathbf{T}^T\,\mathbf{x} \ . \tag{13}$$

$E(\mathbf{x})$ can be regarded as the objective function in an optimization problem for which $\mathbf{x}$ is the design variable. Matrix $\mathbf{W}$ and vector $\mathbf{T}$ are constant valued and arise from the mapping of the optimization problem into the above format.

The change in the energy function resulting from a discrete step, i.e. a change $\Delta\mathbf{x}_k$ in a single element $\mathbf{x}_k$ of $\mathbf{x}$, can be shown to be given as

$$\Delta\mathbf{E}_k = (\,-2\,\mathbf{w}_k\,\mathbf{x} + 2\,\mathbf{T}_k - \mathbf{w}_{kk}\,\Delta\mathbf{x}_k\,)\,\Delta\mathbf{x}_k \tag{14}$$

where $\Delta \mathbf{x}_k = \lambda_k (-2\, \mathbf{w}_k \mathbf{x} + 2\, \mathbf{T}_k)$, $\mathbf{w}_k$ being the k-th row of **W**, $\mathbf{w}_{kk}$ the k,k-th element of **W**, $\mathbf{T}_k$ the k-th element of **T** and λ_k the step size for $\Delta \mathbf{x}_k$. The parallel processing capabilities come into play here since all the elements of **x** can be changed simultaneously, increasing the computational speed.

We now continue changing **x** in this manner until $\Delta \mathbf{E}_k = 0$ for all k. The state so obtained represents a minimum energy state. By adjusting the size of λ_k we can show that $\Delta \mathbf{E}_k \leq 0$ for all $\Delta \mathbf{x}_k$. Since we can reduce equation (14) to

$$\Delta \mathbf{E}_k = \left(\frac{1}{\lambda_k} - \mathbf{w}_{kk} \right) (\Delta \mathbf{x}_k)^2 \ , \tag{15}$$

then $\Delta \mathbf{E}_k = 0$ when $\frac{1}{\lambda_k} \leq \mathbf{w}_{kk}$ for $\lambda_k < 0$.

Hopfield and Tank [9] showed that the stable state reached is a minimum for the optimization problem. Jeffrey and Rosner [10] extended this formulation by allowing for higher order (i.e. non quadratic) terms to be included in the energy function when necessary. The details of their formulation, being similar to the analysis just presented, are not given here.

Note that the neural network processing method of Jeffrey and Rosner is restricted to unconstrained optimization problems. Before applying it to the model reduction application at hand, the constrained optimization problem (9,7) or (12,11b,c) must first be recast as an unconstrained one. We now present three ways in which this can be accomplished: first a penalty function approach, then by solving the problem as a sequence of unconstrained problems in a multi-stage approach, and finally a substitution approach in which the constraint equation is solved and substituted into the objective function.

PENALTY FUNCTION APPROACH

The penalty function approach incorporates all of the constraints into the energy function via penalty terms. The problem becomes an unconstrained problem for the penalty function. This is accomplished in two steps:

1. The equality constraints (7) or (11b,c) are incorporated into the energy function to create a modified Lagrangian or penalty function [11] , that is:

$$\mathbf{E(x)} = \mathbf{F(x)} + \sum_{i,j} [\phi\, \mathbf{h}_{ij}^2 (\mathbf{x}) + \gamma\, \mathbf{l}_{ij}\, \mathbf{h}_{ij}(\mathbf{x})] \tag{16}$$

where $\mathbf{F(x)}$ is the objective function of the constrained problem ($\mathrm{tr}[\mathbf{Q}_a \mathbf{R}_a]$ or $\{\mathrm{tr}[\mathbf{Q}_m\, \mathbf{C}_m^T\, \mathbf{R}\, \mathbf{C}_m]$ - $2\mathrm{tr}[\mathbf{Q}_2^T\, \mathbf{C}^T\, \mathbf{R}\, \mathbf{C}_m]\}$ for the problem at hand), ϕ and γ are penalty parameters, $\mathbf{l}_{ij}$ are Lagrange multipliers and

h_{ij} is the i,j-th element of the equality constraint.

2. The underlying inequality constraint $\mathbf{Q}_a \geq 0$ is enforced by factoring $\mathbf{Q}_a$ as the product of an upper triangular matrix $\mathbf{M}_a$, partitioned as

$$\mathbf{M}_a = \begin{bmatrix} \mathbf{M}_1 & \mathbf{M}_2 \\ 0 & \mathbf{M}_m \end{bmatrix}, \tag{17}$$

and its transpose. In (17) $\mathbf{M}_1$ and $\mathbf{M}_m$ are upper triangular matrices such that $\mathbf{Q}_2 = \mathbf{M}_2 \mathbf{M}_m^T$ and $\mathbf{Q}_m = \mathbf{M}_m \mathbf{M}_m^T$. These are substituted into the energy function so that the vector of design variables $\mathbf{x}$ is made up of (i) elements of $\mathbf{A}_m$, (ii) elements of $\mathbf{B}_m$, (iii) elements of $\mathbf{M}_2$, (iv) non zero (i.e. upper triangular) elements of $\mathbf{M}_m$, and (iv) $\mathbf{l}_{ij}$ the Lagrange multipliers.

The Modified Differential Multiplier Method (MDMM), proposed by Platt [12] for use in neural network processing, is then used to solve the problem. This essentially amounts to applying gradient ascent on the Lagrange multipliers while applying gradient descent on all of the other design variables.

MULTI-STAGE APPROACH

The multi stage approach is loosely based on a model reduction algorithm proposed by Wilson [13]. It is simply the following algorithm:

1. Pick initial guesses for matrices $\mathbf{A}_m$ and $\mathbf{B}_m$.

2. Calculate $\mathbf{Q}_2$ and $\mathbf{Q}_m$.

3. Minimize the objective function using the neural network processing method with elements of $\mathbf{B}_m$ as the only design variables.

4. Update the $\mathbf{A}_m$ matrix using $\mathbf{A}_m = \mathbf{Q}_m^{-1} \mathbf{Q}_2^T \mathbf{A} \mathbf{Q}_2 \mathbf{Q}_m^{-T}$. (This is analagous to the necessary condition for an optimum used by Wilson [13].)

5. Go to step 2 until the objective function stops changing from iteration to iteration

Note that in this approach, the minimization problem of step 3 is an unconstrained problem. Thus the model reduction problem is solved as a sequence of unconstrained optimization problems.

SUBSTITUTION APPROACH

In the substitution approach the $\mathbf{Q}_2$ and $\mathbf{Q}_m$ matrices, as solutions of (11b,c), are functions of $\mathbf{A}_m$ and $\mathbf{B}_m$ which are substituted in the objective function of (12) to yield an unconstrained problem where the elements of $\mathbf{A}_m$ and $\mathbf{B}_m$ are the only design variables. Neural network processing is then used with the energy function $\mathbf{E} = \mathrm{tr}[\mathbf{Q}_m(\mathbf{A}_m,\mathbf{B}_m)\, \mathbf{C}_m^T\, \mathbf{R}\, \mathbf{C}_m] - 2\, \mathrm{tr}[\mathbf{Q}_2^T(\mathbf{A}_m,\mathbf{B}_m)\, \mathbf{C}^T\, \mathbf{R}\, \mathbf{C}_m]$.

RESULTS

In all examples considered we assumed that actuators and sensors were collocated so that $\mathbf{B} = \mathbf{C}^T$ and $\mathbf{B}_m = \mathbf{C}_m{}^T$, and, without loss of generality, that **R** and **V** are identity matrices of appropriate dimensions.

All three methods presented solved only problems of a very limited scope: all methods were able to solve very small real eigenvalue problems, but all showed an inability to solve problems of a practical size and nature. For example all three methods yielded an optimal solution for the following very simple problem considered in [4] (and given here with its solution)

$$\mathbf{A} = \begin{bmatrix} -.005 & -.99 \\ .99 & -5000 \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} 1 \\ 100 \end{bmatrix}, \ \mathbf{A}_m = [-4998.1], \ \mathbf{B}_m = [100.0], \ \mathbf{obj} = -10004.0 \ .$$

The point of interest of this example is that some model reduction techniques yield a solution corresponding to a maximum rather than the minimum [5,7].

a. Penalty Function Approach

The penalty function approach exhibited poor performance in solving model reduction problems. It was able to solve problems in which the original **A** matrix was 4x4 and the reduced matrix $\mathbf{A}_m$ was 2x2; however, this was the largest problem that we were able to solve using this method. The encouraging fact is that the method did yield good, possibly optimal, solutions to a few small problems with complex eigenvalues. For example the following problem (given here with its solution) was solved successfully

$$\mathbf{A} = \begin{bmatrix} -.1 & -10 & 0 & 0 \\ 10 & -.1 & 0 & 0 \\ 0 & 0 & -.5 & -15 \\ 0 & 0 & 15 & -.5 \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} 1 \\ 3 \\ -2 \\ 3 \end{bmatrix}, \ \mathbf{A}_m = \begin{bmatrix} -.124 & -10.075 \\ 9.924 & -.0794 \end{bmatrix}, \ \mathbf{B}_m = \begin{bmatrix} -2.867 \\ -1.392 \end{bmatrix}, \ \mathbf{obj} = -253.4 \ .$$

Difficulties with this approach were due to a lack of good guiding principles in setting step size and penalty parameters, a slow convergence, and an apparent large number of local minima.

b. Multi Stage Approach

The multi stage approach exhibited a slightly different behavior. Since the traditional optimization portion of the algorithm which was carried out using neural network processing involved a much smaller problem, the method was able to solve overall larger problems. However, the approach would not solve problems with complex eigenvalues but would successfully solve problems with strictly real eigenvalues. The maximum size of these models were 6 inputs, 6 outputs with 16x16 **A** matrices. As problems with strictly real eigenvalues have little practical application, this approach was abandoned.

c. Substitution Approach

The substitution approach presented basically the same difficulties as the penalty approach. Although it successfully solved the example given in the penalty function approach subsection above, yielding the same solution, it showed limitations in that it was unable to solve problems with **A** matrices bigger than 4x4.

CONCLUDING REMARKS

The results obtained so far have not lived up to our expectations when we embarked on this investigation. In all fairness it must be pointed out that the difficulties encountered do not appear to be a result of the neural network processing approach. Parallel investigations using a standard optimization software package [8] were also disappointing. The difficulty appears to stem from the fact that the objective function has apparently a large number of local minima. In particular, it appears that any reasonable starting point is a local minimum!

A positive result in our lack of success in solving practical sized problems is the development of a type of modal cost analysis based on the objective function developed for the optimization methods. In this method we transform the system matrices such that the **A** matrix has 2x2 blocks on the main diagonal, each block corresponding to a mode of the structural system, and the **B** matrix is consistent with these new coordinates. Next we calculate the objective function for each 2x2 system individually. The objective values for all of the individual (1 mode) reduced models are sorted and the lowest ones are retained. At this time we have not put enough time into this approach to make any firm statement about the quality and cost of these solutions. However preliminary results are encouraging. We have reduced models with **A** matrices up to 168x168 (the JPL/AFAL experiment structure) down to $\mathbf{A}_m$ matrices of 108x108 yielding excellent results when looking at the time response characteristics. We are now looking into this method in more detail to see if this approach can be used to obtain directly or aid us in finding optimal reduced models. Results will be reported elsewhere as they become available.

REFERENCES

1. Hyland, D.C., "Comparison of Various Controller-Reduction Methods: Suboptimal versus Optimal Projection," **AIAA Paper 84-1033**, 1984.

2. Bécus, G.A. and Cheng, C.W., "Energy Methods for Model Reduction in the Dynamics and Control of Large Structural Systems," **Proc. 3rd Int. Conf. Recent Adv. Structural Dyn.**, Southampton, U.K., 901-910, 1988.

3. Williams, T. and Mostarshedi, M., "Model Reduction Results for Flexible Space Structures," **5th NASA/DoD CSI Tech. Conf., NASA CP- 31**77, 1992.

4. Bécus, G.A., "Direct Numerical Optimization Methods for Model Reduction in the Dynamics and Control of Large Structural Systems," **3rd AF/NASA Symp. Recent Adv. Multidisc. Anal. and Optim.**, San Francisco, CA, 1990.

5. Hyland, D.C. and Bernstein, D.S., "The Optimal Projection Equations for Model Reduction and the Relationships Among the Methods of Wilson, Skelton, and Moore," **IEEE Trans. Aut. Con.**, AC-30, 1201-1211, 1985.

6. Bryson, A.E. Jr. and Carrier, A., "Second order algorithm for optimal model order reduction," **J.Guid. Cont. Dyn., 13,** 887-892, 1990.

7. Kabamba, P.T., "Balanced gains and their significance for balanced model reduction," **IEEE Trans. Aut. Con., AC-30**, 690-693, 1985.

8. Lasdon, L.S. and Waren, A.D., **GRG2 User's Guide**, Univ. of Texas at Austin, 1982.

9. J. J. Hopfield & D. W. Tank, "Neural Computations of Decisions in Optimization Problems," **Biological Cybernetics**, **52**, pp. 141-152, 1985.

10. Jeffrey, W. and R. Rosner, "Optimization Algorithms: Simulated Annealing and Neural Network Processing," **The Astrophysical Journal, 310**, 473-481, 1986.

11. Reklaitis, G. V., A. Ravindran, and K. M. Ragsdell, **Engineering Optimization Methods and Applications**, Wiley and Sons, NY, 1983.

12. Platt, J. C. & A. H. Barr, "Constrained Differential Optimization," in Anderson, Dana Z. (ed.), **Neural Information Processing Systems**, 612-621. American Institute of Physics, NY, 1988.

13. Wilson, D. A., "Optimum solution of Model-Reduction Problem," **Proc. IEE, 117**, 1161-1165, 1970.