Substituting $\mathbf{Y}$ and W into Eqs. (2–5) gives the closed-loop eigenvalues, eigenvectors, the controller gain matrix, and the function $\phi(\mathbf{Y}, W)$ as follows:

$$\Lambda = [-1.0184 \quad -2.0165 \quad -3.0148 \quad -3.9908]$$

$$\phi(\mathbf{Y}, W) = 8.0840$$

$$V_R = \begin{bmatrix} 1.9593 & 0.7860 & 1.3913 & -0.4749 \\ -1.9954 & -1.5850 & -4.1943 & 1.8951 \\ 0.8334 & -3.3813 & 2.1977 & 3.8898 \\ 0.9064 & -1.8470 & 0.7328 & 0.9723 \end{bmatrix}$$

$$K = \begin{bmatrix} 1.6569 & 3.2205 \\ 0.6663 & 1.8746 \\ 2.4237 & 1.8304 \\ -3.5491 & -6.1889 \end{bmatrix}^T$$

From the above, it can be seen that the performance function $\phi(\mathbf{Y}, W)$ (8.0840) is much larger than the maximum (4.0893) of the individual eigenvalue sensitivities $\phi_i(\mathbf{Y}, W)$ ($i = 1, 2, 3, 4$) which have met the design requirements. This indicates that the individual eigenvalue sensitivity functions are more accurate and better to describe the system insensitivity property to perturbations than the performance function $\phi(\mathbf{Y}, W)$. If the gain matrix elements were considered to be too large, a further reduction would be possible by sacrificing parameter insensitivity (and vice versa), i.e., the current level of one or more individual eigenvalue sensitivities, by appropriate adjustments to ε_i ($i = 1, 2, 3, 4$). In this way, we can design the controller to meet the specific needs.

Conclusions

A parameter-insensitive design method using eigenstructure assignment and the method of inequalities has been developed for multivariable control systems. Based on the eigenstructure assignment principle, the control problem is formulated as that of finding a controller to satisfy a set of performance criteria described by a set of inequalities, which includes requirements of individual eigenvalue sensitivities and feedback gains. The performance criteria describe the practical control problem more accurately and less conservatively than would be possible using scalar performance criterion approaches. A numerical solution has been presented using the method of inequalities and illustrated in the design of an aircraft system controller.

References

[1] Moore, B. C., "On the Flexibility Offered by State Feedback in Multivariable Systems Beyond Closed-Loop Eigenvalue Assignment," *IEEE Transactions on Automatic Control*, Vol. 21, No. 6, 1976, pp. 689–692.

[2] Burrows, S. P., and Patton, R. J., "A Comparison of Some Robust Eigenvalue Assignment Techniques," *Optimal Control Application and Methods*, Vol. 11, No. 4, 1990, pp. 355–362.

[3] Burrows, S. P., and Patton, R. J., "Design of a Low Sensitivity, Minimum Norm and Structurally Constrained Control Law Using Eigenstructure Assignment," *Optimal Control Application and Methods*, Vol. 12, No. 3, 1991, pp. 131–140.

[4] Kautsky, J., Nichols, N. K., and Dooren, P. Van., "Robust Pole Assignment in Linear State Feedback," *International Journal of Control*, Vol. 41, No. 5, 1985, pp. 1129–1155.

[5] Roppenecker, G., "On Parametric State Feedback Design," *International Journal of Control*, Vol. 43, No. 3, 1986, pp. 793–804.

[6] Zakian, V., and Al-Naib, U., "Design of Dynamical and Control Systems by the Method of Inequalities," *Proceedings of the Institute of Electrical Engineering*, Vol. 120, No. 11, 1973, pp. 1421–1427.

[7] Liu, G. P., and Patton, R. J., "Parametric State Feedback Design of Multivariable Control Systems Using Eigenstructure Assignment," *Proc. of the 32nd IEEE Conference on Decision and Control*, TX, Dec. 1993, 835-836.

[8] Liu, G. P., "Theory and Design of Critical Control Systems," Ph.D. Thesis, Control Systems Centre, Univ. of Manchester Inst. of Science and Technology, Manchester, England, UK, Feb. 1992.

[9] Whidborne, J. F., and Liu, G. P., *Critical Control Systems: Theory, Design and Applications*, Research Studies Press Limited, UK, 1993.

[10] Andry, A. E., Chung, J. C., and Shapiro, E. Y., "Modalized Observers," *IEEE Transactions on Automatic Control*, Vol. 7, No. 7, 1984, pp. 669–672.

Component Model Reduction Methodology for Articulated Multiflexible Body Structures

Allan Y. Lee* and Walter S. Tsuha†
Jet Propulsion Laboratory,
California Institute of Technology,
Pasadena, California 91109

Introduction

TO simulate and analyze the dynamical motion of an articulated, multiflexible body structure, we can use multibody simulation packages such as Dynamic Interaction Simulation Controls and Structure (DISCOS).[1] To this end, we must supply appropriate reduced-order models for all the flexible components involved. For complex systems, such as the Galileo spacecraft, practical considerations (e.g., simulation time) impose limits on the number of modes that each flexible body can retain in a given simulation. Hence, reduced-order models of the system's flexible components are needed.

Model reduction methodologies typically are used to reduce a large system model to one that is small enough to facilitate analysis and control design, yet "rich" enough that it retains the salient features of the original system model. Although the literature on model reduction is vast (see, e.g., Ref. 2), works that address the model reduction needs of articulated, multiflexible body structures are sparse, and have appeared only recently.[3–6] Here, what is needed is a way of generating reduced-order component models, which, when reassembled, produce a reduced-order system model that retains the important input-to-output mapping of the original system. The enhanced projection and assembly[6] technique is one way to perform this task.

In this method, a composite mode set, consisting of important system modes from all system configurations of interest, not just from one particular system configuration, is first selected. It is then augmented with static correction modes before being projected onto the component models to generate reduced-order component models. To generate the composite mode set, eigenvalue problems for the full-order system models, at all configurations of interest, must be solved repetitively. This is a drawback of the method because solving large eigenvalue problems can be costly. To overcome this difficulty, a two-stage model reduction methodology, combining the classic component mode synthesis method and the enhanced projection and assembly method, is proposed in this research (Fig. 1).

Received Feb. 5, 1993; revision Oct. 8, 1993; accepted for publication Oct. 15, 1993.

*Mail Stop 198-326, Member of Technical Staff, Guidance and Control Section, 4800 Oak Grove Drive.

†Mail Stop 157-410, Member of Technical Staff, Communications Systems Research Section, 4800 Oak Grove Drive.

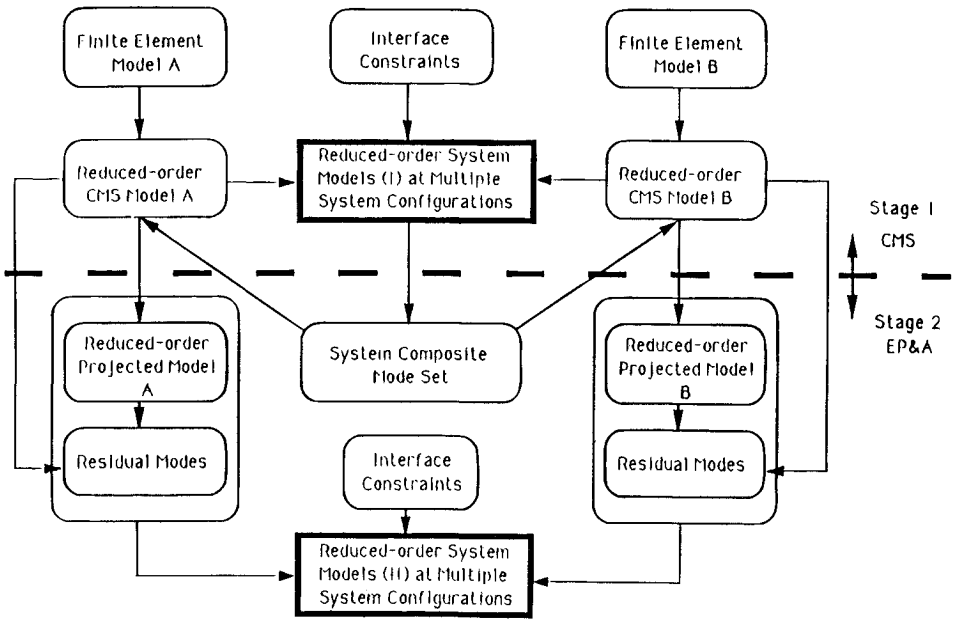

Fig. 1 A two-stage component modes projection and assembly model reduction methodology (COMPARE).

Component Mode Synthesis Method Revisited

The Component Mode Synthesis (CMS)[7] method is commonly used to analyze linear, high-order structural dynamics problems. To use this method, the structure is first subdivided into a number of components (or substructures). Component mode sets, such as the MacNeal-Rubin (M-R) and Craig-Bampton (C-B) mode sets are then used to reduce the model orders of these substructures. To review the constructions of the C-B and M-R mode sets, consider a two-flexible body structure. The undamped motion of each component of the structure can be described by

$$M_{nn}\ddot{x}_n + K_{nn}x_n = f_n \tag{1}$$

where x_n is an $n \times 1$ displacement vector, and M_{nn} and K_{nn} are the $n \times n$ mass and stiffness matrices of the component, respectively. Note that the matrix dimensions are indicated by the matrix subscripts. The $n \times 1$ force vector acting on the component is denoted by f_n.

To generate the C-B or M-R mode set, Eq. (1) is partitioned as follows:

$$\begin{bmatrix} M_{ii} & M_{ij} \\ M_{ji} & M_{jj} \end{bmatrix}\begin{bmatrix} \ddot{x}_i \\ \ddot{x}_j \end{bmatrix} + \begin{bmatrix} K_{ii} & K_{ij} \\ K_{ji} & K_{jj} \end{bmatrix}\begin{bmatrix} x_i \\ x_j \end{bmatrix} = \begin{bmatrix} f_i \\ 0 \end{bmatrix} \tag{2}$$

where x_i and x_j represent the interface and interior coordinates, respectively. The C-B mode set is generated by augmenting a low-frequency subset of the fixed interface I/F, normal modes with a set of constraint modes. The first k fixed I/F normal modes Φ_{jk}, and the ordered eigenvalue matrix Λ_{kk} are related by the following relation:

$$-M_{jj}\Phi_{jk}\Lambda_{kk} + K_{jj}\Phi_{jk} = 0 \tag{3}$$

There are ways to decide on the number of modes to be kept in Φ_{jk}. One way is to keep all modes with frequencies below twice a characteristic frequency of the system (e.g., control bandwidth).[3] These normal modes are then augmented with i constraint modes that are static-shape functions that result when a unit displacement is imposed on one coordinate of the i-set, while the remaining i-set coordinates are held fixed. The interior displacements for these constraint modes are given by: $\Psi_{ji} = -K_{jj}^{-1}K_{ji}$. The C-B mode set is

$$\begin{bmatrix} O_{ik} & I_{ii} \\ \Phi_{jk} & \Psi_{ji} \end{bmatrix} \tag{4}$$

which is then used to reduce the full-order component model.

Like the C-B mode set, the M-R mode set is generated by augmenting a low-frequency subset of the free I/F normal modes with a set of static force response functions, termed residual modes. The first k free I/F normal modes Φ_{nk}, and the ordered eigenvalue matrix Λ_{kk}, are related by the following relation:

$$-M_{nn}\Phi_{nk}\Lambda_{kk} + K_{nn}\Phi_{nk} = 0 \tag{5}$$

The kept eigenvector matrix Φ_{nk}, which has been normalized with respect to M_{nn}, may be partitioned into its rigid-body and flexible parts: $(\Phi_{nr}\Phi_{nf})$. Let Λ_{ff} be the eigenvalue matrix associated with the kept flexible modes. Then, the residual modes are determined by: $\Psi_{na} = (P_{nn}^T S_{nn} P_{nn} - \Phi_{nf}\Lambda_{ff}^{-1}\Phi_{nf}^T)F_{na}$, where $P_{nn} = I_{nn} - M_{nn}\Phi_{nr}\Phi_{nr}^T$, S_{nn} is a "pseudo" flexibility matrix (Refs. 6,7), and F_{na} is given by $(0_{ar}, I_{aa}, 0_{aw})^T$, where the identity matrix I_{aa} is associated with the a interface coordinates, and $r + a + w = n$. The M-R mode set is $(\Phi_{nk}\ \Psi_{na})$, which is then used to generate a reduced-order model for the component.

Component Modes Projection and Assembly Model Reduction Methodology (COMPARE)

Once the CMS-based, reduced-order component models are generated, they are assembled using the interface compatibility conditions to produce reduced-order system models. However, we note that these CMS-based, reduced-order system models were obtained without using any knowledge of system-level input–output information. This drawback is remedied in the

second stage using the Enhanced Projection and Assembly Model Reduction (EP&A) methodology,[6] which does make use of the system's input-to-output information.

Consider a system with two flexible components. The undamped motion of component A, as described by either its M-R or C-B mode set, is given by

$$I_{pp}\ddot{\eta}_p^A + \Lambda_{pp}^A \eta_p^A = G_{pa}^A u_a^A, \qquad y_b^A = H_{bp}^A \eta_p^A \tag{6}$$

In Eq. (6), η_p^A and Λ_{pp}^A are the generalized coordinates and the diagonal stiffness matrix of component A, respectively. The matrix G_{pa}^A is a control distribution matrix, and u_a^A is a control vector. Similarly, the matrix H_{bp}^A is an output distribution matrix, and y_b^A is an output vector. Similar equations can also be written for component B.

The system equations of motion at a particular articulation angle α may be constructed using Eq. (6) and enforcing displacement compatibility at the component I/F. To this end, let $P(\alpha) = [P_{pe}^{AT}(\alpha), P_{qe}^{BT}(\alpha)]^T$ be a full-rank matrix mapping a minimal system state η_e into

$$\begin{bmatrix} \eta_p^A \\ \eta_q^B \end{bmatrix} = \begin{bmatrix} P_{pe}^A(\alpha) \\ P_{qe}^B(\alpha) \end{bmatrix} [\eta_e] \tag{7}$$

where η_e is the $e \times 1$ reduced-order system coordinate, $e = p + q - i$. One way to generate the $P(\alpha)$ matrix was described in Ref. 6. For ease of notation, the dependencies of the matrices P_{pe}^A, etc. on α are dropped in the sequel. Substituting $\eta_p^A = P_{pe}^A \eta_e$ and $\eta_q^B = P_{qe}^B \eta_e$ into Eq. (6) and the corresponding equations for component B, premultiplying the resultant equations by P_{pe}^{AT} and P_{qe}^{BT}, respectively, and summing the resultant equations gives

$$M_{ee}\ddot{\eta}_e + K_{ee}\eta_e = G_{ea}u_a, \qquad y_s = H_{se}\eta_e \tag{8}$$

where $M_{ee} = P_{pe}^{AT}P_{pe}^A + P_{qe}^{BT}P_{qe}^B$, and similar expressions can be written for K_{ee}, G_{ea}, and H_{se}. Let $(\Phi_{ee}, \Lambda_{ee}) = \text{eig}(K_{ee}, M_{ee})$, the "modal" equivalence of Eq. (8) is

$$I_{ee}\ddot{\phi}_e + \Lambda_{ee}\phi_e = \Phi_{ee}^T G_{ea} u_a, \qquad y_s = H_{se}\Phi_{ee}\phi_e \tag{9}$$

where ϕ_e is the modal coordinate; i.e., $\eta_e = \Phi_{ee}\phi_e$, $\Phi_{ee}^T M_{ee}\Phi_{ee} = I_{ee}$, and $\Lambda_{ee} = \Phi_{ee}^T K_{ee}\Phi_{ee}$ contains the undamped, reduced-order system eigenvalues along its diagonal. Equation (9) represents the reduced-order system model obtained from the first stage of the COMPARE methodology.

The EP&A method is used in the second stage of COMPARE. With this method, only k of the system's e modes are kept, while the remaining t $(= e - k)$ modes are removed. Ways to select important system's modes that make significant contribution to the system's input-to-output mapping have been suggested.[2] In our study, we use the modal influence coefficient (MIC) defined in Refs. 5 and 6 to rank the relative importance of the modes.

For articulated, multiflexible bodies structures, a set of important system modes at a particular system configuration might be different from that at another configuration. Hence, the projection mode set selected should be a composite mode set, consisting of important modes from all system configurations of interest, not just from one particular configuration. Also, the system might contain multiple input–output pairs. In this case, important system modes for all input–output pairs, and at all system configurations of interest must be included in the composite modes set. If the resultant mode set becomes unacceptably large, we can consider the use of several composite mode sets. In this case, the reduced-order system model generated using a particular composite mode set can only be used for the input–output pairs for which the mode set was generated.

With this understanding, we let $\eta_e = \Phi_{ee}\phi_e = (\Phi_{ek}\Phi_{et})(\phi_k^T\phi_t^T)^T \doteq \Phi_{ek}\phi_k$. Here, ϕ_k and ϕ_t are the "kept" and "truncated" coordinates, respectively. If the composite mode set Φ_{ek} is projected onto the CMS-based component models, and the resultant reduced-order component models combined using the I/F compatibility conditions, we produce a reduced-order system model.

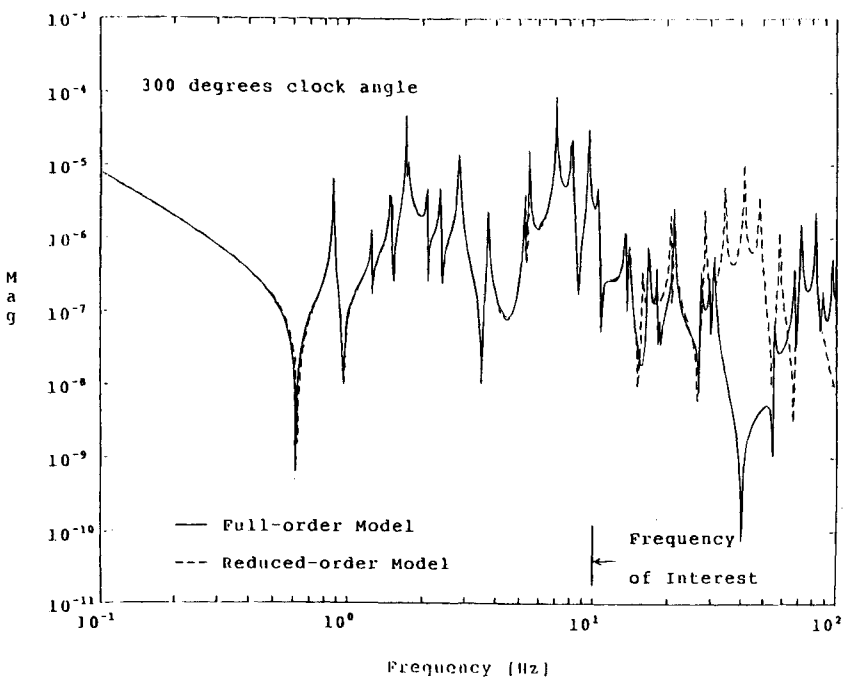

Fig. 2 Bode plot comparison of full and reduced-order models (300 deg).

It has been proved that the modes retained in Φ_{ek} are captured exactly by the generated reduced-order system model.[4] However, the static gain of the resultant reduced-order system modelis not the same as that given by Eq. (9) (Ref. 6).

Two approaches were introduced by Lee and Tsuha[6] to preserve the static gain of the system. In the system level augmentation approach, we first note that the system modal Eq. (9) can be decomposed into its kept and truncated parts:

$$I_{kk}\ddot{\phi}_k + \Lambda_{kk}\phi_k = \Phi_{ek}^T G_{ea} u_a \tag{10}$$

$$I_{tt}\ddot{\phi}_t + \Lambda_{tt}\phi_t = \Phi_{et}^T G_{ea} u_a \tag{11}$$

The static effects of the truncated mode set Φ_{et} can be approximated by a smaller but statically equivalent mode set Φ_{ea}. It can be shown that the static gain of the reduced-order system model obtained using a mode set $(\Phi_{ek}\Phi_{ea}) = (\Phi_{ek}\Phi_{et}\ \Lambda_{tt}^{-1}\Phi_{et}^T G_{ea})$ is identical to that of the original system [cf. Eq. (9)].[6] This augmented mode set can now be projected onto the CMS-based component models: $\eta_p^A \doteq P_{pe}^A(\Phi_{ek}\Phi_{ea})\ \phi_v^A \triangleq \Psi_{pv}^A\ \phi_v^A$. Here, $v = k + a$, and ϕ_v^A denotes a reduced set of generalized coordinates of component A. The substitution of the last relation into Eq. (6) produces the "constrained" equations of motion for component A:

$$\Psi_{pv}^{AT}\Psi_{pv}^A\ddot{\phi}_v^A + \Psi_{pv}^{AT}\Lambda_{pp}^A\Psi_{pv}^A\phi_v^A = \Psi_{pv}^{AT}G_{pa}^A u_a \tag{12}$$

A similar equation can also be written for component B. Again, we use the I/F conditions to "combine" these component models to generate a new reduced-order system model. It was proved in Lee and Tsuha[6] that all the "kept" modes as well as the static gain of the full-order system are captured exactly by the reduced-order system model. The details of the second approach, the component level augmentation approach, are given in Ref. 6. The effectiveness of the proposed COMPARE methodology is demonstrated below.

Applying COMPARE on a High-Order Galileo Model

The dual-spin Galileo spacecraft has three components: rotor, stator, and scan platform.[6] The rotor is the largest and most flexible component, represented with 243 DOF. The smaller and more rigid stator is represented with 57 DOF. The rotor may be rotated with respect to the stator. Lastly, the scan platform is idealized as rigid with 6 DOF.

For the purpose of controller design, a low-order system model, accurate at all configurations of interest and over a frequency range of 0–10 Hz is needed. To this end, we apply the M-R version of the COMPARE methodology on the Galileo model. Following standard procedures, free interface normal modes of both the rotor and stator are first determined, and then truncated at twice the frequency of interest, 20 Hz. These truncated normal mode sets are then each augmented with residual modes to generate the needed M-R mode sets for both the stator and rotor. The resultant CMS-based stator and rotor models have 24 (with 8 rigid-body) and 80 (with 6 rigid-body) modes, respectively.

Next, the CMS-based rotor and stator models are combined using the I/F compatibility condition to generate system models at six system configurations of interest. The resultant system model has 98 (with 8 rigid-body) modes. Using the MIC as a selection criterion, important system modes are selected at system configurations with clock angles of 0, 60, 120, 180, 240, and 300 deg. A composite mode set that encompasses important modes at all configurations is then determined. It contains 8 rigid-body and 21 flexible modes.

For the Galileo example, we augment the composite mode set with two residual modes, one for an input torque about the spin-axis on the rotor side of the rotor/stator interface, and a second one on the stator side. The augmented composite set is then projected onto the flexible components, and a singular value decomposition was used to remove linearly dependent modes in the projected stator mode set. The resultant reduced-order models of the rotor and stator have 29 (with 6 rigid-body) and 21 (with 8 rigid-body) modes, respectively. The reduced-order system model has 44 (with 8 rigid-body) modes.

A comparison of Bode plots of full and reduced-order models, at a clock angle of 300 deg, is given in Fig. 2. Actuation was done by a spin motor located at the rotor-stator I/F, and sensing was done by a gyroscope located on the scan platform. From that figure, we observe that the selected modes of the full-order model have been exactly captured by their reduced-order counterparts. The frequency responses of the full-order models at other system configurations (0, 60, 120, 180, and 240 deg), given in Ref. 8, have also been closely captured by their reduced-order counterparts.

Concluding Remarks

A two-stage model reduction methodology, combining the classic component mode synthesis method and the newly developed enhanced projection and assembly method is proposed in this research. The merit of our methodology is that system models (at various system configurations) assembled using component models obtained via the component mode synthesis techniques are smaller in size than the full-order system models. Hence, this approach alleviates the need to solve large-order eigenvalue problems repetitively.

By comparing the Bode plots of the full and reduced-order system models at multiple configurations of interest, the effectiveness of our methodology has been successfully demonstrated on a high-order model of the Galileo spacecraft. However, because the reduced-order models will ultimately be used for nonlinear simulations, the final comparison should be done in the time domain. Judging by the results obtained, we expect to obtain good time domain results with this approach if the articulation rate is "sufficiently" slow. This judgment, however, must be confirmed and quantified via careful nonlinear time simulations. The effects that increasing slew rate has on the effectiveness of the methodology is an interesting topic to pursue in the future.

Acknowledgments

The research described in this paper was conducted at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. The authors thank their colleagues, including F. Hadaegh, M. Lou, G. Man, J. Spanos, and M. Wette, for many helpful discussions and valuable suggestions.

References

[1] Bodley, C. S., Devers, A. D., Park, A. C., and Frisch, H. P., "A Digital Computer Program for the Dynamic Interaction Simulation of Controls and Structure (DISCOS)," NASA TP 1219, Vols. I and II, NASA Center for Aerospace Information, Baltimore, MD, May 1978.

[2] Moore, B. C., "Principal Component Analysis in Linear Systems: Controllability, Observability, and Model Reduction," *IEEE Transactions on Automatic Control*, Vol. AC-26, No. 2, 1981, pp. 17–32.

[3] Spanos, J. T., and Tsuha, W., "Selection of Component Modes for the Simulation of Flexible Multibody Spacecraft," *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 2, 1991, pp. 278–286.

[4] Bernard, D., "Projection and Assembly Method for Multibody Component Model Reduction," *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 5, 1990, pp. 905–912.

[5] Eke, F. O., and Man, G. K., "Model Reduction in the Simulation of Interconnected Flexible Bodies," *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference* (Kalispell, MT), Aug. 1987 (AAS Paper 87–455).

[6]Lee, A. Y., and Tsuha, W. S., "An Enhanced Projection and Assembly Model Reduction Methodology," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 1, 1994, pp. 69–75.

[7]Craig, R. R., Jr., *Structural Dynamics: An Introduction to Computer Methods*, John Wiley & Sons, New York, 1981.

[8]Lee, A. Y., and Tsuha, W. S., "A Component Modes Projection and Assembly Model Reduction Methodology for Articulated Flexible Structures," JPL Internal Document D-9364, Jet Propulsion Laboratory, California Institute of Technology, Dec. 1991.

Synthesis of Minimum-Time Feedback Laws for Dynamic Systems Using Neural Networks

Allan Y. Lee* and Padhraic Smyth†
Jet Propulsion Laboratory,
Pasadena, California 91109

Introduction

GUIDANCE and control dynamical optimization problems in aeronautics, astronautics, and other fields have been routinely formulated and solved over the past 30–35 years. Of necessity, these complex optimization problems are often numerically solved. Past research efforts have produced several software packages (see, e.g., Refs. 1 and 2) which can be used to determine the open-loop solutions of dynamical optimization problems.

By an open-loop solution we mean that the control at any time instant is not explicitly determined by the states of the system at that instant. It is well-known that a system with an open-loop controller can be sensitive to noise and external disturbances. In contrast, feedback control, in which the control is a function of the instantaneous state of the system is generally robust with respect to such disturbances. Unfortunately, only rarely is it feasible to determine the feedback laws for nonlinear systems of any practical significance.

A novel approach of using neural networks to synthesize nonlinear feedback laws is explored in the present research. The details and efficacy of this approach will be demonstrated using a minimum-time orbit injection problem.

Open-Loop Solutions of Dynamical Optimization Problems

Consider the following dynamical optimization problem:

$$\min_{\vartheta(\tau),\,\pi} I = \phi_1(x_1, \pi) + \int_0^1 L(x, \vartheta, \pi, \tau)\, d\tau \tag{1}$$

$$\dot{x} = f(x, \vartheta, \pi, \tau), \qquad x(0) = \text{given} \tag{2}$$

$$\psi(x_1, \pi) = 0 \tag{3}$$

Here, $x(n \times 1)$, $\vartheta(m \times 1)$, and $\pi(p \times 1)$ are the state, control, and unknown parameter vectors, respectively. The independent variable (usually time) is τ. The scalar function L denotes a running cost while $f(n \times 1)$ is a vector function. The parameter vector π and the control vector $\vartheta(\tau)$ are to be optimally determined to minimize the cost functional I subject to the initial condition given in Eq. (2) and the terminal constraint $\psi(q \times 1)$ given in Eq. (3). The subscript 1 in Eqs. (1) and (3) denotes the condition of the variable concerned at the end time T. If T is free, we can define $\tau = t/T$ (which now varies from 0 to 1), and use it to transform the problem to one with a fixed end time.[2] The unknown end time T is now part of π.

The optimality conditions of this optimization problem are well-known, and are given in, for example, Refs. 2 and 3. These conditions are all that are needed to solve the dynamical optimization problem. However, unless the functions L and f are quite simple, analytical results are difficult to obtain and the optimization problem must be solved numerically, using, for example, the combined parameter and function optimization algorithm (CPFA).[2]

The solution obtained is only open-loop. A system with an open-loop controller can be sensitive to noise and external disturbances. In contrast, feedback control, in which the control is a function of both the state and time, $\vartheta = \vartheta(x, \tau)$, is generally robust with respect to such disturbances. For stationary systems where f and L are not explicit functions of time, and the end time is free, we have $\vartheta = \vartheta(x)$.[3] Unfortunately, only rarely is it feasible to analytically determine these feedback laws for nonlinear systems of any practical significance.

To overcome this problem, Breakwell et al.[4] obtained a linear neighboring optimal feedback law (perturbation guidance law) via the minimization of the second variation of the performance index. The ranges over which these perturbation guidance laws are useful will depend on the system and might be small for nonlinear systems. In Ref. 5, open-loop optimal control data are used to train a feedforward multilayer neural network. The network is then used effectively to control a two-link robotic arm between its end conditions in minimum time. In this study, a similar approach is used to synthesize a feedback controller for a minimum-time orbit injection problem.

Artificial Neural Networks as Nonlinear Mapping Approximators

An artificial neural network (henceforth referred to as neural network) is a system which is composed of many simple and similar nonlinear processing elements (neurons). A typical neural network consists of an input layer, one or more hidden layers, and an output layer. The input layer works as a buffer and may include some preprocessing of the input data. The outputs of the hidden nodes are calculated from: $z_j^2 = F(\sum_{i=1}^{N_1} w_{ij}^1 z_i^1 + \theta_j^1)$, for $j = 1, \ldots, N_2$. Here, z_i^1 $(N_1 \times 1)$ are inputs to the network, z_j^2 $(N_2 \times 1)$ are outputs of the hidden layer, w_{ij}^1 $(N_1 \times N_2)$ are weights between the network inputs and the hidden layer outputs, and $\theta_j^1(N_2 \times 1)$ are internal offsets of the hidden layer. The function F is a monotone continuous function, usually a sigmoid function, or a hyperbolic tangent function. The sigmoid function $F(s) = 1/[1 + e^{-s}]$ is used here. The outputs of the network are calculated similarly.

To perform a desired mapping, the network has to adapt its weights through a training process. During this process, the network is presented with an input vector z_i^1, which causes its output to be z_k^3 (z_i^1). We then adjust the weights to cause the output to be something closer to the $d_k(z_i^1)$, the desired output. Back propagation algorithms are usually used to train multilayered neural networks. One version of these algorithms is described in Ref. 6.

In the present research, we train the network to approximate the optimal, nonlinear states-to-controls mapping. First, the open-loop solutions of an optimization problem are computed for multiple sets of end conditions at and around the nominal condition. Next, the computed time histories of the states and controls are then used to train a neural network with a prese-

Received July 10, 1993; revision received Sept. 8, 1993; accepted for publication Sept. 10, 1993.

*Member of Technical Staff, Guidance and Control Section, 4800 Oak Grove Drive.

†Member of Technical Staff, Communications Systems Research Section, 4800 Oak Grove Drive.