Nonlinear Model Simplification in Flight Control System Design

A.A. Desrochers* and R.Y. Al-Jaar†
Rensselaer Polytechnic Institute, Troy, New York

An algorithm is presented for obtaining a reduced-order or simplified nonlinear model of a nonlinear plant. Controllers are then designed for this nonlinear model and used to control the original plant. The method is illustrated on an analytical model of the F-8 aircraft. In this example, a linear controller can be designed using the simplified model. This controller's performance is indistinguishable from nonlinear optimal controllers designed for the original plant.

Introduction

REDUCED-order modeling for linear systems has received much attention in recent years,[1] and its success has enabled the design of reduced-order observers and controllers which can effectively control the original system.[2] A similar problem exists for nonlinear systems, for which it is desirable and usually only practical to design controllers based on reduced-order or simplified models of the full nonlinear system. Until now, this has been a relatively difficult task, since obtaining a simplified model is almost always a heuristic, or trial and error, process. In this paper, a systematic approach is developed for quickly obtaining the simplified model.

The need to take into account the nonlinearities of a system has become more and more important as the demands for better performance have increased. This has been the case with the high performance F-8 aircraft, for which Garrard and Jordan[3] point out that ignoring the nonlinearities can lead to abnormal, if not hazardous, flying conditions. Their study proposed the design of a closed-loop, third-order, nonlinear controller. It was found that such a controller did indeed increase the magnitude of the angle of attack for which the aircraft could recover from stall. However, the authors note that obtaining such a controller by using the Hamilton-Jacobi-Bellman principle is quite tedious and laborious. It is our objective to show that a linear controller designed from a simplified nonlinear model can be found with performance is indistinguishable from the third-order nonlinear controller derived by Garrard and Jordan.[3] Thus, another purpose of this paper is the demonstration of the effectiveness of reduced-order controllers operating in a nonlinear environment.

Problem Statement

The problem is to reduce the complexity and/or dimensionality of a nonlinear system modeled in a Euclidean space S of dimension m:

$$z(k+1)=A\Phi[z(k),u(k)] \tag{1}$$

with $z(k)$ representing a $(p\times 1)$ state vector, $\Phi[z(k),u(k)]$ an $(m\times 1)$ vector of functions, $u(k)$ the input, and A a $(p\times m)$ matrix of constant coefficients. Therefore, the system to be reduced can be written as:

$$\begin{bmatrix} z_1(k+1)\\ z_2(k+1)\\ \vdots\\ z_p(k+1)\end{bmatrix}=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m}\\ a_{21} & a_{22} & \cdots & a_{2m}\\ \vdots & \vdots & & \vdots\\ a_{p1} & a_{p2} & \cdots & a_{pm}\end{bmatrix}\begin{bmatrix} \phi_1[z(k),u(k)]\\ \phi_2[z(k),u(k)]\\ \vdots\\ \phi_m[z(k),u(k)]\end{bmatrix} \tag{2}$$

and the $\phi_i[z(k),u(k)]$ form a set of linearly independent, nonlinear continuous functions spanning the m-space S.

The objective here is to find a system that approximates the original system, Eq. (1), while minimizing the equation error between the original full-order model and the reduced-order model in a least-squares error sense. Thus, it is desired to find a system described by

$$z_R(k+1)=A_R\Phi_R[z_R(k),u(k)] \tag{3}$$

To simplify notation we let $\phi_i(k)$ represent $\phi_i[z(k),u(k)]$ and define $\Phi_R(k)$ similarly. Equation (3) can then be expressed as

$$\begin{bmatrix} z_{R1}(k+1)\\ z_{R2}(k+1)\\ \vdots\\ z_{Rn'}(k+1)\end{bmatrix}=\begin{bmatrix} a_{R11} & a_{R12} & \cdots & a_{R1n}\\ a_{R21} & a_{R22} & \cdots & a_{R2n}\\ \vdots & \vdots & & \vdots\\ a_{Rn'1} & a_{Rn'2} & \cdots & a_{Rn'n}\end{bmatrix}\begin{bmatrix} \phi_{R1}(k)\\ \phi_{R2}(k)\\ \vdots\\ \phi_{Rn}(k)\end{bmatrix} \tag{4}$$

with $n<m$ and $n'\le p$.

This approach to model reduction can be envisioned as a least-squares error projection of the m-dimensional space S, spanned by $[\phi_1(k),\ldots,\phi_m(k)]$, onto the space $\bar{S}$, spanned by $[\phi_{R1}(k),\ldots,\phi_{Rn}(k)]$.

It is worthwhile mentioning that if model simplification is desired, then $n'=p$ and the states of the original system are the same as those of the reduced system; however, $\Phi_R(k)$ will contain only n of the elements of $\Phi(k)$. On the other hand, when $n'<p$ a model reduction occurs.

Now, define the set B that spans the space S as,

$$B\triangleq[\phi_i(k);\quad i=1,2,\ldots,m] \tag{5}$$

and let the subset $\bar{B}$ span the space $\bar{S}$ and be defined by

$$\bar{B}\triangleq\{\phi_{P_1}(k),\ldots,\phi_{p_n}(k)\};\qquad n<m$$

where $1\le p_1,p_2\ldots p_n\le m$ are distinct integer subscripts. The

Submitted July 12, 1983; revision received Nov. 14, 1983.

*Assistant Professor, Electrical, Computer and Systems Engineering Department.

†Graduate Student, Electrical, Computer and Systems Engineering Department.

simplification algorithm is required to find the optimal subset $\bar{B}$.

Nonlinear Model Simplification Algorithm

The projection of S onto $\bar{S}$ involves the consideration of many models. In fact, there are $m!/n!(m-n)!$ models with n terms. One approach is to design and test all possible models using, say, a least-squares approach. This exhaustive approach is not very practical.

With the aim of avoiding this brute-force method, Desrochers proposed[4,5] to represent the model-reduction problem as a tree-structured problem, where each node in the tree represents a possible model structure. Let the admissible set of models be those which are formed from linear combinations of any and all of the elements of B. Furthermore, let each model structure be represented as a node in a tree, as shown in Fig. 1 for the case $m=4$. To avoid designing and testing all possible models, a cost is assigned, based on a least-squares error criterion, as a measure of the reduction in error that results when a term is added to a particular model. Basically, this assigns a weight to the relative importance of each component of the model. These costs appear on the branches of the tree and are derived such that the optimal n term model is the maximum cost path from the root to a node at depth n. Furthermore, the sum of the costs from the root to any node is related to the squared error for that model, so that the tree generation ceases when some acceptable error has been achieved. This avoids an arbitrary choice on n and also avoids generating costs for all possible model structures. Thus, in practice, the tree does not really include a node for every possible structure.

For a tree with the above properties, the maximum cost path search becomes a dynamic programming problem.[4] If model simplification (reduction) has anything to offer to a specific problem, then $n \ll m$ and the search is kept simple.

The above approach insures that no matrix inversion is done since no model parameters have to be calculated (except those of the final optimal n term model). This is where computation time is saved. Similarly, input-output data sequences need only be of length n.

In addition, knowing the error of the approximant at each level of the tree avoids the pursuit of many useless models. In fact, experience has shown that the optimal n-term model (in achieving some acceptable error) can be found in about the same time it takes to design and evaluate one of the many n-term models using traditional least-squares methods.

The derivation of the algorithm begins by expressing the system in Eq. (1) as

$$z(k+1)=\sum_{i=1}^{m} A_i\phi_i(k) \tag{6}$$

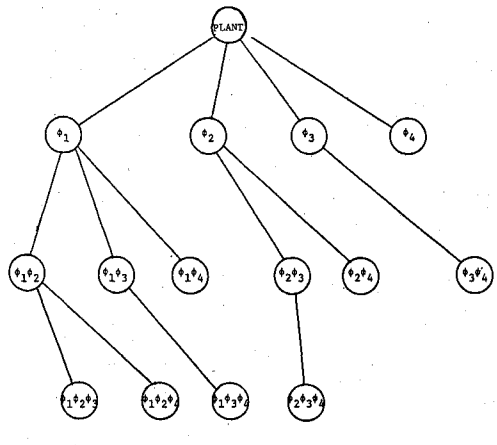

Fig. 1 Possible model structures.

where A_i represents the ith column of the matrix A, and $\phi_i(k)$, $i=1,2,\ldots,m$ are the elements of $\Phi(k)$. The goal is to develop a tree-structured algorithm which retains the most dominant nonlinearities in Eq. (6). If no $\phi_i(k)$ is a function of more than one state, then eliminating certain $\phi_i(k)$ will eliminate certain states, and a model reduction will occur. Otherwise, eliminating terms in Eq. (6) will lead to a model simplification.

To derive the criterion necessary to select the optimal model let the original system be approximated by one function, say $\phi_i(k)$. Then define the equation error as

$$e(k+1)=z(k+1)-\bar{A}_i\phi_i(k) \tag{7}$$

where A_i does not necessarily equal $\bar{A}_i$ and define

$$\epsilon_i=\min_{\bar{A}_i}\sum_{k=1}^{N} e^T(k+1)e(k+1) \tag{8}$$

For ease of presentation let $n=2$ and $\bar{A}_i=[\bar{a}_{1i}\bar{a}_{2i}]^T$ with summations understood to be from 1 to N. To simplify notation, we define an inner product by

$$\langle\phi_a(k),\phi_b(k)\rangle \triangleq \sum\phi_a(k)\phi_b(k)$$

Then Eq. (8) becomes

$$\epsilon_i=\min_{\bar{A}_i}\sum\left[\{z_1(k+1)-\bar{a}_{1i}\phi_i(k)\}^2 + \{z_2(k+1)-\bar{a}_{2i}\phi_i(k)\}^2\right] \tag{9}$$

with $i=1,2,\ldots,m$. Differentiating Eq. (9) with respect to $\bar{a}_{1i}$ and $\bar{a}_{2i}$, setting these derivatives equal to zero, then solving for $\bar{a}_{1i}$ and $\bar{a}_{2i}$ and substituting back into Eq. (9) yields

$$\epsilon_i=\sum\left[\left\{z_1(k+1)-\frac{\langle z_1(k+1),\phi_i(k)\rangle}{\langle\phi_i(k),\phi_i(k)\rangle}\phi_i(k)\right\}^2 + \left\{z_2(k+1)-\frac{\langle z_2(k+1),\phi_i(k)\rangle}{\langle\phi_i(k),\phi_i(k)\rangle}\phi_i(k)\right\}^2\right] \tag{10}$$

This represents the error when the original system is approximated by one function, $\phi_i(k)$.

Similarly, let the error that results when approximating the original system by the jth function, $\phi_j(k)$, $j\neq i$, be ϵ_j. If we assume that the ith function represents the optimal one-term model, it is then required that

$$\epsilon^* \triangleq \epsilon_i < \epsilon_j \tag{11}$$

After some manipulation, Eq. (11) becomes

$$\frac{(\langle z_1(k+1),\phi_i(k)\rangle)^2+(\langle z_2(k+1),\phi_i(k)\rangle)^2}{\langle\phi_i(k),\phi_i(k)\rangle} > \frac{(\langle z_1(k+1),\phi_j(k)\rangle)^2+(\langle z_2(k+1),\phi_j(k)\rangle)^2}{\langle\phi_j(k),\phi_j(k)\rangle} \tag{12}$$

which provides a criterion for selecting the optimal one-term model. Define ρ_i as

$$\rho_i \triangleq \left[\sum_{l=1}^{2}(\langle z_l(k+1),\phi_i(k)\rangle)^2\right]\Big/[\langle\phi_i(k),\phi_i(k)\rangle] \qquad i=1,2,\ldots,m \tag{13}$$

Then the optimal one-term model is the one that yields the

maximum value of ρ_i; call it ρ_i^*. It is also possible to find the error generated from representing the original system by this optimal one-term model. Call this ϵ_i^*, where the result is

$$\epsilon_i^* = \sum_{l=1}^{2} \{\langle z_l(k+1), z_l(k+1)\rangle\} - \rho_i^* \qquad i = 1, 2, \ldots, m \tag{14}$$

Note that the parameters $\bar{a}_{1i}$ and $\bar{a}_{2i}$ need not be identified since they do not enter into the evaluation of Eqs. (13) and (14). This is a major source of computational savings over the brute-force approach.

As for the selection of the second function in the optimal two-term model, the Gram-Schmidt orthogonalization process must be used to take into account the choice of $\phi_i(k)$. Therefore, the functions

$$\{\phi_j(k),\ j = 1, 2, \ldots, m;\ j \neq i\}$$

are orthogonalized with respect to $\phi_i(k)$. Call these orthogonalized functions $[\phi_j^{(2)}(k);\ j = 1, 2, \ldots, m,\ j \neq i]$. Then the second optimal term can be chosen using Eq. (13) along with orthogonality to obtain

$$\rho_j^* \triangleq \max_j \left[\rho_j^{(2)}\right] = \max_j \left\{ \frac{\sum_{l=1}^{2} \left(\langle z_l(k+1), \phi_j^{(2)}(k)\rangle\right)^2}{\left(\langle \phi_j^{(2)}(k), \phi_j^{(2)}(k)\rangle\right)^2} \right\} \tag{15}$$

where $j = 1, 2, \ldots, m$ but $j \neq i$.

Note that the form of Eq. (15) is the same as that of Eq. (13). Therefore, for all n-term models, Eq. (13) represents the desired optimality criterion by which the algorithm will select the optimal n-term model.

As for the error resulting from the selection of the second optimal term $\phi_j(k)$, it can be derived in the same way as Eq. (14). The resulting error is

$$\epsilon_{ij}^* = \sum_{l=1}^{2} \{\langle z_l(k+1), z_l(k+1)\rangle\} - \rho_i^* - \rho_j^* \tag{16}$$

$$\epsilon_{ij}^* = \epsilon_i^* - \rho_j^* \tag{17}$$

These are all of the concepts and details needed to arrive at the model simplification method for nonlinear dynamic systems. The actual algorithm is in the Appendix.

Flight Control Design Using Nonlinear Model Simplification

For our purposes, we will state only the equations representing the dynamics of the aircraft.[3] The nonlinear equations of motion are:

$$\begin{aligned}
\dot{x}_1 &= -0.877x_1 + x_3 - 0.215u - 0.088x_1x_3 \\
&\quad + 0.47x_1^2 - 0.019x_2^2 - x_1^2x_3 + 3.846x_1^3 + 0.28ux_1^2 \\
&\quad + 0.47u^2x_1 + 0.63u^3 \\
\dot{x}_2 &= x_3 \\
\dot{x}_3 &= -4.208x_1 - 0.396x_3 - 20.967u - 0.47x_1^2 - 3.564x_1^3 \\
&\quad + 6.265ux_1^2 + 46u^2x_1 + 61.4u^3
\end{aligned} \tag{18}$$

where x_1 is the angle of attack in radians, x_2 the pitch angle in radians, x_3 the pitch rate in radians per second, and u the control input.

Finding a linear control law from a simplified model of the aircraft dynamics is required. This controller is to be compared with the linear, second- and third-order nonlinear controllers derived by Garrard and Jordan.[3]

Using linear quadratic regulator theory, Garrard and Jordan computed the linear control law, based on the linearized version of the nonlinear model. The optimal control law for the linearized system was found to be

$$u_1 = -0.053x_1 + 0.5x_2 + 0.521x_3$$

with $Q = 0.25I$ (where I denotes the 3×3 identity matrix) and $R = 1$ for the state and control weighting matrices in the quadratic performance index.

The second- and third-order nonlinear controllers were also derived[3] and found to be

$$u_2 = u_1 + 0.04x_1^2 - 0.048x_1x_2$$

and

$$u_3 = u_2 + 0.374x_1^3 - 0.312x_1^2x_2$$

Their derivation was based on a result by Lee and Markus[6] where for a system of the form

$$\dot{x}(t) = Ax(t) + \phi(x) + bu(t)$$

with $\phi(x)$ an analytic vector function representing the system nonlinearities, the feedback control law is

$$u = -\tfrac{1}{2}R^{-1}b^T\frac{\partial V(x)}{\partial x}$$

where $V(x)$ satisfies the Hamilton-Jacobi equation

$$\frac{\partial V^T}{\partial x}Ax + \frac{\partial V^T}{\partial x}\phi - \frac{1}{4}\frac{\partial V^T}{\partial x}bR^{-1}b^T\frac{\partial V}{\partial x} + x^TQx = 0; \qquad V(0) = 0$$

Table 1 Model simplification results for the F-8 aircraft fighter

Model	Model basis	Squared error	Ratio of reduction in error
i			$(i-1)/(i)$
3	(x_1, x_2, u)	5.01	—
4	(x_1, x_3, u, x_1^3)	1.14×10^{-1}	44
5	$(x_1, x_3, u, x_1^2x_3, x_1^3)$	4.76×10^{-2}	2
6	$(x_1, x_3, u, x_1^3, ux_1^2, u^2x_1)$	1.16×10^{-2}	4
7	$(x_1, x_3, u, x_1^2, x_1^3, ux_1^2, u^2x_1)$	4.44×10^{-3}	3
8	$(x_1, x_3, u, x_1^2, x_1^2x_3, x_1^3, ux_1^2, u^2x_1)$	8.82×10^{-4}	5
9	$(x_1, x_3, u, x_1^2, x_1^2x_3, x_1^3, ux_1^2, u^2x_1, u^3)$	1.51×10^{-5}	58
10	$(x_1, x_2, u, x_1x_3, x_1^2, x_1^2x_3, x_1^3, ux_1^2, u^2x_1, u^3)$	2.74×10^{-6}	6

This equation usually cannot be solved analytically, and so a series solution is used which leads to a tedious algebraic determination of many coefficients: 10 for the second-order controller, 15 for the third-order controller. Even for the system at hand the practicality of this method is questionable.[3] In the following sections, it will be demonstrated that utilizing the simplified nonlinear model is a more practical strategy.

Model Simplification Results

The system described in Eq. (18) is put in the form required by the simplification algorithm, namely,

$$\dot{x} = \sum_{i=1}^{11} A_i \phi_i[x(t), u(t)] \tag{19}$$

where A_i is a (3×1) vector of coefficients. In an expanded form, Eq. (19) becomes:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -0.877 \\ 0 \\ -4.208 \end{bmatrix} x_1 + \begin{bmatrix} 1 \\ 1 \\ -0.396 \end{bmatrix} x_3 + \begin{bmatrix} -0.215 \\ 0 \\ -20.967 \end{bmatrix} u + \begin{bmatrix} -0.088 \\ 0 \\ 0 \end{bmatrix} x_1 x_3 + \begin{bmatrix} 0.47 \\ 0 \\ -0.47 \end{bmatrix} x_1^2 + \begin{bmatrix} -0.019 \\ 0 \\ 0 \end{bmatrix} x_2^2 + \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} x_1^2 x_3 + \begin{bmatrix} 3.846 \\ 0 \\ -3.564 \end{bmatrix} x_1^3 + \begin{bmatrix} 0.28 \\ 0 \\ 6.265 \end{bmatrix} u x_1^2 + \begin{bmatrix} 0.47 \\ 0 \\ 46 \end{bmatrix} u^2 x_1 + \begin{bmatrix} 0.63 \\ 0 \\ 61.4 \end{bmatrix} u^3$$

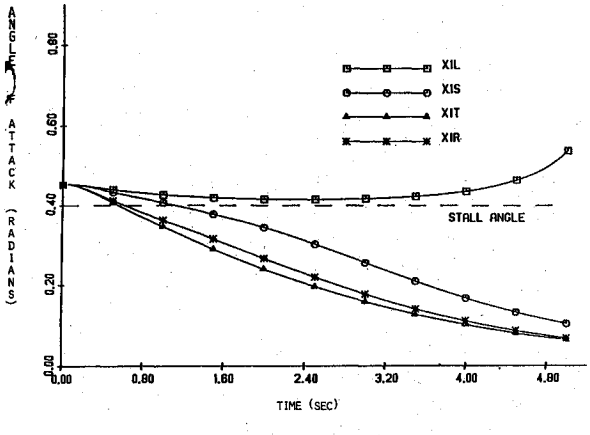

Fig. 2 Angle-of-attack response for $x_1(0) = 0.452$ rad.

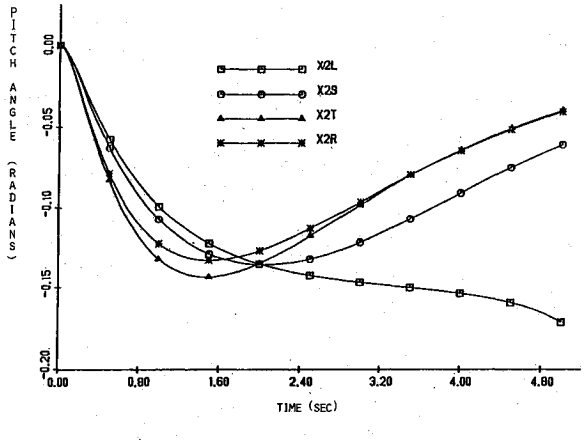

Fig. 3 Pitch angle response for $x_1(0) = 0.452$ rad.

Since it was not known exactly what input the simplified model could expect to see, the simplification process was carried out using a random input of Gaussian distribution with mean equal to -0.045 and a standard deviation of 0.018. These parameters were obtained from the physical limits on the control actuators. This choice of input is consistent with the Wiener theory of nonlinear systems.[7]

Table 1 gives a summary of all the simplified models and the error induced due to the model simplification. These results justify neglecting the nonlinear terms[3] in u. Also, they reveal that the most dominant nonlinearity is the x_1^3 term. Therefore, the optimal four-term model was chosen to find a linear controller for the original full-term model. Note that it

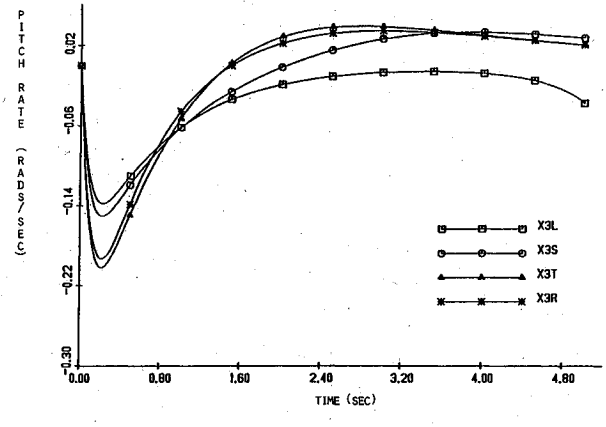

Fig. 4 Pitch rate response for $x_1(0) = 0.452$ rad.

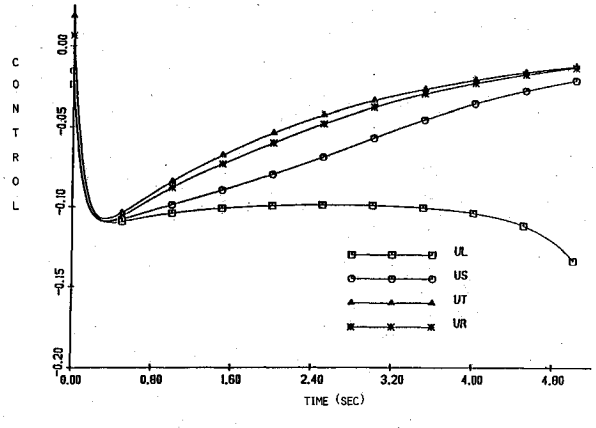

Fig. 5 Control applied for $x_1(0) = 0.452$ rad.

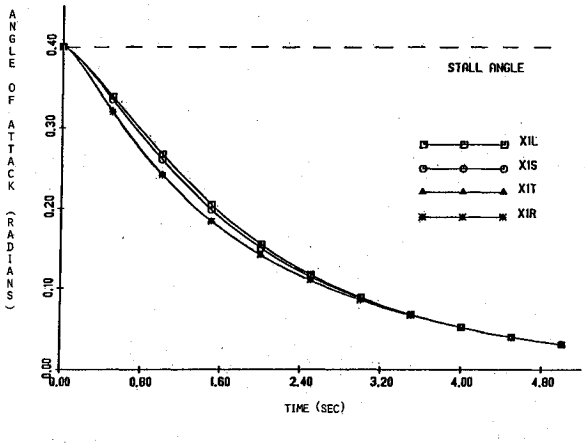

Fig. 6 Angle-of-attack response for $x_1(0) = 0.410$ rad.

is the simplification algorithm which makes this all possible and readily apparent.

Simulation was done on an IBM 370 system. Execution time for searching through all 2046 models was 20 s. This included the identification of the parameters for the optimal four-term model which is given by

$$\dot{x} = \begin{bmatrix} -0.835 & 0 & 0.940 \\ 0 & 0 & 1 \\ -4.260 & 0 & -0.395 \end{bmatrix} x + \begin{bmatrix} -0.217 \\ 0 \\ -20.992 \end{bmatrix} u + \begin{bmatrix} 5.025 \\ 0 \\ -4.584 \end{bmatrix} x_1^3$$

Optimal Control Based on the Simplified Model

The optimal open-loop nonlinear control for the simplified four-term model was found using a standard gradient technique,[8] which minimizes the Hamiltonian of the system. This control was then fitted to a linear function of the states using a least-squares error criterion. This yielded the closed-loop linear control law,

$$u^* = 0.0172x_1 + 0.584x_2 + 0.490x_3$$

This closed-loop control was then evaluated on the full term model.

Figure 2 shows the angle-of-attack (in radians) response of the F-8 Crusader Fighter when using the various controllers. The initial angle of attack is 0.452 rad and the stall angle is 0.41 rad. Note that while the linear control ($X1L$) does not recover from the stall (it even goes unstable), the second-order nonlinear control ($X1S$) recovers at 1.06 s, and the third-order nonlinear control ($X1T$) recovers at 0.512 s, while the linear control, designed from the optimal open-loop nonlinear control of the simplified four-term model, ($X1R$), recovers at 0.523 s.

Figures 3, 4, and 5 show the various responses of x_2, x_3, and u, respectively. Again, note the fact that the responses to the linear control, proposed here, and the nonlinear third-order control[3] are almost identical.

Figure 6 shows that the response of the aircraft using the different control laws is almost identical if the initial angle of attack is below the stall angle at 0.400 rad.

Conclusion

It has been demonstrated that designing optimal controllers for nonlinear plants may be easier if one works with an optimal, simplified, nonlinear model of the plant. Clearly, this may not always be the case since model reduction does not have something to offer to every problem, i.e., there may not exist a good simple approximant to the original plant. However the algorithm presented here is very efficient and allows the consideration of many model structures in a short period of time. This was demonstrated on a third-order system containing eight nonlinearities. In addition, one can consider model structures made up from any terms that are considered candidates for inclusion in the model. This is particularly valuable when there is little a priori knowledge of the plant structure.

The control results also indicate that efforts to derive high-order controllers may not always be of substantial value. Instead, a linear controller designed with an awareness of the significant plant nonlinearity can be equally effective. Although this effectiveness is not guaranteed a priori, the efficiency of the model simplification method makes consideration of this method practical and worthwhile.

Appendix: Algorithm for Selecting the Optimal Simplified Model

We seek a subset of $[\phi_i(k),\ i=1,2,\ldots,m]$ used in Eq. (6) which yields the optimal model in $\bar{S}$. Define this set as

$$G_0 = \left[\phi_{p_1}(k), \phi_{p_2}(k) \cdots \phi_{p_n}(k)\right] \quad n < m$$

where $1 \le p_1, p_2 \cdots p_n \le m$ are distinct integer subscripts. The algorithm is then:

I. From the input and output data calculate the following matrices:

$$Z_i^{(j)} \triangleq \text{diag}\left[\langle z_i(k+1), \phi_1^{(j)}(k)\rangle, \ldots, \langle z_i(k+1), \phi_m^{(j)}(k)\rangle\right] \qquad i = 1,2,\ldots,p$$

$$\Phi^{(j)} \triangleq \text{diag}\left[\langle \phi_1^{(j)}(k), \phi_1^{(j)}(k)\rangle, \ldots, \langle \phi_m^{(j)}(k), \phi_m^{(j)}(k)\rangle\right]$$

$$C_l^{(j)} \triangleq \text{diag}\left[\langle \phi_1(k), \phi_l^{(j)}(k)\rangle, \ldots, \langle \phi_{l-1}(k), \phi_l^{(j)}(k)\rangle, 0, \langle \phi_{l+1}(k), \phi_l^{(j)}(k)\rangle, \ldots, \langle \phi_m(k), \phi_l^{(j)}(k)\rangle\right] \qquad l = 1,2,\ldots,m$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product and j refers to the nodes at depth j.

Then for the costs from the root to the nodes at depth 1, calculate

$$R^{(1)} \triangleq \sum_{i=1}^{p} \left(Z_i^{(1)}\right)^2 \left[\left(\Phi^{(1)}\right)^{-1}\right]$$

$$\epsilon_{p_1} \triangleq \sum_{i=1}^{p} \left[\langle z_i(k+1), z_i(k+1)\rangle\right] - \left[R^{(1)}\right]_{p_1 p_1} \qquad p_1 = 1,2,\ldots,m$$

where $R^{(j)}$ is an $(m \times m)$ diagonal matrix and ϵ_{p_1} is an $(m \times 1)$ vector, with $[\cdot]_{p_1 p_1}$ denoting the element in row p_1 and column p_1. Now, use the following recursive algorithm, described in Steps II through VI. Set $p_1 = 0$.

II. Set $j = 1$ and increment p_1 by one. If $p_1 = m+1$, STOP; otherwise, proceed.

III. For the parent node $\phi_{p_1}\phi_{p_2} \cdots \phi_{p_j}$, the descendant nodes are

$$\phi_{p_1}\phi_{p_2} \cdots \phi_{p_{j+1}}, \text{ where } p_{j+1} = p_j + 1, p_j + 2, \ldots, m$$

IV. To calculate the costs along the branches from the parent node to the descendant nodes, compute

$$Z_i^{(j+1)} \triangleq Z_i^{(j)} - C_{p_j}^{(j)} \left[Z_i^{(j)}\right]_{p_j p_j} \Big/ \left[\Phi^{(j)}\right]_{p_j p_j} \qquad i = 1,2,\ldots,p$$

$$\Phi^{(j+1)} \triangleq \Phi^{(j)} - \left(C_{p_j}^{(j)}\right)^2 \Big/ \left[\Phi^{(j)}\right]_{p_j p_j}$$

$$C_{p_{j+1}}^{(j+1)} \triangleq C_{p_{j+1}}^{(j)} - \left\{\left[C_{p_j}^{(j)}\right]_{p_{j+1} p_{j+1}} \Big/ \left[\Phi^{(j)}\right]_{p_j p_j}\right\} C_{p_j}^{(j)}$$

$$R^{(j+1)} \triangleq \sum_{i=1}^{p} \left(Z_i^{(j+1)}\right)^2 \left\{\left(\Phi^{(j+1)}\right)^{-1}\right\}$$

where the costs are $[R^{(j+1)}]_{p_{j+1} p_{j+1}}$.

V. Calculate the error of the approximant as

$$\epsilon_{p_1 p_2 \cdots p_{j+1}} \triangleq \epsilon_{p_1 p_2 \cdots p_j} - \left[R^{(j+1)}\right]_{p_{j+1} p_{j+1}}$$

VI. When $p_{j+1} = m$, STOP; if there are no more descendant nodes, then, increment j. If $j = m-1$, go to Step II, otherwise, the descendant nodes become parent nodes. Then return to Step III and expand each of these new parent nodes.

Note that at each stage the costs are assigned from the diagonal elements of $R^{(j+1)}$.

Note that since all matrices involved are diagonal, the inverse and square operations are trivial.

Acknowledgments

This material is based upon work supported by the National Aeronautics and Space Administration under Grant NAG-1-171. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of NASA.

References

[1]Genesio, R. and Milanese, M., "A Note on the Derivation and Use of Reduced Order Models," *IEEE Transactions on Automatic Control*, Vol. AC-21, No. 1, Feb. 1976, pp. 118-122.

[2]Marshall, S.A., "The Design of Reduced Order Systems," *International Journal of Control*, Vol. 31, No. 4, 1980, pp. 677-690.

[3]Garrard, W.L. and Jordan, J.M., "Design of Nonlinear Automatic Flight Control Systems," *Automatica*, Vol. 13, Sept. 1977, pp. 497-505.

[4]Desrochers, A.A., "On an Improved Model Reduction Technique for Nonlinear Systems," *Automatica*, Vol. 17, No. 2, March 1981, pp. 407-409.

[5]Desrochers, A.A., "Optimal Model Reduction for Nonlinear Systems," *Proceedings of the 1981 Joint Automatic Control Conference*, June 15-17, 1981, University of Virginia, Charlottesville, Va.

[6]Lee, E.B. and Markus, L., *Foundations of Optimal Control Theory*, John Wiley and Sons, New York, 1967.

[7]Wiener, N., *Nonlinear Problems in Random Theory*, MIT Press, Cambridge, Mass., 1958.

[8]Kirk, D.E., *Introduction to Optimal Control Theory*, Prentice Hall, Englewood Cliffs, N.J., 1970.