A MODEL-ORDER-DEDUCTION ALGORITHM FOR NONLINEAR SYSTEMS

Bruce H. Wilson
Department of Mechanical Engineering
Northeastern University
Boston, MA
brucew@meceng.coe.neu.edu

James H. Taylor
Manager, Control and Manufacturing
Odyssey Research Associates, Inc.
Ithaca, NY
jim@oracorp.com

Abstract

An important aspect of mathematical modeling for controller design or system performance evaluation is obtaining an appropriate model order that meets the needs of the problem at hand. In modeling systems with distributed and discrete mechanical components, a model builder needs to decide the number of compliant elements to use to represent each distributed component — a potentially time-consuming and error-prone process. Nonlinear component behavior, such as dry friction and backlash, is present in most mechanical systems and further complicates the modeling process.

An algorithm was developed previously to coordinate the synthesis of the minimum-order *linear* system model that accurately characterizes the frequency response of the system over a frequency range of interest $\langle 0, \omega_{req} \rangle$. This is accomplished by iteratively determining which component refinement causes the smallest increase in model spectral radius when a more complicated component submodel is used in the system model. A more complicated submodel of this component is incorporated in the system model, and the iterative process continues until any further increase in model order results in poles beyond ω_{req}.

In the current research, this algorithm has been extended to synthesize models of nonlinear systems. The extended algorithm follows a procedure similar to the original algorithm, but uses describing-function theory to develop an amplitude-dependent quasilinear representation of the nonlinear system model. The extended algorithm synthesizes models that are also of minimum order. Algorithm operation is demonstrated using a nonlinear modeling example. Extending the original modeling algorithm to synthesize minimum-order models of nonlinear systems represents a significant extension of its utility, as a much broader class of modeling problems can be tackled.

1 INTRODUCTION

1.1 Scope

When developing a mathematical model of some physical device during the design process, modelers, both human and automated, face a bewildering number and variety of decisions. First and foremost, a modeler needs to decide the type of modeling formalism to use, e.g., a continuum approach, the finite element method, or a discrete (lumped) approach. Within a given formalism, a modeler faces additional decisions regarding the appropriate model order, the effects to include in the model, and how to treat the inherent nonlinearities involved in the physical process.

A great simplification results if the modeler represents a physical device in terms of a continuous, state-determined model, implicitly assuming that the lumped-parameter representation adequately describes the physical system under study. This representation (also called a "state-space" model) allows the system to be described with a finite (often low) number of state variables. The n state variables of such a representation are governed by a set of n ordinary-differential equations. Even with this simplification, however, a modeler still needs to decide on the appropriate order for the model, decide what effects to include in the model, and decide how to model these effects.

While a general technique for identifying the effects to include in a model and their mathematical representation is not available, a *model-order deduction algorithm* (MODA) for selecting optimal model order does exist. MODA, described in Wilson and Stein [1992], parses a set of components, each of which has one or more submodels associated with it, and coordinates the synthesis of the minimum-order lumped-parameter model that accurately predicts the frequency response of the system over a given frequency range of interest. A limitation of MODA is its inability to be applied to systems with nonlinear behavior. However, describing functions (DFs) provide a means to quasilinearize a model, and MODA can be applied to these linearized models. The combination of DFs and MODA to synthesize nonlinear models has produced a new algorithm, called *Model-Order-Deduction Algorithm for Nonlinear Systems* (MODANS), which is the main topic of this paper.

1.2 Background Overview

The algorithm described in this paper is intended for use in an automated modeling environment. One important aspect of automated modeling is the determination of the optimal model order for a state-determined system. This is a superset of the general problem of parameter lumping; hence, some of the approaches in this area will be described. System nonlinearities are accommodated via the use of describing functions to develop quasilinear models, and so a discussion of this technique is also presented.

Automated modeling, defined here as the development of algorithms and software to expedite one or more aspects of the modeling process, has begun to generate considerable interest within the mechanical engineering community [Stein, 1991; Falkenhainer and Stein, 1992], within the artificial intelligence community [de Kleer and Williams, 1991], and within the multidisciplinary simulation community [Granda and Zeigler, 1993]. The research in automated modeling encompasses software development, e.g., [van Dijk and Breedveld, 1991; Delgado, 1993; Granda, 1993; Wilson and Stein, 1993]; algorithms to derive symbolically the state equations from a bond graph [Macfarlane, 1989]; algorithms to simplify models [Rinderle and Subramaniam, 1991]; and strategies to automate the modeling process [Stein and Tseng, 1991].

Parameter lumping refers to the process by which spatially distributed phenomena are represented with a finite (usually low) number of discrete elements. Cannon [1967] provides a thorough discussion of the process. Dorny [1993] describes parameter lumping and provides a procedure for modeling translational systems. Huston [1990] describes how to represent a distributed translational shaft with a lumped 'finite-segment model', and how to determine the parameter coefficients of the latter. An alternative approach to parameter lumping is the use of finite-mode bond graphs to represent distributed behavior. This technique, invented by Karnopp [1968], is also described in [Margolis, 1985] and [Karnopp *et al.*, 1990].

Describing functions provide a means for representing nonlinear behavior in terms of amplitude-dependent "quasilinear" gains. The advantage of the DF approach is that it often enables a straightforward generalization of linear analysis and design methods to the nonlinear case. Gelb and Vander Velde [1968] and Atherton [1975] provide an extensive background on their development and use. One class of DF techniques deals with sinusoidal signals; this is called the sinusoidal-input DF (SIDF) approach. Some examples of their use for control applications are given in [Taylor and Strobel, 1985] and [Taylor and O'Donnell, 1990]. In this paper, we demonstrate the use of SIDFs to extend MODA to handle the nonlinear case.

1.3 Model-Order-Deduction Algorithm

A model-order-deduction algorithm has been developed by Wilson and Stein [1992]. MODA coordinates the search for the combination of component submodels that results in the minimum-order model that accurately characterizes the frequency response of the system over a given frequency range of interest $\langle 0, \omega_{req} \rangle$. MODA finds this model by testing the effect on the spectral radius of a model of adding a more complex component submodel to the overall system model. Eigenvalue analysis techniques are used to calculate the spectral radius, the assumption being that the model is linear.

MODA is limited to synthesizing models of linear systems. Most electromechanical systems are nonlinear, so MODA is thus quite restricted in its application. As MODA relies heavily on linear-system properties such as eigenvalue analysis and spectral radius, it cannot be used directly to synthesize nonlinear models. However, if the nonlinear models are linearized appropriately, MODA can be used to synthesize a quasilinear model with the same minimum-order properties as those found in the linear case. SIDFs, described in the next section, are used to perform this quasilinearization.

1.4 Describing Functions

The basic idea of the describing function (DF) approach for studying and modeling nonlinear system behavior is to replace each system nonlinearity with a (quasi)linear term whose "gain" is a function of "input amplitude", where the form of input signal is assumed in advance and the amplitude-dependence of the gain is based on that assumption plus an approximation-error criterion. In this work, the signals are assumed to be sinusoidal; therefore the approach is called the sinusoidal-input DF (SIDF) method. This technique is dealt with very thoroughly in a number of texts for the case of a single nonlinearity [Atherton, 1975; Gelb and Vander Velde, 1968]; for systems with more than one nonlinearity in an arbitrary configuration, the most general extensions may be attributed to [Taylor, 1975; Hannebrink *et al.*, 1977; Taylor, 1980]. These developments have been presented in tutorial form in [Ramnath *et al.*, 1980].

The SIDF approach can be used for two primary purposes: limit-cycle analysis [Taylor, 1975; Hannebrink *et al.*, 1977; Ramnath *et al.*, 1980] and characterizing the input/output behavior of a nonlinear plant [Taylor, 1980; Taylor, 1983]. It is the latter application that serves as the basis for the research presented here. The basic equations of harmonic balance that result in such an SIDF-based I/O model for a nonlinear plant are given in [Taylor, 1983]; in this paper, we will simply repeat the key equations necessary to define a SIDF I/O model as follows:

The nonlinear plant under consideration is characterized by the general state-variable differential equation and output equation

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}\left(\mathbf{x}, \mathbf{u}\right), && (1) \\ \mathbf{y} &= \mathbf{h}\left(\mathbf{x}, \mathbf{u}\right) && (2) \end{aligned}$$

where $\mathbf{x}$ is an n-dimensional state vector, $\mathbf{u}$ is an m-dimensional input vector, and $\mathbf{y}$ is a p-dimensional output vector. As stated previously, we are concerned with the behavior of the plant in the presence of sinusoidal signals, so we take $\mathbf{u}$ and $\mathbf{x}$ to have the form

$$\mathbf{u}(t) = \mathbf{u}_o + Re\left[\mathbf{a}\,\exp(j\omega t)\right], \tag{3}$$

$$\mathbf{x}(t) \approx \mathbf{x}_c + Re\left[\mathbf{b}\,\exp(j\omega t)\right] \tag{4}$$

where $\mathbf{u}_o$ represents the operating point or "d.c. value" of u(t), and $\mathbf{a}$ is a complex-valued vector designating the sinusoidal component amplitude and phase in the standard phasor notation. In developing SIDF models the state variables are assumed to be approximately sinusoidal, as indicated, where $\mathbf{b}$ is a *complex amplitude vector* and $\mathbf{x}_c$ is the state vector *center value*[1]. Then we neglect higher harmonics, to approximate the right-hand sides of Eqns. (1,2) as follows:

$$\begin{aligned} \mathbf{f}\left(\mathbf{x}, \mathbf{u}\right) &\approx \mathbf{f}_c(\mathbf{u}_o, \mathbf{a}, \mathbf{x}_c, \mathbf{b}) \\ &\quad + Re\left[A(\mathbf{u}_o, \mathbf{a}, \mathbf{x}_c, \mathbf{b}) \cdot \mathbf{b}\,\exp(j\omega t)\right] \\ &\quad + Re\left[B(\mathbf{u}_o, \mathbf{a}, \mathbf{x}_c, \mathbf{b}) \cdot \mathbf{a}\,\exp(j\omega t)\right] && (5) \\ \mathbf{h}\left(\mathbf{x}, \mathbf{u}\right) &\approx \mathbf{h}_c(\mathbf{u}_o, \mathbf{a}, \mathbf{x}_c, \mathbf{b}) \\ &\quad + Re\left[C(\mathbf{u}_o, \mathbf{a}, \mathbf{x}_c, \mathbf{b}) \cdot \mathbf{b}\,\exp(j\omega t)\right] \\ &\quad + Re\left[D(\mathbf{u}_o, \mathbf{a}, \mathbf{x}_c, \mathbf{b}) \cdot \mathbf{a}\,\exp(j\omega t)\right] && (6) \end{aligned}$$

Minimum mean square approximation error is obtained when the real vectors $\mathbf{f}_c$ and $\mathbf{h}_c$ and the matrix set $\{A, B, C, D\}$ are obtained by taking the first two terms of the Fourier expansions of the elements of $\mathbf{f}\left(\mathbf{x}, \mathbf{u}\right)$ and $\mathbf{h}\left(\mathbf{x}, \mathbf{u}\right)$ with $\mathbf{x}, \mathbf{u}$ of the form indicated in Eqns. (3,4). This approach has been discussed and illustrated in detail in [Atherton, 1975; Gelb and Vander Velde, 1968; Ramnath *et al.*, 1980]. The DF arrays $\mathbf{f}_c, \mathbf{h}_c$ and $\{A, B, C, D\}$ in Eqns. (5,6) provide the quasilinear representation of the nonlinear plant in Eqns. (1,2) for a particular amplitude and frequency; the constant or d.c. portion of the model is described by $\mathbf{f}_c$ and $\mathbf{h}_c$, while the matrix set $\{A, B, C, D\}$ (which conforms to the

[1] The center value $\mathbf{x}_c$ is generally not the equilibrium associated with $\mathbf{u}_o$, as the sinusoidal signal component gives rise to offsets due to rectification effects.

usual linearized model notation) characterizes the plant response to the sinusoidal input component. The two signal components (d.c., first harmonic) are coupled, as the above notation suggests, due to the failure of superposition in nonlinear systems. Setting up the SIDF arrays in Eqns. (5,6) in analytic form is often a simple matter: A wide variety of SIDFs for single-input nonlinearities have been catalogued, and handling multi-input nonlinearities has been dealt with in some detail in [Taylor, 1975; Ramnath *et al.*, 1980].

The use of SIDFs to determine the approximate response of a nonlinear plant in the multiple nonlinearity case is treated in a modern algebraic setting in [Taylor, 1980]. Applying the conditions of harmonic balance to the quasilinear model given in Eqns. (5,6) leads to a set of nonlinear algebraic equations that can be solved to obtain $\mathbf{x}_c, \mathbf{b}$ as determined by $\mathbf{u}_o, \mathbf{a}$ as follows:

$$\mathbf{f}_c(\mathbf{u}_o, \mathbf{a}, \mathbf{x}_c, \mathbf{b}) = \mathbf{0}, \tag{7}$$

$$\mathbf{b} \; = [j\omega I - A(\mathbf{u}_o, \mathbf{a}, \mathbf{x}_c, \mathbf{b})]^{-1} B(\mathbf{u}_o, \mathbf{a}, \mathbf{x}_c, \mathbf{b})\mathbf{a} \tag{8}$$

These $2n$ nonlinear algebraic equations (n of which are complex-valued) can be solved readily using standard computer routines; for example, MINPACK [More *et al.*, 1980] has proven to be an excellent package for this application. In this case, one should be careful to ensure that A does not have eigenvalues on or very close to the imaginary axis; otherwise limit cycles may exist in addition to the response to the sinusoidal input, in contradiction to the assumptions underlying Eqns. (3-6).

The sinusoidal component of the plant response can then be characterized by the input-amplitude-dependent matrix “transfer function”

$$G(j\omega; \mathbf{u}_o, \mathbf{a}) = C(j\omega I - A)^{-1} B + D \tag{9}$$

where the SIDF matrices are explicitly determined by $\mathbf{u}_o, \mathbf{a}$, since $\mathbf{x}_c, \mathbf{b}$ are eliminated using Eqns. (7,8). To simplify notation, in cases where the operating point $\mathbf{u}_o$ is zero, the I/O relation will be denoted $G(j\omega; \mathbf{a})$. The center $\mathbf{x}_c$ can usually be taken to be zero as well, resulting in the following simplified version of Eqn. (8):

$$\mathbf{b} \; = [j\omega I - A(\mathbf{a}, \mathbf{b})]^{-1} B(\mathbf{a}, \mathbf{b})\, \mathbf{a} \tag{10}$$

This modern algebraic method for determining the “frequency response” for a nonlinear plant (Eqn. (9)) serves as one basis for nonlinear controller design [Taylor and Strobel, 1985; Taylor and O’Donnell, 1990]; it is also directly applicable to the extension of MODA as detailed below.

2 DESCRIPTION OF ALGORITHM

The requirements of MODANS are identical to those of the original MODA. This section begins with a review of these requirements, which are discussed in more detail in Wilson and Stein [1992]. Similarly, to coordinate model synthesis, MODANS, like MODA, requires component submodel-synthesis algorithms (SSAs) to synthesize the individual component submodels. These are also reviewed in this section; a more thorough discussion is again available in [Wilson and Stein, 1992]. This section closes with a discussion of the operation of MODANS.

2.1 Algorithm Requirements

MODANS coordinates the synthesis of a lumped-parameter model of a nonlinear, distributed physical system, using as input a geometric description of the system. The resulting model should meet the following criteria: (1) the model parameters and coefficients should be directly related to the component dimensions and material constants; (2) the sum of the individual inertias in the model should equal the total inertia of the actual system; and (3) the model should be the minimum-order model that characterizes all of the poles within a frequency range of interest $\langle 0, \omega_{req} \rangle$.

The first requirement ensures that the connection between the actual system and the model is direct. In this manner, an engineer using the model during the iterative design and analysis cycle can relate changes in the dimensions and material constants of the actual system to (1) changes in the model parameters and coefficients and to (2) changes in predicted system performance. The second requirement is motivated by a need to include each component inertia in the system model, because these inertias affect the dynamic response of the system and material strength and power requirements.

The third requirement is motivated by two common model applications: the use of model to predict system response to wide-band excitation and the use of a model to design a closed-loop controller. Karnopp *et al.* [1990] discuss the use of a model to predict system response to a system input with known frequency content. They recommend that a model of a component should only be accurate to two to five times the maximum excitation frequency in the input. In the current context, accurate means that the model contains sufficient modal information to predict how the actual system would respond to frequencies in a range $\langle 0, 5\,\omega_{max} \rangle$. In this case, the frequency of $5\,\omega_{max}$ can be referred to as ω_{req}, the upper limit of the frequency range of interest.

Controller design also places frequency-related requirements on the model. In particular, it is well known that the model must accurately characterize the frequency response of the compensated system beyond the important crossings of the negative-real axis, to ensure that stability is guaranteed via the Nyquist criterion at least, and preferably that adequate bandwidth, gain- and phase-margins are achieved. In this application, ω_{max} and ω_{req} are thus determined by closed-loop system performance requirements and may have to be determined iteratively as part of the design process.

In either application, a frequency range of interest must be considered as key information in the modeling process. Thus, ω_{req} drives the *required model spectral radius*, i.e., the state-determined system model must include all system poles with Euclidean norm less than or equal to ω_{req}. In practice, MODANS will synthesize a model whose spectral radius ρ_{model} (also referred to as model bandwidth) is generally smaller than ω_{req}; however, any increase in model order will result in a model bandwidth greater than ω_{req}. Thus, the model that MODANS synthesizes is able to respond accurately to excitations with a frequency content defined by $\langle 0, \omega_{req} \rangle$.

2.2 Submodel-Synthesis Algorithms (SSAs)

SSAs are not part of MODANS; however, they are used by the algorithm during system model synthesis. SSAs are component-specific sets of instructions that describe how a geometrical description of a component is to be modeled. An important feature of an SSA is its ability to synthesize more than one component submodel, the idea being that more complex submodels are required for more accurate models. The SSAs use a variable called *rank* to control the complexity of the submodel. As an example of a simple SSA, consider the torsional shaft and its corresponding lumped-parameter model shown in Figure 1.

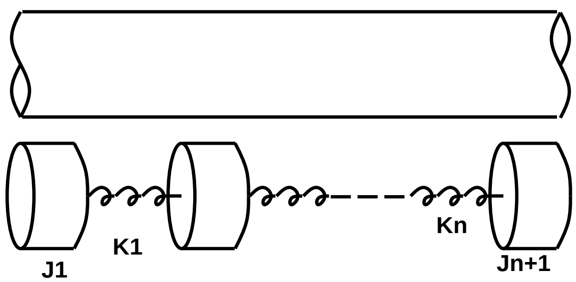

Figure 1: Torsional Shaft and Physical Model

Referring to Figure 1, the SSA for the torsional shaft relates the dimensions and material properties of the shaft to the individual inertias and compliances by the following equalities:

$$J_{shaft} = \frac{\rho \times \pi \times L \times D^4}{32} \quad (11)$$

$$J_i = \frac{J_{shaft}}{N+1} \quad (12)$$

$$K_{shaft} = \frac{G \times \pi \times D^4}{32 \times L} \quad (13)$$

$$K_i = N \times K_{shaft} \quad (14)$$

$$N \geq 0 \quad (15)$$

where K_{shaft} = the torsional spring rate of the shaft
N = the rank associated with the shaft lumped-parameter model
L = the length of the shaft
D = the diameter of the shaft
G = the shear modulus of the shaft
J_{shaft} = the torsional inertia of the shaft
J_i = inertia coefficients in the lumped-parameter model
K_i = spring rate coefficients in the lumped-parameter model

Note that for the shaft, the lumped-parameter model rank N equals the number of torsional springs in the lumped-parameter physical model of the shaft. SSAs for other components are described elsewhere [Wilson and Stein, 1992].

2.3 Algorithm Operation

MODANS uses two nested loops to coordinate model synthesis: an outer *incremental loop* that coordinates the increase of model order, and an inner *iterative loop* that determines which component of the system configuration should be modeled with a more complicated submodel. The iterative loop determines this component by identifying it as the *weak-dynamic-link* (WDL) component of a configuration:

> The WDL component of a configuration is the component that causes the smallest increase in ρ_{model} when a more complicated component submodel is incorporated in the system model, i.e., when the component's rank is increased.

The steps in the iterative loop to determine the WDL component within an M-component configuration are: For $i = 1$ to M −

1. Change rank of $component_i$
2. Use the component SSAs and submodel assembly routines to synthesize a system model (linear or nonlinear)
3. Use the system model to generate a state matrix:
   (a) if the system is linear: A is directly available as a constant;
   (b) if model nonlinearities exist: solve Eqn. 10 for $\mathbf{b}$ and $A(\mathbf{a}, \mathbf{b})$ based on the value of $\mathbf{a}$.
4. Compute ρ_{model} using A or $A(\mathbf{a}, \mathbf{b})$
5. If ρ_{model} is minimum for this iteration, $component_i$ is the WDL component.

In Step 5 above, it's important to note that the WDL component will change as model complexity increases. For example, in a three-inertia-two-shaft system, $shaft_i$ may be the WDL component during the first iteration. In most cases, $shaft_j$ ($j \neq i$) will be the WDL component during the second iteration.

The incremental loop is fairly straightforward:

1. Use the iterative loop to identify the WDL component
2. Increase the rank of the WDL component; calculate ρ_{model}
3. If $\rho_{model} \leq \omega_{req}$, repeat steps 1 and 2
4. If $\rho_{model} > \omega_{req}$, decrease the rank of the most recent WDL component

Step 4 is necessary, because the first three steps may synthesize a model with $\rho_{model} > \omega_{req}$, whereas the goal is to synthesize a model that includes only modes at frequencies less than or equal to ω_{req}.

3 USE OF ALGORITHM

Experiments were performed to test the viability of using MODANS to synthesize nonlinear state-determined system models. Three experiments were performed. The first entailed using MODANS in a simulated automated-modeling program, i.e., the steps that such a program (if it incorporated the algorithm) would perform in synthesizing a model were mimicked and the models that it synthesized were recorded. The second and third experiments studied the effect of changing the excitation on model bandwidth. In the second experiment a large excitation amplitude at several input frequencies was used; conditions were identical for the third experiment, except a smaller excitation amplitude was used.

3.1 Step-by-step Model Synthesis

MODANS was used to synthesize a model of the drive train in Figure 2, whose component dimensions and material properties are given in the Appendix. The upper bound of the frequency range of interest, ω_{req}, was set at 10 radians/second.

The gear teeth in Figure 2 have backlash and a non-zero compliance. A plot of the

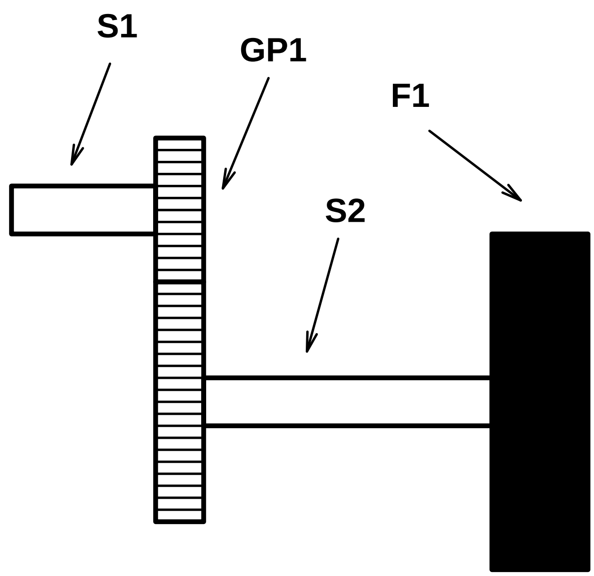

Figure 2: Drive Train with Nonlinear Behavior

translational force developed between the teeth vs. the relative deflection between the teeth is shown in Figure 3.

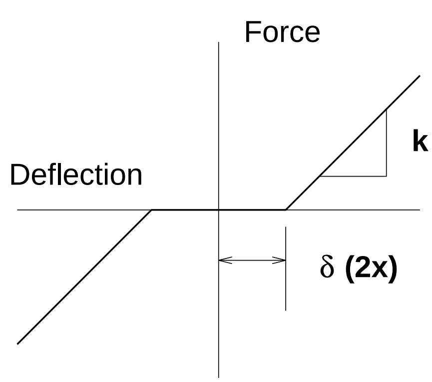

Figure 3: Force at Gear Interface vs. Deflection

To synthesize a model of the drive train, MODANS first uses its iterative algorithm to find the first WDL. This algorithm sequentially increases the rank of each component in the configuration, and the resulting ρ_{model} is calculated for each model. The results for the first iteration of model synthesis for the drive train Figure 2 are summarized in Table 1.

Rank of Component 1	Rank of Component 2	Rank of Component 3	ρ_{model}
1	0	0	45.18
0	1	0	0.782
0	0	1	2.21

Table 1: Model bandwidths during Pass 1

In Table 1, the “010” model (i.e., the model that resulted from a rank-0 model of the first shaft, a rank-1 model of the gear pair, and a rank-0 model of the second shaft) has the smallest bandwidth. At this point the incremental algorithm determines that $\rho_{model} < \omega_{req}$, and the 010 model becomes the starting point for the second pass of MODANS. The results from the second pass are given in Table 2.

Rank of Component 1	Rank of Component 2	Rank of Component 3	ρ_{model}
1	1	0	51.64
0	1	1	2.528

Table 2: Model bandwidths during Pass 2

During the second pass of the algorithm, the 011 model has the minimum bandwidth. Note that during this iteration, the rank of the gear pair is not increased – which, according to the algorithm, would be expected. The rank is constant, because the gear pair is a *bounded-rank* component ([Wilson, 1992, p. 24] discusses this type of component), and it has achieved maximum rank . In this case, a submodel of an order greater than that associated with the rank-1 gear pair model is not available.

To continue with the model synthesis process, the incremental algorithm determines that $\rho_{model} < \omega_{req}$, and uses the 011 model is the starting point for the third pass of MODANS. The iterative algorithm is called for each of the configuration components whose rank can still be increased. The results from the third pass of MODANS are presented in Table 3.

Rank of Component 1	Rank of Component 2	Rank of Component 3	ρ_{model}
1	1	1	51.64
0	1	2	11.26

Table 3: Model bandwidths during Pass 3

During the third pass of the algorithm, an increase in the rank of each component results in $\rho_{model} > \omega_{req}$. As the goal is to synthesize a model that estimates all modes up to and including ω_{req} (but not beyond this frequency), the incremental algorithm would determine that the 011 model, obtained during pass 2, is the optimal model.

3.2 Effect of Different Excitations on Model Bandwidth

During the first pass of the algorithm, the results of which are reported in Table 1, the inclusion of the gear-train compliance led to the model with the lowest bandwidth. To test the sensitivity of the bandwidth of this "010" model to changes in the excitation frequency and amplitude, two experiments were performed. In the first, the input amplitude was set to 200 N-m, and the excitation frequency was given values of 10, 20, and 40 rad/sec. Model bandwidth and the maximum relative displacement between the compliant gear teeth were recorded. The resulting model bandwidths and displacements are tabulated in Table 4, which indicates that higher input frequencies result in lower model bandwidths. Note that the model bandwidth for the zero-backlash case (i.e., for a linear compliance

Amplitude	ω_{in}	ρ_{model}	Max Displacement
200	1	0.825	15.65
200	10	0.814	0.0503
200	20	0.782	0.0125
200	40	0.639	0.003

Table 4: Model bandwidths for 200 N-m input magnitude

between the gears) is 0.825 rad/sec.

In the second experiment, the input amplitude was set to 50 N-m, and the same excitation frequencies were used. As before, model bandwidths and gear displacements were recorded, and the resulting model bandwidths and displacements are tabulated in Table 5. In Table 5 the change in input frequency from 10 rad/sec to 20 rad/sec

Amplitude	ω_{in}	ρ_{model}	Max Displacement
50	10	0.782	0.0125
50	20	0.639	0.003
50	40	0.709	0.00078

Table 5: Model bandwidths for 50 N-m input magnitude

corresponds to a decrease in the model bandwidth. Conversely, the change in input frequency from 20 rad/sec to 40 rad/sec corresponds to an increase in model bandwidth. This anomaly will be explained in the next section.

4 DISCUSSION

This paper presents an algorithm that coordinates the synthesis of a state-determined system mathematical model from a high-level physical system description. This algorithm, MODANS, employs two algorithmic loops. The outer, supervisory loop increases model order until any subsequent increase in model order results in a model bandwidth greater than the required model bandwidth. The inner, iterative loop identifies the WDL component, i.e., the component that causes the lowest increase in model bandwidth when a more complicated component submodel is used in the overall system model. MODANS is intended for use with any unidimensional system with distributed and discrete components. To coordinate model synthesis, MODANS will need to be supported by submodel-synthesis algorithms, which build submodels of the individual components, and a collection of describing functions to represent nonlinear component behavior in the models.

The response characteristics of linear systems, such as the placement of the eigenvalues and the transfer function, are independent of the input. The same cannot be said for nonlinear systems or the quasilinearized models that MODANS synthesizes. Both the frequency and the amplitude of the input signal affect the response characteristics of the quasilinear model. As shown in the previous section, depending on the input signal and the particular nonlinearities being represented in the model, response characteristics may asymptotically approach equivalent linear model behavior (Table 4), or conversely, may diverge from this behavior (Table 5). Both of these cases are discussed in the following section. After that, a discussion of related research is presented, followed by a suggested application for MODANS. This section closes with a discussion of recommended extensions to MODANS.

4.1 Effect of Input Signal Amplitude on Model Bandwidth

In Section 3.2 a constant-amplitude, variable-frequency input excitation was used to study how this frequency variation would affect model bandwidth. As excitation frequency was decreased the corresponding model bandwidth approached the model bandwidth for the zero-backlash (linear) case. This tendency can be attributed to the fact that the lower input frequency results in larger excursions of the states, including the relative displacement between the gears. This relatively large displacement (15,000 times that of the backlash) resulted in an operating condition that was primarily in the linear region of gear operation. Hence, the describing function representation of the compliance between the gears resulted in a coefficient almost identical to that of the zero-backlash gears. Conversely, as the excitation frequency was increased and state displacements decreased, the describing function representation of the gear backlash and compliance was that of a more compliant spring. Hence, the model bandwidth decreased as the excitation frequency increased.

In Table 5 model bandwidths and gear displacements are reported for the case of an input amplitude of 50 N-m and excitation frequencies of 10, 20 and 40 rad/sec. In the change of 10 to 20 rad/sec, model bandwidth changed from 0.782 to 0.639. Based on the logic in the previous paragraph, this trend is expected, because higher input frequency corresponds to smaller state excursions. When the input frequency was set at 40 rad/sec, model bandwidth *increased* to 0.709 rad/sec. Furthermore, the eigenvalue locations used to calculate this bandwidth were $0.5 \pm 0.5j$. Such a result is counter-intuitive for two reasons: model bandwidth was expected to decrease, and system eigenvalues were expected to lie solely on the imaginary axis (as the system had no dissipative elements).

Closer scrutiny reveals that this anomalous behavior was caused by the fact that the amplitude of the relative displacement between the gears was smaller than the gear backlash (0.001 meter) for $\omega_{in} = 40$ rad/sec – see Table 5. Thus, in reality, at the 50 N-m / 40 rad/sec input condition the second gear would not be driven; the SIDF is set to zero and the resulting degenerate model gave rise to spurious eigenvalues which were not located on the imaginary axis.

As in most engineering analysis, the onus is on the user (or the automated-modeling software) to ensure that spurious results are not accepted with blind faith. In this case, one may easily guard against this by including a check in the automated-modeling software for conditions such as the input amplitude to a backlash being below the threshold δ (Fig. 3). To cite another example, one can also check that the predicted forces (or torques) that act on an inertia must be larger than the magnitude of the dry friction (if it exists) that will resist its motion.

4.2 Related Research

This research addresses the problem of choosing the appropriate complexity of component submodels which are to be included in a larger system model. Other work in this area includes an algorithm developed by Wilson and Stein [1992], finite-mode bond graphs [Karnopp *et al.*, 1990], and a heuristic method to determine how many segments are needed to model a distributed component [Dorny, 1993].

For linear systems, the functionality of MODANS is identical to the algorithm developed by Wilson and Stein [1992]. The principle difference between these algorithms is MODANS' added capability to synthesize minimum-order, quasilinearized models of systems whose components submodels exhibit nonlinear behavior. The hill-climbing search strategy and the quantity of models tested, both discussed in [Wilson and Stein, 1992], are identical for the two algorithms.

As discussed in Section 2.2, submodel-synthesis algorithms are used to obtain component submodels. For distributed components such as the torsional (and translational) shaft, the method of dividing the component into $N + 1$ inertial elements and N compliant elements is known as finite-segment modeling [Huston, 1990]. With MODANS, N (the component rank) is increased until any subsequent increase results in $\rho_{model} > \omega_{req}$. Finite-mode bond graphs [Karnopp *et al.*, 1990] provide a method to obtain a component submodel that does not involve the iterative procedure of MODANS. The technique of Karnopp *et al.* involves solving a partial-differential equation with a series solution and truncating this solution at a specific modal frequency. A bond graph of this truncated solution is created and used to represent component dynamics in the system model. In that modal damping can easily be added to the model, this method has an advantage over finite-segment model. Furthermore, the model, which is obtained from an exact solution to a partial-differential equation, is more accurate for higher modal frequencies. However, finite-segment models are generally sufficient for fundamental modes of distributed components, they are simpler — their coefficients are readily related to component dimensions and material properties (a key advantage), and, if necessary, modal damping can be added to the system matrix.

Dorny [1993] provides a heuristic for determining the number of lumped elements that are needed to model a distributed component. He states that "a distributed object must be subdivided into segments at least 10 times shorter than the wavelength of the [a] propagating wave." For a sinusoidal input acting on a distributed torsional shaft, the maximum segment length can be determined by the following equation:

$$L_{seg} \leq \frac{2\pi\sqrt{G/\rho}}{10 \times F_{excitation}} \tag{16}$$

where L_{seg} = the length of the segment
G = the shear modulus of the segment

ρ = the density of the segment
$F_{excitation}$ = torsional excitation frequency

If ω_{req} is used as the excitation frequency, Eqn. 16 can be used to determine the number of lumped inertias needed to model a torsional shaft.

Although the finite-mode bond graphs and the heuristic discussed in the previous paragraph both provide methods for modeling distributed components, neither provides a criterion or methodology for accommodating nonlinear component behavior in a model.

4.3 Applications

MODANS synthesizes system models that meets specific criteria; the intermediate models obtained during this synthesis process contain useful design information. When MODANS identifies a WDL component, this is an indication that this component causes the next mode (generally a structural resonance) in a system model. If the modal frequency caused by this component is too low, an engineer can make the appropriate changes to *this component*, e.g., alter the material, a physical dimension, or even a nonlinear characteristic such as backlash, so that the modal frequency is increased. In this manner, the intermediate models that MODANS synthesizes also become useful design tools.

MODANS provides a means of systematically selecting the required complexity of component submodels within a larger system model. As a by-product, the required complexity of the system model is also determined. MODANS is intended to be used in an automated-modeling program. With such a program, a user would be expected to provide a system-configuration description, a desired ρ_{model} or other model criteria, and a specific input amplitude and frequency. The program would then use MODANS to build the system model.

MODANS predecessor algorithm, MODA, has already been successfully implemented in an automated-modeling software called "Model-Building Assistant" (MBA) [Wilson and Stein, 1993], which used linear component models to synthesize linear system models. Much of the MBA program architecture [Wilson, 1992] can be reused with MODANS; however, the desire to model nonlinear component behavior will require some modifications in the program. Whereas MBA simply had to calculate the spectral radius of a constant coefficient state matrix, a program using MODANS will require the following additional steps to synthesize a quasilinearized state matrix (for which eigenvalues can be calculated):

1. Nonlinear component behavior will need to be represented as unknown SIDF coefficients in a system state matrix.
2. The quasi-steady-state response of a set of nonlinear state equations will need to be determined via solution of Eqn. 10.
3. This response will be used to determine the numerical values of the coefficients of the quasilinear state matrix.

4.4 Extensions

In the modeling of actual physical systems, several situations may arise that have not been addressed in the current research. Some of these situations are: inertial elements may be separated by very stiff compliances with backlash and a large number of nonlinearities may be present in the actual system.

MODANS uses a describing function technique to approximate nonlinear behavior. In the case of inertial elements separated by a compliance and backlash, a higher value of the zero-backlash compliance replaces the original compliance. If the model bandwidth caused by this linearized compliance is the lowest during a given MODANS iteration, this compliance will be included in the model. Cases may arise, however, in which the compliance between the inertial elements may be very low (resulting in a very high frequency model bandwidth), but backlash may still be present. If this backlash is not included in the model, an overly optimistic prediction of system behavior may result[2]. The high-frequency model bandwidth may necessitate excessively small integration intervals or a large number of data points. Further research is needed to determine how to handle such cases. One possible solution is to modify MODANS so that it leaves the compliance out of the model, but kinematically couples the inertias during contact and uncouples them inside of the backlash.

A large number of potential nonlinearities is another problem facing modelers. Closer attention to real-world phenomena invariably results in fewer cases of linear behavior. Indeed, linear behavior is perhaps more of a convenient mathematical abstraction than a precise description of real-world behavior. Yet during the modeling and analysis process, a few key nonlinearities (or even a single one) are generally sufficient to depict the phenomena. The automated identification or ranking of the most significant nonlinearities in a system model is a topic that merits further research – it is likely, based on the results presented here, that SIDF methods will provide the means of meeting this need.

5 CONCLUSION

The main contribution of this research is an algorithm, MODANS, that coordinates the synthesis of mathematical models of linear and nonlinear systems. This algorithm was developed by modifying a linear model-order-deduction algorithm. The original algorithm relies on the linear-system property of spectral radius to determine which poles to include in a model. While this property cannot be directly used with nonlinear systems, describing function theory can be employed to quasilinearize the nonlinear model and the idea of model spectral radius can be employed in that generalized setting. To synthesize a minimum-order model, MODANS first synthesizes a nonlinear set of state equations. Next, describing functions are used to obtain an amplitude-dependent, constant-coefficient state matrix. The spectral radius is then calculated from this state matrix, and used as a basis for determining the need to include a more complicated component submodel in the system model.

In the Results section, MODANS was used to synthesize a model that accurately characterizes the frequency response of the system over a given frequency range of interest. We believe that MODANS can be used to coordinate the model synthesis of any unidimensional system, linear or nonlinear, that contains distributed and discrete components. Provided that appropriate component submodel-synthesis algorithms exist, MODANS can be used on any set of components.

[2] Reduced backlash generally results in improved system behavior.

References

[Atherton, 1975] D. P. Atherton. *Nonlinear Control Engineering*. Van Nostrand Reinhold Company Ltd., London, UK, 1975.

[Cannon, 1967] Robert H. Cannon. *Dynamics of Physical Systems*. McGraw-Hill, New York, NY, 1967.

[de Kleer and Williams, 1991] Johan de Kleer and Brian C. Williams, editors. *Artificial Intelligence: Special Volume Qualitative Reasoning about Physical Systems II*, volume 51. North-Holland Publishing Company, 1991.

[Delgado, 1993] Marisol Delgado. Non-linear desis — a package for the modeling and simulation using bond graphs. In Jose Granda and Bernard Zeigler, editors, *Proceedings of the 1993 SCS Western Multiconference*, San Diego, CA, 1993. Society for Computer Simulation.

[Dorny, 1993] C. Nelson Dorny. *Understanding Dynamic Systems: Approaches to Modeling, Analysis, and Design*. Prentice Hall, Englewood Cliffs, NJ, 1993.

[Falkenhainer and Stein, 1992] Brian Falkenhainer and Jeffrey L. Stein, editors. *Automated Modeling, DSC-Vol. 41*. American Society of Mechanical Engineers, New York, NY, 1992.

[Gelb and Vander Velde, 1968] Arthur Gelb and Wallace E. Vander Velde. *Multiple-Input Describing Functions and Nonlinear System Design*. McGraw-Hill Book Company, New York, NY, 1968.

[Granda and Zeigler, 1993] Jose Granda and Bernard Zeigler, editors. *Proceedings of the 1993 SCS Western Multiconference*. Society for Computer Simulation, San Diego, CA, 1993.

[Granda, 1993] Jose J. Granda. Computer aided modeling of multiport elements and large bond graph models with camp-g. In Jose Granda and Bernard Zeigler, editors, *Proceedings of the 1993 SCS Western Multiconference*, San Diego, CA, 1993. Society for Computer Simulation.

[Hannebrink *et al.*, 1977] D.N. Hannebrink, H.S.N. Lee, H. Weinstock, and J.K. Hedrick. Influence of axle load, track gauge, and wheel profile on rail vehicle hunting. *Transactions of the ASME - Journal of Engineering and Industry*, 1977.

[Huston, 1990] Ronald L. Huston. *Multibody Dynamics*. Butterworth-Heinemann, Boston, MA, 1990.

[Karnopp *et al.*, 1990] Dean Karnopp, Donald Margolis, and Ronald C. Rosenberg. *System Dynamics: A Unified Approach*. Wiley-Interscience, New York, NY, 1990.

[Karnopp, 1968] Dean Karnopp. *Analysis and Simulation of Multiport Systems*. MIT Press, Cambridge, MA, 1968.

[Macfarlane, 1989] Jane Frances Macfarlane. *Qualitative and Symbolic Analysis of Dynamic Physical Systems*. PhD thesis, The University of Minnesota, 1989.

[Margolis, 1985] Donald L. Margolis. A survey of bond graph modeling for interacting lumped and distributed systems. *Journal of the Franklin Institute*, 319(1/2):125–135, 1985.

[More *et al.*, 1980] J.J. More, B.S. Garbow, and K.E. Hillstrom. *User Guide for MINPACK-1*. Argonne National Laboratory, August 1980. Report No. ANL-80-74.

[Ramnath *et al.*, 1980] R.V. Ramnath, J.K. Hedrick, and H.M. Paynter, editors. *Nonlinear System Analysis and Synthesis: Volume 2 - Techniques and Applications*. American Society of Mechanical Engineers, New York, NY, 1980. Book G00178; Chapters 7 – 9.

[Rinderle and Subramaniam, 1991] James R. Rinderle and Bala L. Subramaniam. Automated bond graph modeling simplification to support design. In Jeffrey L. Stein, editor, *Automated Modeling 1991, DSC-Vol 34*. American Society of Mechanical Engineers, December 1991. The Winter Annual Meeting.

[Stein and Tseng, 1991] Jeffrey L. Stein and Yih Tun Tseng. Strategies for automating the modeling process. In Jeffrey L. Stein, editor, *Automated Modeling 1991, DSC-Vol 34*. American Society of Mechanical Engineers, December 1991. The 1991 Winter Annual Meeting.

[Stein, 1991] Jeffrey L. Stein, editor. *Automated Modeling DSC-Vol. 34*. American Society of Mechanical Engineers, 1991.

[Taylor and O'Donnell, 1990] James H. Taylor and James R. O'Donnell. Synthesis of nonlinear controllers with rate feedback via sinusoidal-input describing function methods. In *Proceedings of the American Control Conference*, San Diego, CA, May 1990.

[Taylor and Strobel, 1985] James H. Taylor and Kevin L. Strobel. Nonlinear compensator synthesis via sinusoidal-input describing functions. In *Proceedings of the American Control Conference*, Boston, MA, June 19-21 1985.

[Taylor, 1975] James H. Taylor. *An Algorithmic State-Space/Describing Function Technique for Limit Cycle Analysis*. The Analytic Sciences Corp., Reading, MA, 1975.

[Taylor, 1980] James H. Taylor. Describing function methods for high-order highly nonlinear systems. In *Proceedings of the International Congress on Applied Systems Research and Cybernetics*, Acapulco, Mexico, December 1980.

[Taylor, 1983] James H. Taylor. A systematic nonlinear controller design method based on quasilinear system models. In *Proceedings of the American Control Conference*, San Francisco, CA, June 1983.

[van Dijk and Breedveld, 1991] J. van Dijk and P.C. Breedveld. Automated mechatronic system modeling. In *Proceedings of the IMACS 13th World Congress on Bond Graphs*, Dublin, 1991.

[Wilson and Stein, 1992] Bruce H. Wilson and Jeffrey L. Stein. A algorithm for obtaining minimum-order models of distributed and discrete systems. In Brian Falkenhainer and Jeffrey L. Stein, editors, *Automated Modeling, DSC-Vol. 41*, New York, NY, 1992. American Society of Mechanical Engineers.

[Wilson and Stein, 1993] Bruce H. Wilson and Jeffrey L. Stein. Model-building assistant: An automated modeling tool for machine-tool drive systems. In Jose Granda and Bernard Zeigler, editors, *Proceedings of the 1993 SCS Western Multiconference*, San Diego, CA, 1993. Society for Computer Simulation.

[Wilson, 1992] Bruce H. Wilson. *Model-Building Assistant: An Automated Modeling Tool for Machine-Tool Drive-Trains.* PhD thesis, The University of Michigan, 1992.

APPENDIX : MODEL PARAMETERS FOR THE EXAMPLE IN SECTION 3

All units in SI, e.g., inertia = kg/m^2
Shaft 1
Inertia of S1: 1
Stiffness of S1: 1000
Gear Pair
Inertia of Gear 1: 1
Radius of Gear 1 / Radius of Gear 2: 0.5
Stiffness of Gear 1 / Gear 2 Interface: 500
Backlash of Gear 1 / Gear 2 Interface: 0.001
Inertia of Gear 2: 16
Shaft 2 Inertia of S2: 10
Stiffness of S2: 100
Flywheel Inertia of F1: 64