Accurate Models for Bushings and Dampers using the Empirical Dynamics Method

Andrew J. Barber

Modeling Integration Products and Services
MTS Systems Corporation

Abstract

Conventional models for complex nonlinear components (like bushings and dampers) are often inadequate to represent behavior over wide frequency ranges, and at large amplitudes. Empirical Dynamics™ models provide a solution to this problem, via a nonlinear, dynamic, blackbox approach. These models accurately replicate both frequency and amplitude dependence effects, without the limitations of conventional schemes. Examples of Empirical Dynamics models will be presented for automotive shock absorbers and a biaxial rubber bushing. Benefits and limitations of the Empirical Dynamics approach are discussed along with requirements for interfacing these models to the ADAMS virtual prototyping environment.

1 Introduction

ADAMS virtual prototype models have shown impressive results in the ability to simulate behavior of complex engineering systems. They make it possible to study the effects of any type of dynamic input, for a wide range of systems. They also permit manipulation of system variables in a controlled manner, as required for optimizing system design. As a result, these techniques are rapidly revolutionizing the way engineering systems are designed.

Despite the successes of the ADAMS approach, there are nevertheless limitations. Systems that include friction, rubber or other elastomeric materials, complex fluid dynamic behavior, biological systems, and others, can be difficult to model accurately in reasonable time. The difficulty arises for some systems because they require a very large number of state variables for an adequate representation. For example, fluid dynamic systems may require simulation of large, complex flow fields, just to assess force vs. velocity relations. For other systems, difficulties arise because important physical properties cannot be easily measured or understood. Another source of problems occurs for systems whose properties are not repeatable, or whose properties change significantly with temperature.

Often, these complexities are handled by using blackbox models. Here, experimental data is obtained for inputs and outputs of a complex system or component, and a relation between the inputs and outputs is formed using a mathematical curve fitting technique. Examples include polynomial or spline curve fits for damper force vs. velocity, and frequency dependent bushing models, which provide stiffness coefficients as functions of frequency. Blackbox methods are effective because they largely ignore internal operation of the physical system. This is an obvious advantage, when the physical principles are not well understood, or when the system model requires an enormous number of states.

The blackbox approach contrasts sharply with analytical modeling, or 'first principles' modeling, where geometric parameters, mechanical properties and physical principles are used to generate differential equations, and these are solved numerically. Examples of these include the finite element method, computational fluid dynamics, and multi-body dynamics (in a limited sense). Not surprisingly, these latter methods have been called 'whitebox' methods. Of course, the whitebox and blackbox methods are not mutually exclusive. Curve fits representing complex component behaviors are easily integrated into the ADAMS 'whitebox' environment.

One important difference between blackbox and whitebox models is their adjustability. A major disadvantage for blackbox models is their lack of adjustability. Many of the important physical or design parameters are hidden in the blackbox, and cannot be changed. Alternatively, a blackbox model has parameters (coefficients for a curve fit), but these often do not have meaningful physical counterparts. This makes blackbox models less than desirable for design optimization. In contrast, the parameters of a whitebox model often have immediate and clear interpretations, and are very useful for design studies.

Another important difference between blackbox and whitebox models is their computational efficiency. Blackbox models are superior in this regard. They are typically represented by straightforward algebraic relations, which can be solved by direct (non-iterative) calculation. Whitebox models, by contrast, require a stepwise solution of complex differential and algebraic equations (DAE's), often with considerable iteration required.

Clearly, there is a direct tradeoff between adjustability and efficiency. The choice of blackbox vs. whitebox models must be based on a clear understanding of the modeling objective. For example, a vehicle model, built to study ride and handling characteristics, requires a tire model. A whitebox tire model might be constructed using finite elements, with nonlinear stiffness and damping coefficients for the rubber. A blackbox model of the same tire could be constructed by fitting curves to experimental data (for example, lateral force vs. slip angle).

The choice of blackbox vs. whitebox in this case may depend on the focus of the model. If it is to improve the tire design, then clearly the whitebox model is best. If it is to improve a suspension design, then the blackbox model may be sufficient. As a general philosophy, it makes sense to use blackbox models whenever possible, i.e., for elements that are not the focus of the design optimization.

Of course, the accuracy of the two methods must also be considered. Historically, accuracy has been one of the most serious challenges for both blackbox & whitebox modeling. Whitebox methods may be inadequate for certain types of systems, as mentioned above. Blackbox methods have had shortcomings for systems exhibiting characteristics such as:

- amplitude dependence (or nonlinearity)
- frequency dependence (or memory)
- arbitrary (or random) input
- multiple inputs and outputs

Blackbox modeling is relatively straightforward, if the system has only amplitude dependence or only frequency dependence. When these are present *simultaneously*, however, the necessary mathematical sophistication increases dramatically. Limitations in the type of system, type of input, or number of inputs, then become necessary to keep the blackbox methods manageable.

In recent years, blackbox techniques have been developed which can handle joint amplitude and frequency dependence, with very few restrictions on the class of system or type of inputs. This paper will discuss one such technique called Empirical Dynamics Modeling (ED modeling or EDM™).

The presentation begins with some background, covering definitions and explaining why blackbox modeling becomes difficult when amplitude and frequency dependence are present simultaneously. The next section introduces Empirical Dynamics models, with a brief overview of their principles of operation. A series of case studies follow, with examples of Empirical Dynamics models for complex vehicle subsystems and components such as shock absorbers and rubber bushings. In the next section, some general characteristics of Empirical Dynamics models are presented, including benefits, limitations, and methods to circumvent those limitations. Finally, the methods to interface ED models to the ADAMS environment will be discussed

2 Background

This section provides a brief overview of conventional blackbox modeling. It discusses systems having only amplitude dependence or only frequency dependence, and how these are modeled using blackbox methods. It then describes the complications involved when a system has both amplitude and frequency dependence, especially how the conventional models fail to provide an adequate representation. Finally, a brief discussion of some alternative methods and workarounds is provided.

Amplitude dependence is synonymous with nonlinearity, which refers to the situation where superposition fails to hold. That is, if system input amplitude is scaled by some factor, the output amplitude doesn't scale by the same factor. Most physical systems display this property to some extent. A simple example of a nonlinear system is an automobile shock absorber, whose force vs. velocity relation is modeled by a simple curve or a two-part linear relation (Figure 2.1). Note that linearity should not be confused with proportionality. Proportional systems (i.e., output is a scalar multiple of input) are linear, but the converse need not be true. The correct way to understand linearity is by the effect of scaling: if the input is multiplied by some scale factor, the output will scale by the same factor, but the input and output may look completely different.

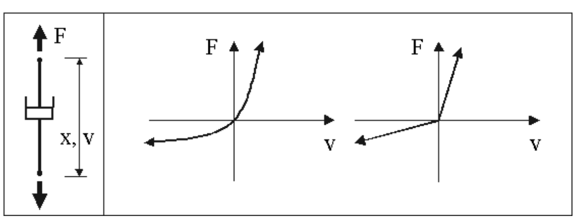


Figure 2.1 Simple Nonlinear Damper Models

Amplitude dependent behavior is often modeled using simple input-output curves or algebraic functions, as shown for the shock absorber above. This method applies when the system has a single input and single output (SISO). When such a system has multiple inputs, the simple curve characterization must be replaced with a higher dimensional equivalent. For two inputs, the relation must be represented by a family of curves in two dimensions, or by a surface in three dimensions (3D). For three inputs, it becomes more complex: now a family of surfaces in 3D is needed. With more inputs, more dimensions are needed, and the ability to visualize the relationship is lost. Nevertheless, equations may be fitted to define a useful model. This approach of using simple curves or surfaces, or equivalent equations, is applicable when there are no frequency dependent effects. Under these conditions, the model can be used for any type of input signal: sinusoidal, step, sweep, random, or arbitrary.

Frequency dependence refers to a system whose behavior depends on the rate or time scale of input. For instance, a frequency dependent system subjected to a 1 Hz sinusoidal input may give a 1 Hz sine output, but with 30 degrees phase lag. When 10 Hz is applied, the phase lag may be 45 degrees (Figure 2.2). The ratio of output/ input amplitudes may differ as well, at the two frequencies. Resonance is an example of such a frequency dependent effect.

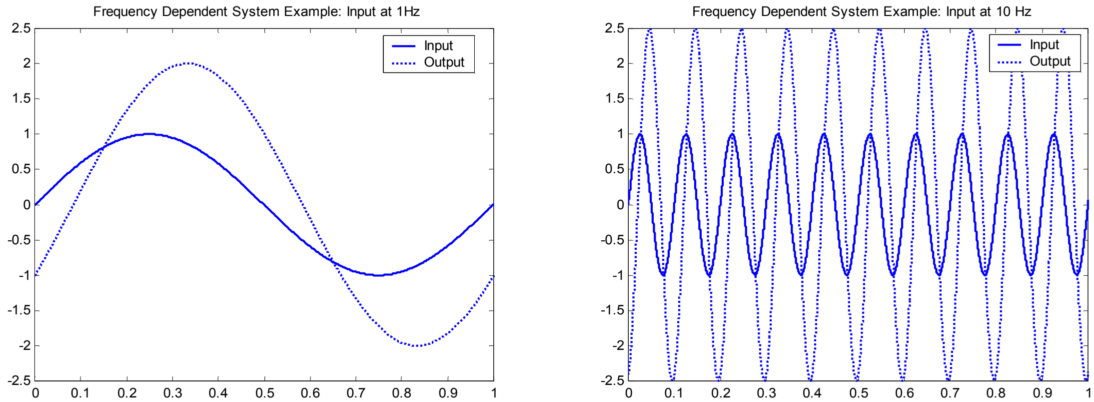


Figure 2.2 Frequency Dependent Behavior

Frequency dependent systems are often modeled via frequency response functions (FRFs), where the relation between input and output signals is characterized in terms of magnitude and phase vs. frequency. FRFs are typically explained for the case of sinusoidal input and output: The FRF magnitude gives the ratio of sinusoidal output amplitude over sinusoidal input amplitude, and the FRF phase is the difference between output phase and input phase (Figure 2.3). Transfer functions, based on Laplace transforms, offer a similar capability. Frequency dependent systems can alternatively be modeled in the time domain, using ARMA type models, adaptive digital filters, and others. The time domain model is characterized using the impulse response of the system, and the output at any time is evaluated by convolution of the signal with the impulse response.

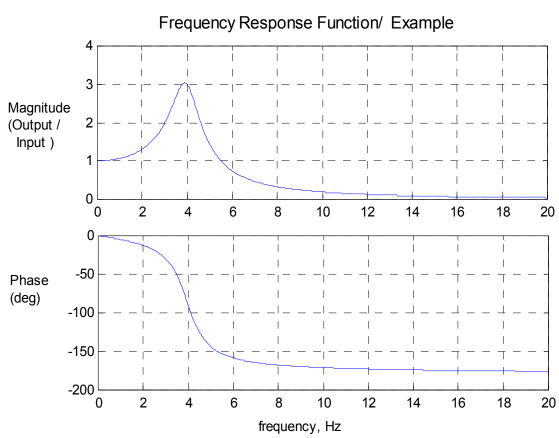


Figure 2.3 Frequency Response Function (FRF)

Note that these techniques for modeling frequency dependent systems are applicable, in a strict sense, only for *linear* systems (i.e., no amplitude dependence). Linear frequency dependent systems can often be identified by the fact that a sinusoidal input gives a sinusoidal output. If the output is not sinusoidal, this is a sure sign of nonlinearity. It's important to keep in mind, however, that linearity or nonlinearity is a property of the *system*, independent of input signal type. Note also that linearity does not mean the FRF magnitude is flat. Resonant peaks and rolloff at high or low frequencies may (and typically do) occur for linear systems. A flat FRF magnitude does have a special interpretation, however: it means the system has no memory. This property will be explained shortly.

The methods for modeling frequency dependent systems can also deal with arbitrary input waveforms. For example, FRFs can be used in a straightforward manner to model systems with random inputs. In this case, the random signals are taken to be composed of a large number of sinusoidal components. The amplitudes of these components in the signal are characterized in terms of power spectral density (PSD, or power spectrum, Figure 2.4). When random inputs are applied to a dynamic system, a random output is produced. The effect of the system is to modify the shape of the power spectrum between input and output. The output power spectrum is just the input spectrum by multiplied by the FRF magnitude squared (Figure 2.5).

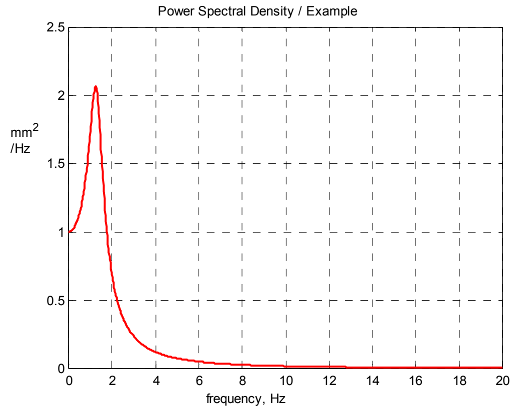


Figure 2.4 Power Spectral Density Function (PSD)

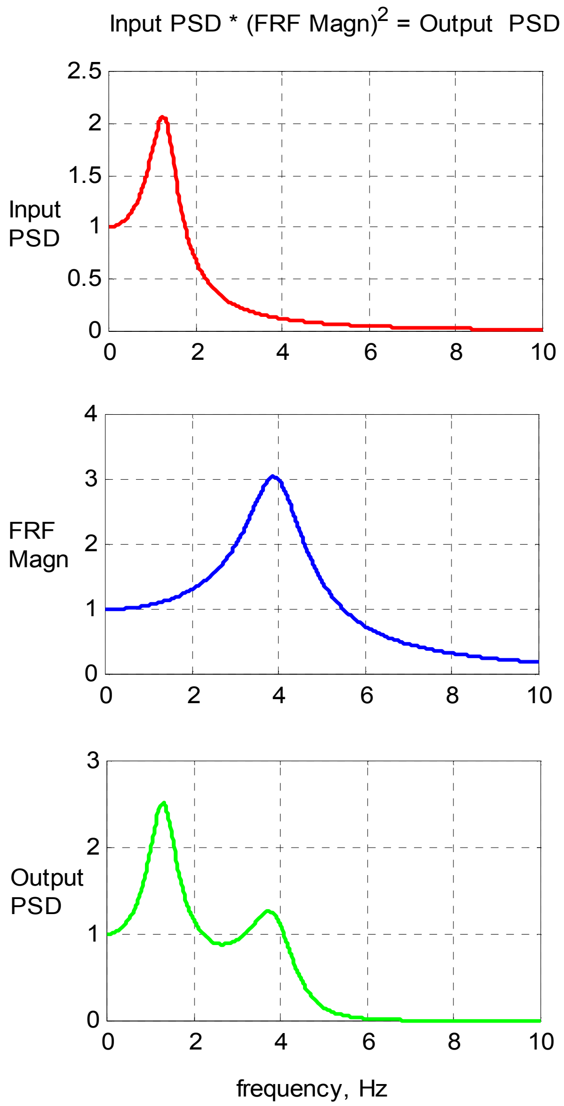


Figure 2.5 Obtaining Output PSD from Input PSD and FRF

The dynamic system essentially functions as a filter, to amplify sinusoidal components at some frequencies, and to attenuate them at other frequencies. For example, a building structure subjected to wind loading vibrates predominantly at the building's natural frequency. This is not because the wind necessarily flows at that frequency, but rather that the random wind may be considered to contain a wide band of frequencies, and the building is a filter that acts to focus the motion at one frequency.

When FRFs are used with power spectral density descriptions of random input, an important limitation is that the input must be Gaussian, i.e., its probability density function (pdf) must follow the normal (bell shaped) distribution. Statisticians often argue that many random signals have such a pdf, owing to the central limit theorem. However, engineers responsible for designing systems subject to random loading (especially road vehicles) understand that non-Gaussian random input is typical (but not 'normal'). Luckily, FRFs may be used in the case of non-Gaussian inputs as well. The catch is that the input power spectrum description must then be replaced by an explicit description of input waveform, i.e., a time history. Moreover, the output can no longer be calculated by simple frequency domain multiplication; instead, convolution operations must be performed. These complications are not serious, since analytical tools for performing such operations are well established. Likewise, the many of the concepts for random inputs apply: the system can still be considered as a filter, the input may be understood as a sum of sinusoidal components, etc. In the remainder of this discussion, the terms 'random' and 'arbitrary' will be used interchangeably, although their strict definitions are quite different.

When the system has multiple inputs and outputs (MIMO), there arises the possibility of cross-coupling: there may be a different functional relation for each input-output signal pair. For FRF models, these are often shown as a matrix (Figure 2.6). Each element of the matrix is a frequency response function, and the rows and columns of the matrix correspond to different inputs and outputs, respectively. An output signal is calculated using a combination of matrix multiplication and convolution.

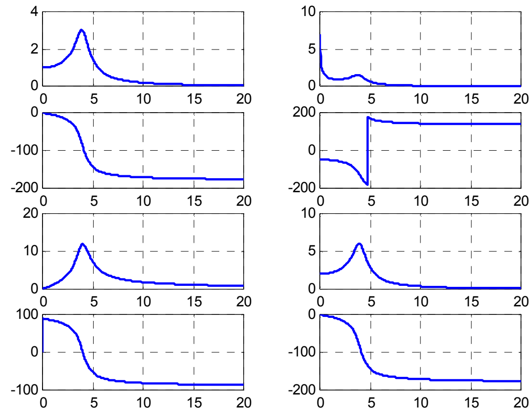
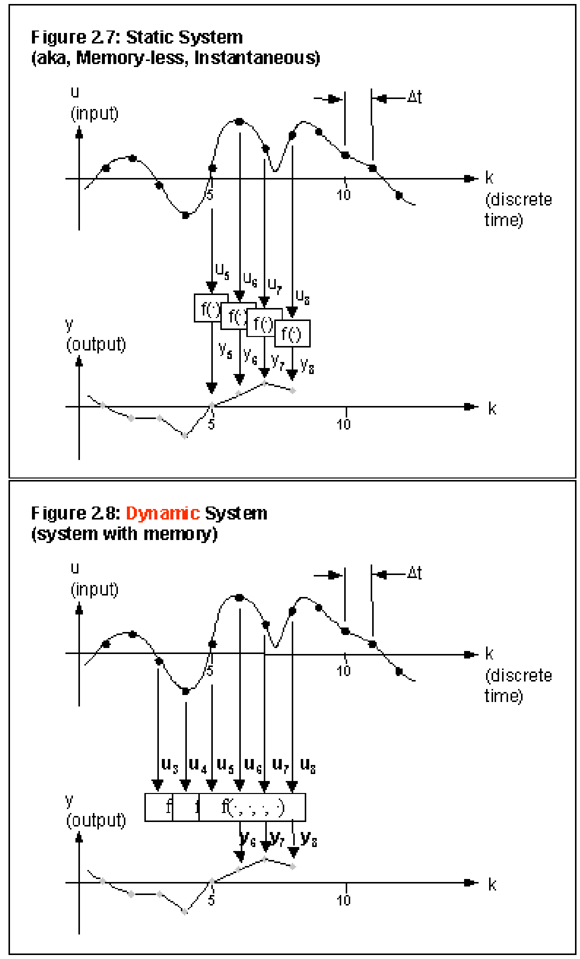


Figure 2.6 FRF Matrix

Frequency dependence is synonymous with concept of system memory [10]. A system has memory when its output at any time depends on the input at that same time, but also on past inputs and outputs. Equivalently, a system has memory if there is any phase lag between input and output. In physical systems, memory may be associated with certain types of energy storage elements. In whitebox modeling, memory is associated with integrators and state variables. Classic linear state space formulations using A, B, C, D matrices fall into this category. The concept of memory is useful because it ties together the integrator, state, and energy storage perspectives. However, it may be considered more broadly, to include systems that don't fit into these categories. Furthermore, memory focuses on the relationship between inputs and outputs, rather than on the internal operation of the system. This is particularly useful for blackbox modeling, where the concepts of state variables and energy storage aren't necessarily relevant.

When there are no memory effects, system output at any time depends only on the input at that same time. Such systems are called 'memoryless', 'instantaneous', or 'static'. Figures 2.7 & 2.8 demonstrate the relations between input and output, for memoryless (static) and memory (dynamic) systems. In this case, static does not mean motionless, or 'at equilibrium'; it means that the model equations for the static system can be used, even when the input is time varying. In whitebox terms, the system dynamics can be described without need for state variables. Likewise, the term 'dynamic' may used to describe frequency dependent systems. In this case, dynamic doesn't only mean 'varying in time'; it means 'having memory', or, 'requiring equations that use state variables'.



Figures 2.7 & 2.8 Static and Dynamic Relations between Input and Output Signals

Although these interpretations are somewhat esoteric, they are useful to describe systems and models more concisely. Specifically, the following substitutions can be used:

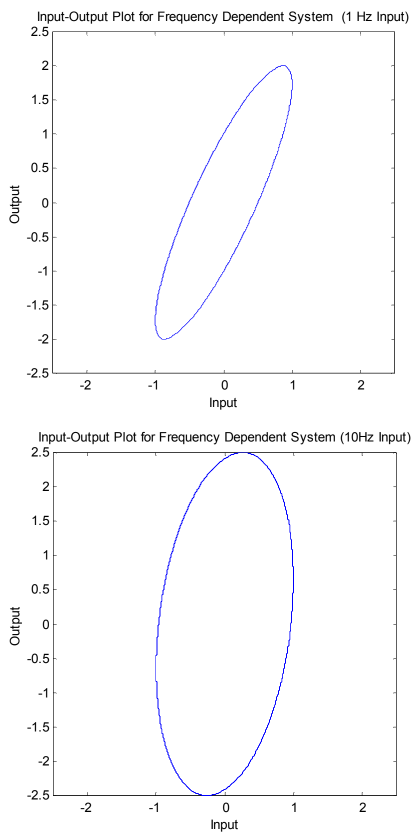
amplitude dependent, but not frequency dependent	static nonlinear
frequency dependent, but not amplitude dependent	linear dynamic (or just linear)
amplitude dependent and frequency dependent	nonlinear dynamic

Note that these terms may apply to both systems and blackbox models. For example, a nonlinear dynamic system may be modeled using static nonlinear equations. Such a model would ignore the memory effects.

The following paragraphs describe some of the limitations of static nonlinear and linear dynamic modeling methods when applied to nonlinear dynamic systems. This is generally a difficult task, because nonlinear dynamic behavior is difficult to describe. However, much can be learned by simple cross-application of the techniques, i.e.,

- static nonlinear models don't work well for linear dynamic systems
- linear dynamic models don't work well for static nonlinear systems

Consider first using a static nonlinear model for a linear dynamic system. The static nonlinear model is just a simple input-output curve (for a single input single output (SISO) system). However, when memory is present, the input-output curves may exhibit hysteresis (multiple values, or loops). For example, the input-output curves for sinusoidal excitation may yield an elliptical (Lissajous) pattern (Figure 2.9).



Figures 2.9 & 2.10 Lissajous Patterns

According to this representation, there are two output values for each input, and the correct output value at any time depends on the path or history of input. Another problem is that the size and shape of the elliptical plot may change with frequency. Thus, to model a frequency dependent system, it would be necessary to define a family of such curves. To make matters worse, the elliptical shape occurs only for sinusoidal inputs. Complex input waveforms, composed of sums of sinusoids, can yield extremely complex hysteresis patterns. Further complication arises when multiple inputs are involved. Try to imagine a family of surfaces in three dimensions, whose shape changes with frequency. Clearly, the static nonlinear modeling approach rapidly becomes impractical for linear dynamic systems.

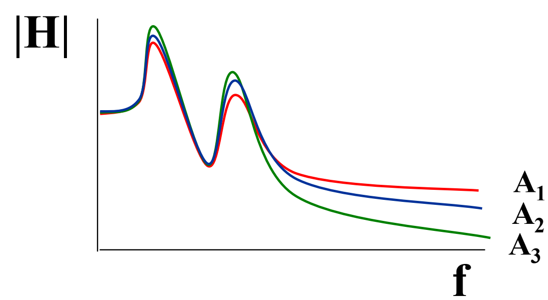


Figure 2.10 Family of FRFS (3 amplitudes)

Similarly, linear dynamic models are often inadequate for characterizing static nonlinear systems. Consider the simple case of a SISO system, modeled using an FRF. The amplitude dependence of the system means that there may be a different FRF for each input amplitude (Figure 2.10). Occasionally, one hears of

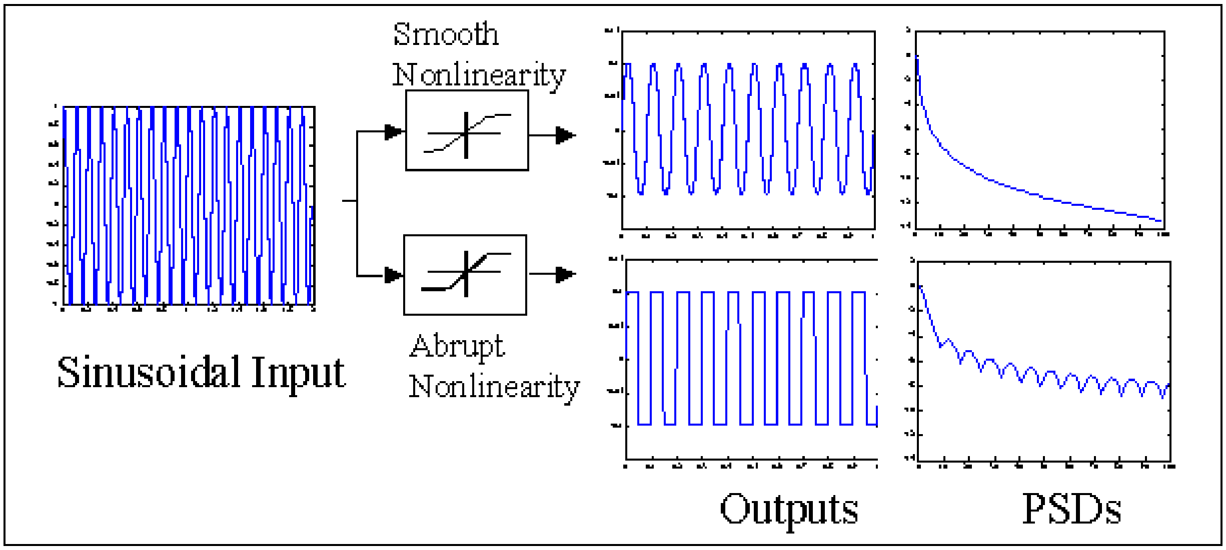


Figure 2.11 Generation of Harmonics by Smooth and Abrupt Nonlinearities

proposals to build a family of FRFs, each associated with a different amplitude, and to interpolate between them. Problems with this approach can be understood from the perspective of harmonic distortion. Whenever a sinusoidal input is applied to a nonlinear system, the output waveform is periodic, but not sinusoidal. Such a waveform may be considered to consist of one sinusoidal component having the same frequency as the input, plus additional harmonic components (i.e., at multiples of that frequency). A system with a 'smooth' nonlinearity may generate only few weak harmonic components in the output, whereas one with an abrupt nonlinearity generates strong harmonics over a wide frequency range (Figure 2.11).

Further problems arise when the input signal is composed of a few sinusoids at different frequencies. In this case, one effect of the nonlinearity is to generate frequency components at the sum and difference frequencies. So, if a sine-on-sine input having a 5 Hz component and a 1 Hz component is applied to an amplitude dependent system, the output waveform would have components at 1 and 5 Hz, but also at 4 and 6 Hz, the sum and difference frequencies. Harmonics and sum & difference frequencies ("cross-frequency" effects) are the reason FRFs are not viable for nonlinear systems: the output at 6 Hz depends not only on the input at 6 Hz, but also on the input at 1 Hz and 5 Hz. Or, the output at 6 Hz may be a harmonic for input at 1 Hz, 2 Hz, or 3 Hz. Even worse, the output component at 6 Hz could be the result of 4+ 2 Hz, or 8 -2 Hz, 10-4, etc., or all of these together. These effects cannot be quantified with a family of FRFs, since the FRF assumes that output at a given frequency is caused only by input at that frequency.

Clearly, the situation is even more complex for random input signals, since they consist of a large number of sinusoidal components. In this case, a family of FRFs (each member corresponding to a different amplitude) cannot be defined, because there is no single amplitude that characterizes the random input. Instead, the family would have to include a different FRF for *each input power spectrum*, which is completely impractical.

The preceding paragraphs showed how cross-application of the models fails for simple SISO systems. New problems arise when there are multiple inputs. In particular, there may be "nonlinear cross-coupling" effects. For example, an output resulting from 2 inputs may depend in some way on the *product* of those inputs. There is no provision for such terms in the FRF matrix framework.

Despite all these problems, many nonlinear dynamic systems can be effectively modeled as blackboxes. Two basic ways to do this are:

- Restrict the types of inputs that can be used
- Restrict the types of systems that can be modeled.

Examples of restricted inputs include amplitude limits and frequency limits. For example, linear dynamic methods are sometimes used for nonlinear dynamic systems, when amplitudes are small, say when determining vehicle NVH properties. Similarly, static nonlinear models may be used for nonlinear dynamic systems when large amplitudes are incurred, but the input are limited to a narrow frequency band, where the FRF doesn't vary much. Such is the case for events like vehicle maneuvering and handling, where frequencies are typically low.

Nonlinear dynamic systems can also be modeled if the system is of a limited class. For example, if the amount of nonlinearity is small, an FRF may be used as a least squares approximation. Alternatively, some systems can be characterized as a series combination of linear dynamic and static nonlinear parts. Models for these are known as Wiener or Hammerstein models [5], depending on how the two parts are connected (Figure 2.12). Volterra series models offer another way to handle nonlinear dynamics by restricting the class of systems [10]. They use higher order frequency response functions, where magnitude and phase are functions not just of frequency, but of several frequencies. For instance, consider a dynamic system with 'quadratic' nonlinear behavior: a whitebox representation for this system would include squares and product of two terms. The Volterra series model for this system gives sinusoidal output amplitude as a function of two frequencies (f1, f2). That is, the system generates sums and differences of two frequency components at a time, so there must be two independent variables for frequency. The problem with such an approach is that few nonlinear systems fit into the quadratic nonlinear category. A cubic nonlinearity may be more useful but this requires three frequency axes, which doesn't work well in practice.

This brief background has outlined some of the challenges and limitations of blackbox system modeling. It's important to remember that, despite these limitations, the linear dynamic and static nonlinear approaches cover a wide spectrum of applications. They deserve appropriate recognition, and should be included as part of any modelers toolkit. Variants of these conventional methods, to handle nonlinear dynamic systems, should be used cautiously: they may severely compromise the accuracy of the model, or they may be appropriate for only limited classes of systems or inputs. Of course, whitebox methods may provide a workaround in these situations. Alternatively, the Empirical Dynamics modeling method may be useful, as it enables accurate blackbox modeling for a very wide class of nonlinear dynamic systems.

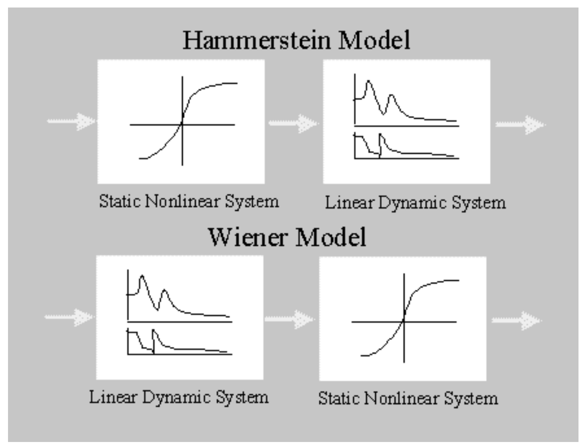


Figure 2.12 Wiener and Hammerstein Nonlinear Systems

3 Empirical Dynamics Models

Recent developments in dynamic system modeling methods now enable generation of accurate models from experimental data, with very few restrictions on the type of system or type of input [5,6,8]. These models can properly replicate amplitude dependence and frequency dependence together. The complications of multiple inputs and outputs, and cross-coupling, are easily handled. Moreover, the advances in computational horsepower allow such models to be built in reasonable time.

This convergence of modeling capabilities may be considered an important breakthrough. However, the nature of this breakthrough is subtle. It is the combination of multiple factors:

- blackbox
- amplitude dependence
- frequency dependence
- multiple input and output
- arbitrary input waveform
- applicability to a wide class of systems

that define the technique. As described in the Background section, conventional methods have been able to encompass many or most of these characteristics, but not *all* of them. A descriptive title for such a method should include all of these important aspects. However, even the most concise description, ‘nonlinear multi-input dynamic blackbox for arbitrary input and system types’, is a mouthful. The name Empirical Dynamics Modeling (EDM) has been coined as a convenient substitute.

ED models are based on neural network technology. A neural network is a computational paradigm (i.e., data structure and algorithms) originally developed to model the physiological function of brains [1,4]. Neural techniques have been primarily aimed at problems in pattern recognition, but they have also enabled improved solutions for nonlinear regression (curve fitting) problems, which includes nonlinear dynamic modeling. Following is a brief introduction describing how neural networks are used for this objective.

A neural network is constructed from fundamental units called neurons. A neuron takes a series of varying inputs u_k and multiplies them by constant weights w_k, sums these along with a constant bias term, and then applies the result to a nonlinear 'activation' function, to give an output value y. This process is shown in Figures 3.1 and 3.2.. The first figure presents the process in a conventional block diagram form, the second shows the same process as commonly depicted in the neural network literature. The activation function is usually a sigmoid (S shaped) function; for example, a hyperbolic tangent (Figure 3.3) . A neural network is constructed by connecting multiple neurons to the same inputs to make a 'layer', and by using the outputs of one layer as inputs to another layer. This structure is known as a multilayer perceptron (MLP) (Figures 3.4-3.5).

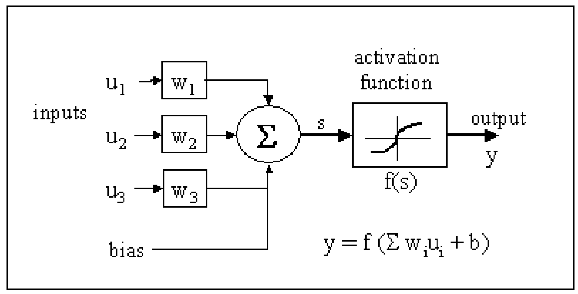


Figure 3.1 Neuron, Block Diagram

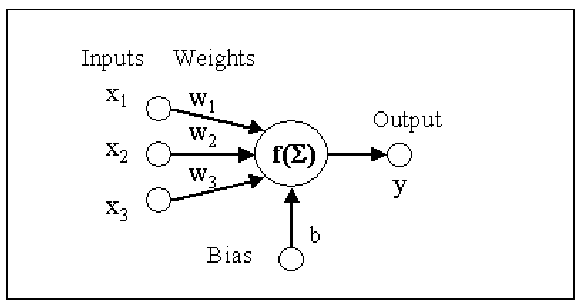


Figure 3.2 Neuron, Simplified Representation

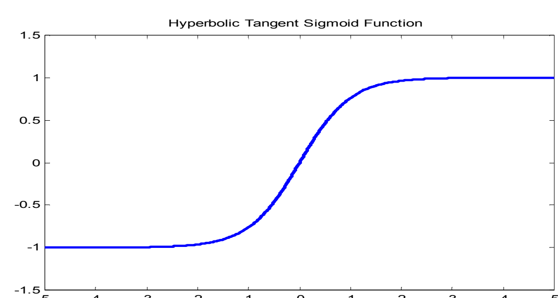


Figure 3.3 Hyperbolic Tangent Sigmoid Function

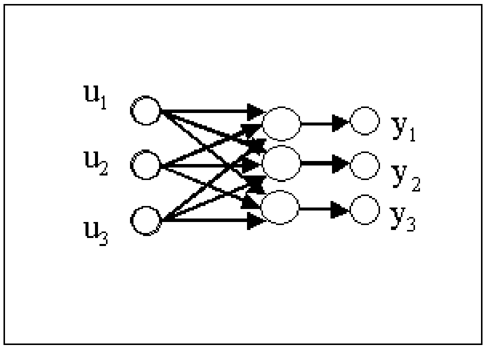


Figure 3.4 Neurons formed into a Layer

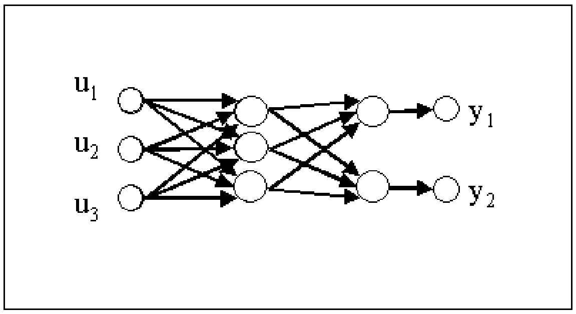


Figure 3.5 Multi-Layer Perceptron (MLP)

To model a dynamic system, inputs to the neural network are obtained from discrete time signals via tapped delay lines (Figure 3.6) [6]. The number of taps (i.e., number of current and past inputs and outputs) is chosen based on sample rate and on the characteristics of the dynamic system being modeled. Within the neural network, the number of layers and neurons in each layer are chosen similarly.

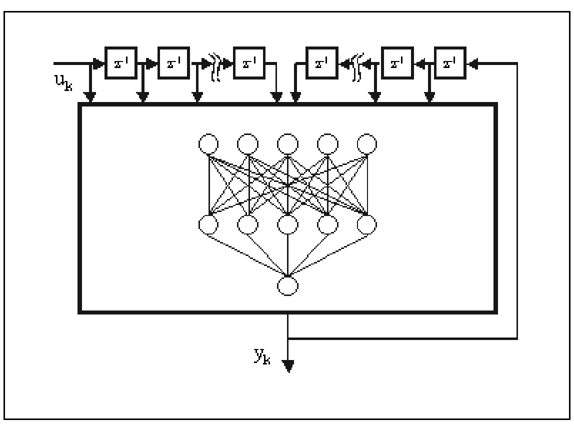


Figure 3.6 Tapped Delay Structure for Neural Net Input

Following this 'design' phase, training can begin. Training (or supervised learning) is performed by submitting a set of input data (signal samples) to the tapped delay inputs, and adjusting the neural network weights for each layer, until the neural net output approximates the corresponding output signals from the dynamic system, with minimum mean square error [4]. This adjustment process is called backpropagation, because the calculations are performed beginning with the neural net output and proceeding backward through the network.

The process of tapping the inputs and outputs, in conjunction with adjustment of neural network weights, can be understood as a nonlinear regression (curve fitting) problem [1]. Since the output for a dynamic system at any time depends on values of present and past inputs plus past outputs, a functional relationship of the following form is used [5,6,8]:

$$y_k = f\,(u_k, u_{k-1}, \ldots, u_{k-M}\ ,\ \ y_{k-1}, \ldots, y_{k-N}\ )$$

where k is a time index;
u's are inputs
y's are outputs
f is a nonlinear function, to be determined (the primary goal of the regression problem).
M is a parameter, = number of past inputs to include in the model. Ideally, this is an infinite number, to represent the fact that all past inputs affect the current output; practically, this is truncated at some point where the effect of past inputs becomes insignificant.
N is a parameter denoting the number of past outputs to use, the result of a truncation process similar to that for M.

The function f maps a space of M+N dimensions to a space of one dimension. The M+N dimensional space is populated by sample points, where one sample point represents a combination of M inputs and N past outputs, and has an associated function value equal to one value of system output. The goal of regression is to find one function f(˙) which approximately fits the sample values at these points.

An important facet of this process is the separation of the input and output signals into multiple sets, for training and prediction (or evaluation) [1]. A training set is a subset of the signal that is used for creating the model, as defined above. The prediction set is used to determine how well the model predicts outputs when supplied with 'new' inputs (i.e., not used for training). The use of these two sets is motivated by the following reasoning: The training process aims to fit a function to points from the training set, with minimum mean square error. It's possible with sufficient number of neurons to fit the training data perfectly (this property is called the universal approximation property). A perfect fit is not desired, however, for several reasons. One is that more computation time is incurred when more neurons are used. More importantly, a perfect fit will accommodate any and all measurement noise.

Figure 3.7 demonstrates how a simple but noisy data set may be fit with various curves (models). The left panel shows a linear fit, the right panel shows a perfect fit, and the middle panel shows an intermediate fit. Assuming the fit in the center panel is somehow optimal, the left and right panels are described as 'underfit' and 'overfit', respectively.

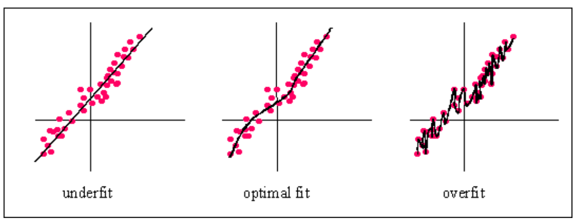


Figure 3.7 Multiple Fits to Noisy Data

Overfitting may lead to large errors when the neural net model is applied to a new data set (which is ultimately the goal of the modeling process). In the neural network jargon, this situation is called 'memorization'. The preferred behavior of the neural model is then described as 'generalization', where the model predicts outputs with low error when new inputs are supplied. Generalization may be facilitated by limiting the number of neurons to a small number (e.g., less than 50). This causes the regression function f to track the general trend of the input points while averaging out the noise. The ability of the neural network to generalize can only be assessed by testing it on data that was not used for model generation; this data is the prediction set.

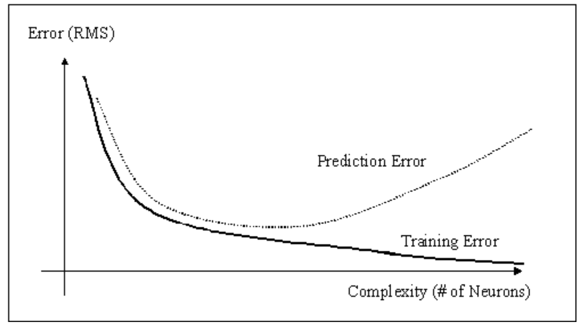


Figure 3.8 Training vs. Prediction Error

Errors for the training set and prediction set both change with the number of neurons, but in different ways (Figure 3.8). Training error generally decreases monotonically with more neurons, asymptotically approaching zero as the number of neurons becomes large. Prediction set error normally decreases for low numbers of neurons, then increases, so that there is an optimal number of neurons that gives the lowest prediction error. The process of finding this minimum is known as regularization, model selection, or complexity control [1]. Ideally, complexity control should be performed whenever a model is built. Moreover, a third data set (the validation set) should be used to perform this process. Practically, however, this optimization process may be extremely time consuming, as ever larger models take longer to generate, with diminishing returns in prediction error. It is often the case that excellent models may be obtained with a small but arbitrary number of neurons, and that optimal model selection is then not necessary.

These ideas about training and prediction errors and generalization are relevant for all blackbox modeling methods, not just for neural networks. They apply whenever the model order can be adjusted. For example, when polynomials are used for modeling, the model order is the degree of the polynomial. This can be adjusted to give optimal prediction error, just as the number of neurons can be chosen for a neural network.

Perhaps the most important lesson from this discussion is the idea that model performance should always be evaluated using a prediction data set; it should never be evaluated based on training data alone. All of the plots and other results presented in this report are based on a prediction data set.

Its important to understand that Empirical Dynamics method is more than just a neural network dynamic modeling (system identification) method. Other aspects of modeling may be considered integral to the technique, including test excitation, parameter adjustment, and interfacing to ADAMS, which are described in later sections. In addition, special proprietary enhancements have been employed to give the results shown in the following sections. The nature of these enhancements may be disclosed pending patentization.

4 Case Studies

This section will present case studies involving Empirical Dynamics models. These include models for 2 different shock absorbers and for a biaxially loaded rubber bushing, from a late model passenger car. Neural networks have been used for modeling shock absorbers since the early 1990's [2,3]. The results of the current case study confirm and extend these results. It's not apparent whether neural approaches for modeling rubber bushings have been investigated.

Case Study 1: Shock Absorber #1

Automobile shock absorbers (or dampers) are often modeled by fitting a curve to a plot of force vs. velocity. These plots are generated by applying a sine sweep displacement as input, and measuring the resultant force as output. Velocity is calculated from the known displacement history. The changing frequency of the sweep gives rise to varying velocities.

The current case study takes a slightly different approach, by using a *random* displacement signal as input. Of course, this is more realistic than a sine sweep, as the displacement profile for most roads is random, not sinusoidal. Another reason to use a random input is that it provides a richer content than a sine sweep, i.e., a wider variety of events are present in the signal. Training of the neural network model is best performed under these conditions.

The random signal for shock absorber excitation was generated as a discrete Gaussian time history, with power spectral density having magnitude proportional to 1/frequency^2. The exponent 2 gives a random signal with most of its energy at low frequencies, similar to the power spectrum of real road input. Moreover, the exponent 2 gives a flat velocity power spectrum, which is arbitrary but a reasonable choice for a velocity-based characterization.

The shock absorbers were tested in an MTS servohydraulic test rig. Use of hydraulic power with servo-control of displacement allows accurate application of the displacement profiles to the specimen, as the 'independent variable'. Force was measured as the 'dependent variable', using a strain gage load cell. Despite their accuracy, all servocontrollers have some response limitations. The actual displacement of the specimen may differ from the commanded value, in the form of small discrepancies in amplitude and phase at higher frequencies. These effects were circumvented in this investigation, by using the *actual* shock absorber displacement, measured with an LVDT, for subsequent model generation. Essentially, this procedure allows the dynamics of the test rig to be excluded from those of the shock absorber.

Velocity was calculated from the displacement signal, using a high order digital differentiator [7]. Such a differentiator reduces the high frequency phase errors of conventional first or second finite difference differentiators.

The force-velocity plot resulting from the random input shows a complex pattern (Figure 4.1). The pattern has two important characteristics: a curved 'backbone', and some hysteresis (width or thickness) around the backbone. These two parts coincide roughly with the presence of nonlinearity and memory, respectively.

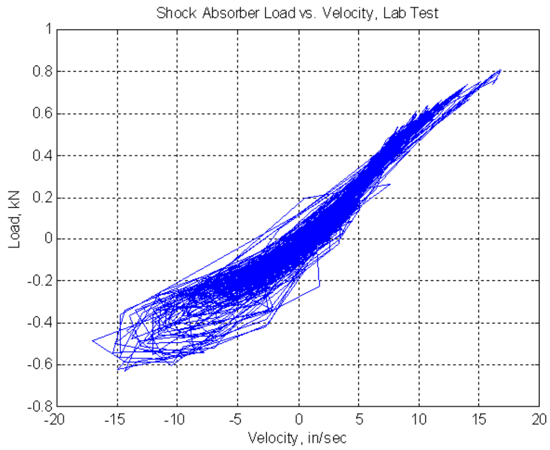


Figure 4.1 Shock Absorber #1 Force-Velocity, Original Lab Data

It is instructive to demonstrate how well conventional blackbox models work to replicate this pattern. For this purpose, a static nonlinear model was generated using a least squares polynomial curve fit, and a linear dynamic model was generated as an FRF. Each of these models was then used with velocity as input to predict the force. Results are plotted in Figures 4.2-4.3.

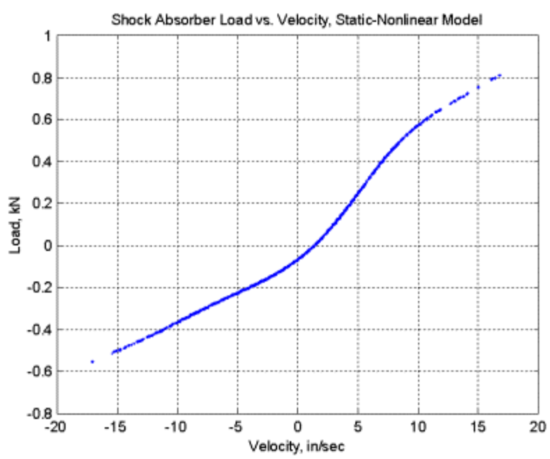


Figure 4.2 Shock Absorber #1 Force-Velocity, Polynomial Model

The polynomial model yields a plot which faithfully replicates the 'backbone', but fails to capture any of the original hysteresis. This compromise is expected for a static nonlinear characterization.

The FRF model gives a force-velocity plot that replicates the hysteresis to some extent, but fails to capture the curved backbone nature of the original data. This is the expected result for a linear dynamic model.

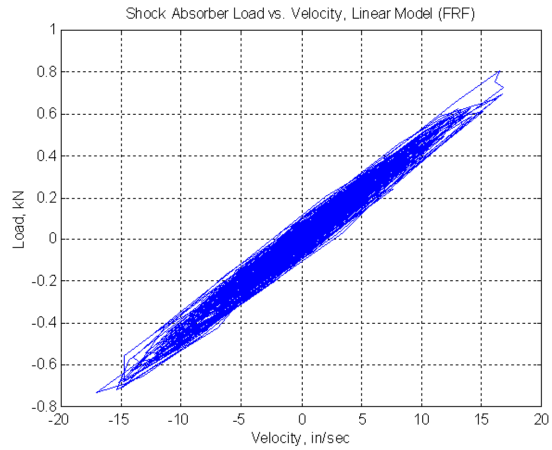


Figure 4.3 Shock Absorber #1 Force-Velocity, FRF Model

The Empirical Dynamics model for this shock absorber gives a force-velocity plot very close to the original lab data (Figure 4.4). It replicates both the backbone (nonlinear part) and the hysteresis (dynamic or memory part) of the original data.

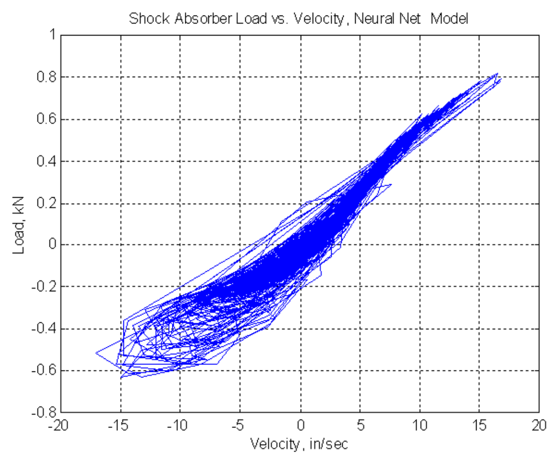


Figure 4.4 Shock Absorber #1 Force-Velocity, ED Model

These plots give a quick overview of the capabilities of the Empirical Dynamics method. Further detail of these capabilities is provided in the next section.

Case Study 2: Shock Absorber #2

For this specimen, the laboratory test was essentially the same as described above. However, in this case, displacement was used rather than velocity, as the independent variable for the model. This approach may seem unusual, since curves of force vs. displacement for a shock absorber are usually large loops, with no apparent 'curve fit' (Figure 4.5). In other words, there is no way that a static nonlinear curve (polynomial) could be used to model this relationship. The large hysteresis loops suggest that an FRF may work; however, the nonlinearity of this specimen isn't apparent from this plot, so blind trial and error is the only way to determine if an FRF will work.

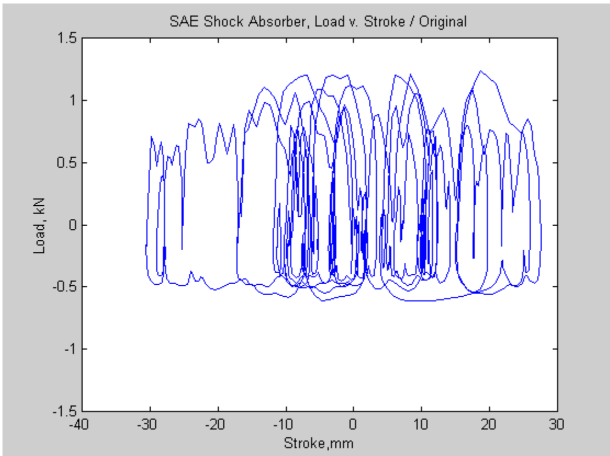


Figure 4.5 Shock Absorber #2, Force-Displacement, Original Lab Data

Use of displacement as input rather than velocity is motivated by a couple of ideas:

- A system with displacement as input and force as output is equivalent to system composed of a differentiator, followed by a system with velocity input and force output. A differentiator is just a linear dynamic system, so it doesn't add any appreciable challenge (i.e., nonlinearity) to the problem. If the differentiator is included in the blackbox, the model generation process should be able to account for it.
- By relying on the model generation process to handle the differentiation, problems that typically arise from numerical differentiation may be avoided. For example, conventional 'first finite difference' differentiation causes large phase shifts at high frequencies. Also, the same scheme yields amplitude errors, at high frequencies, owing to measurement noise. These problems could be overcome by using high order differentiators [7], with corrections to the gain function based on the noise spectrum (i.e., optimum filtering). However, a similar effect can be obtained by letting the model generation process take care of these problems automatically. This is possible because the optimal filtering approach and the FRF and ED models are all based on minimizing the mean square error.

Force-displacement plots for the 2nd shock absorber are shown, for the original lab data, the FRF model, and the Empirical Dynamics model (Figures 4.5-4.7). Clearly, the ED model replicates the original data most accurately.

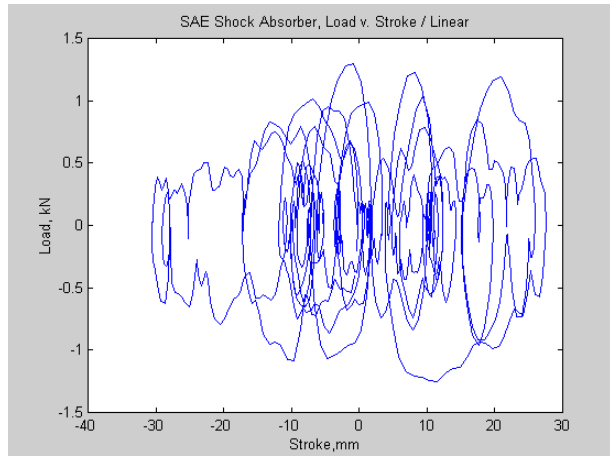


Figure 4.6 Shock Absorber #2, Force-Displacement, FRF Model

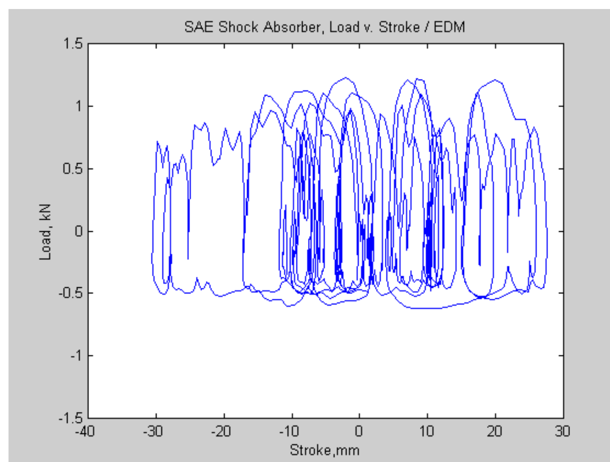


Figure 4.7 Shock Absorber #2, Force-Displacement, Empirical Dynamics Model

Note that the Empirical Dynamics model doesn't just replicate these 'on average'. Rather, every event is replicated closely. This can be seen more directly on a plot of time histories (Figure 4.9). A plot of original (laboratory) force overlaid with the model-predicted force shows that the ED model is predicting the signal very precisely at every point in time. An additional error time history overlaid on the same plot shows that the discrepancies are small. For comparison, time histories of original and predicted force are also shown for the FRF model (Figure 4.8). These plots show that the conventional models may introduce substantial errors in both amplitude and phase of the signals.

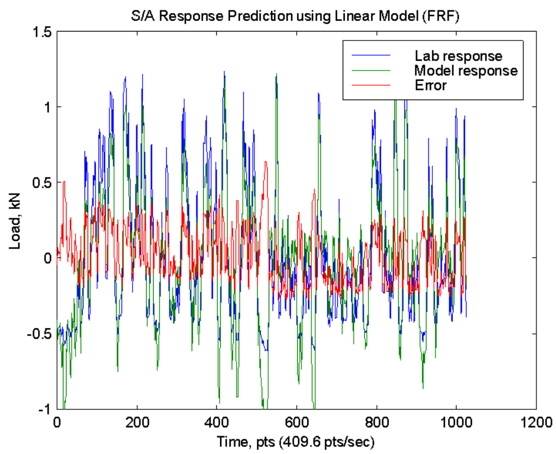


Figure 4.8 Shock Absorber #2, Time Histories, FRF Model

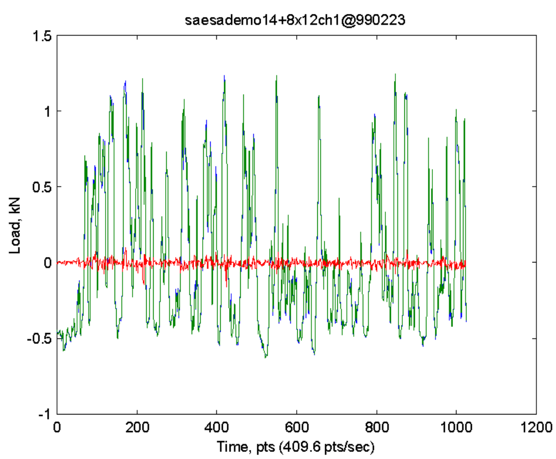


Figure 4.9 Shock Absorber #2, Time Histories, Empirical Dynamics Model (Legend in Fig 4.8)

Differences between the models can also be evaluated in the frequency domain. Fourier Transform (FFT) plots of load amplitude vs. frequency are plotted for the original lab data and each model (Figures 4.10-4.11); however, the FFT's of error (original – model) show most clearly that the Empirical Dynamics model is accurate over a wider frequency range. It's important to note again, not only the trend is represented, but the small detail as well. The FFT plot for the Empirical Dynamics model indicates that this model accurately predicts the force for a bandwidth of DC-100Hz. Another item to note is the spectrum of the error is very flat. This indicates that the modeling process has extracted essentially all of the information from the data; the error that remains is white, or completely uncorrelated with the input (or any other signal).

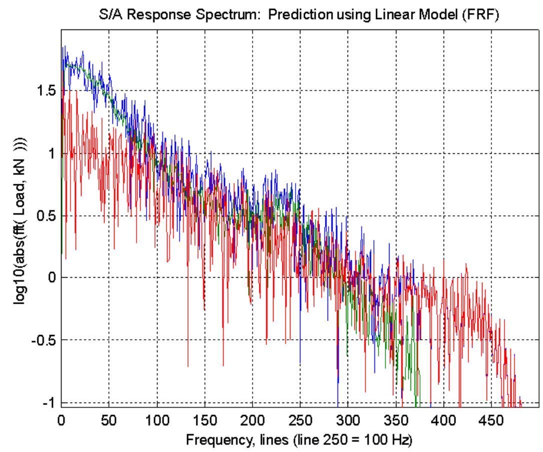


Figure 4.10 Shock Absorber #2 FFT's, FRF Model

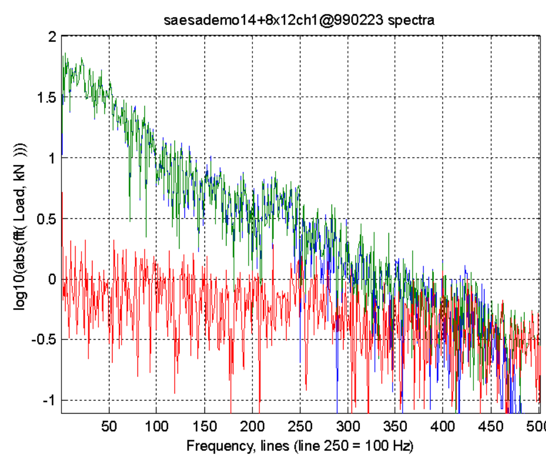


Figure 4.11 Shock Absorber #2 FFT's, Empirical Dynamics Model

The higher accuracy of the Empirical Dynamics model suggests that the specimen may have appreciable nonlinearity. However, there is no direct indication of this from any of the plots thus far. Therefore, velocity was calculated using a high order differentiator, and a force-velocity plot was generated (Figure 4.12). This plot is significant in that it indicates a very strong level of nonlinearity, as indicated by the abrupt change. Again, it must be emphasized that this velocity was not used to generate the model.

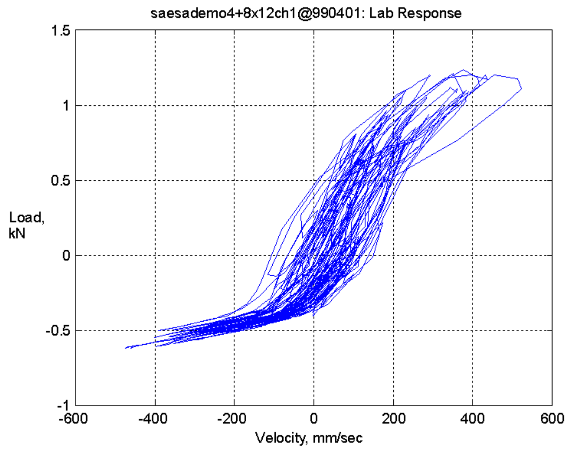


Figure 4.12 Shock Absorber #2 Force-Velocity, Original

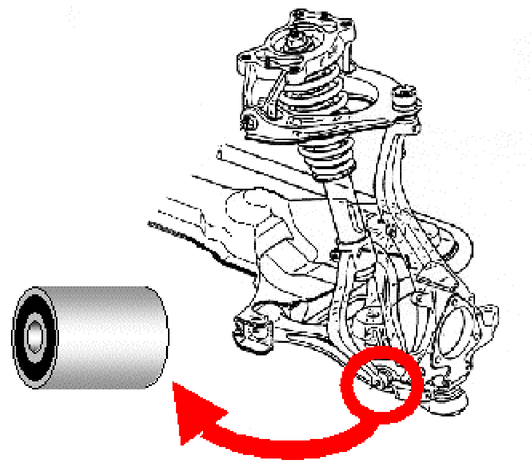


Figure 4.13 Bushing Specimen

Case Study 3: Biaxial Bushing

This case study demonstrates that the ED technique can be effectively used for multiple inputs, and for a specimen whose nonlinear character is different than that of a damper.

The specimen in this case is a rubber bushing from a passenger car front suspension, from the joint where the strut-fork connects to the lower control arm. The bushing consists of a rubber annulus sandwiched between two steel cylinders (Figure 4.13). The operating loading experienced by this bushing is primarily uniaxial, along the axis of the strut. However, for this case study, biaxial motion was applied along two orthogonal radial axes (x, z). The objective was to provide demonstration data for a multiple input nonlinear model, not to simulate the true bushing environment.

The bushing lab test was performed using an MTS multi-axial elastomer test rig (Figure 4.14). The inputs were displacements, and outputs were forces, for each of the orthogonal directions. The actual displacements were measured with LVDT's, and these were used as inputs for the models, to isolate the specimen from the servoloop dynamics (as explained earlier for the shock absorbers).

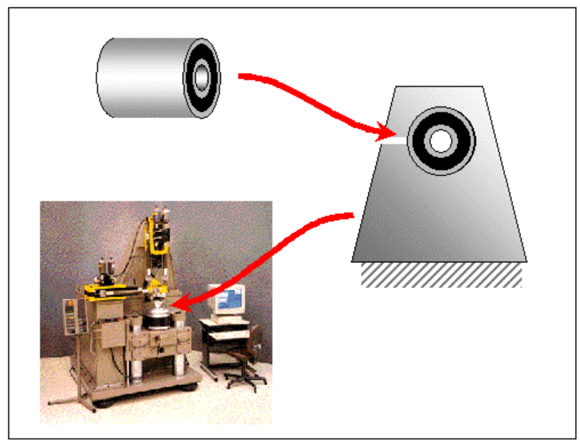


Figure 4.14 Bushing Adapter and Test Rig

The displacement signals were discrete Gaussian time series, with a power spectrum whose magnitude was proportional to 1/frequency. This spectral shape was selected to give most of the energy at low frequencies. Amplitudes were selected to give noticeable nonlinear behavior.

A special characteristic of the two random signal channels was that they were statistically orthogonal. This means the statistical correlation between the signals is essentially zero. Excitation having this characteristic is useful when generating FRF models: it enables the contributions of each input to be easily identified in either output.

The nonlinearity of the specimen can be seen from plots of force vs. displacement measured along matching axes (Figures 4.15, 4.16). For measurements along the z-axis, the overall pattern displays a dual-inflection shape, similar to a cubic function. Along the x-axis, the same tendency appears, although less pronounced. Both plots show some hysteresis. Note that the hysteresis here may be caused by cross-coupling, in addition to memory effects.

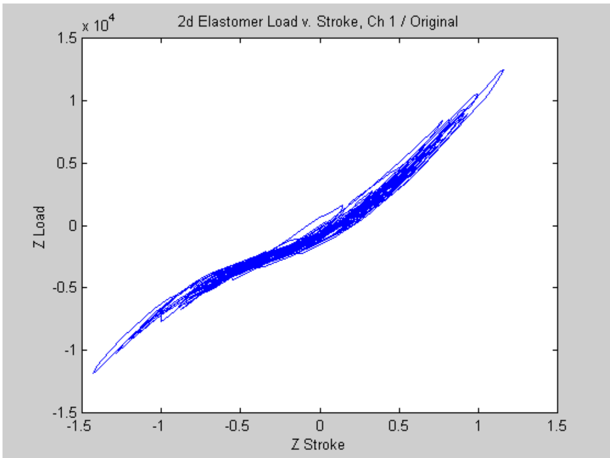


Figure 4.15 Bushing Force-Displacment, Original Lab Data, z axis

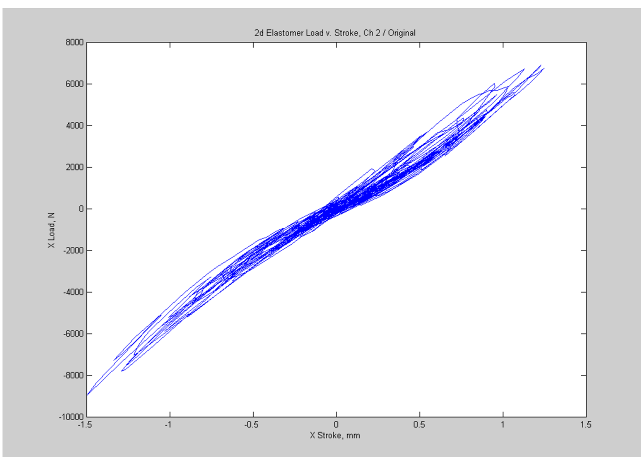


Figure 4.16 Bushing Force-Displacement, Original Lab Data, x axis

Using these signals, polynomial, linear and ED models were generated.

The linear model is a 2x2 matrix of FRFs, whose magnitude parts are plotted in Figure 4.17. Columns correspond to inputs, and rows correspond to outputs (responses). The main diagonal shows the direct coupling effects: the nearly constant FRFs indicate that the bushing is behaving like a spring, as expected. The fact that the upper left magnitude is larger than the lower right suggests some asymmetry is present in the bushing. The off-diagonal elements are non-zero, indicating that cross-coupling is present. This provides further evidence of asymmetry, since a radially symmetric bushing should generate only symmetric (even order) cross terms, which would not appear in the FRFs. Visual inspection of the specimen suggests the asymmetry may originate at the inner sleeve, which has a seam.

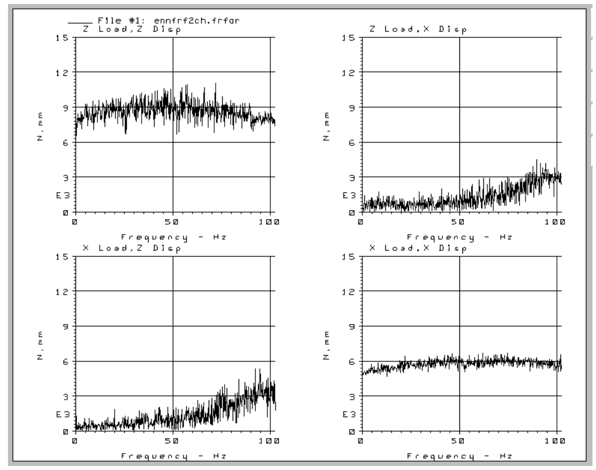


Figure 4.17 Bushing FRF Matrix

The static nonlinear model was generated as two independent polynomials (Figure 4.18), between force and displacement for matched axes. In other words, cross-coupling was ignored. This was done primarily for simplicity, as a model with cross-coupling would require representation by two independent surfaces in a three dimensional space (i.e., two axes are the inputs, and the third is one of the outputs). This simplification is justified by the fact that cross-coupling is small, as shown by the FRF.

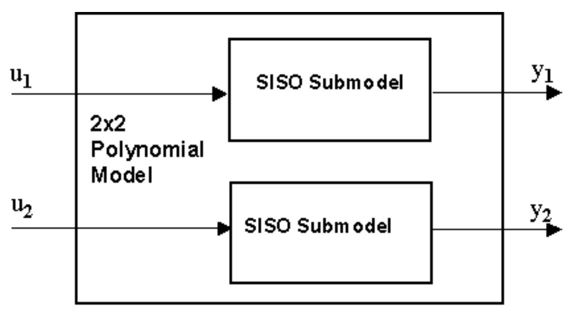


Figure 4.18 Uncoupled Polynomial Model

The ED model for the bushing was generated as a pair of independent multiple input, single output (MISO) units. That is, each output may be affected by both inputs, but the outputs may be considered independent of each other. (Figure 4.19)

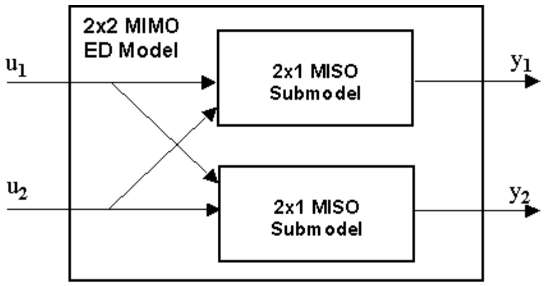


Figure 4.19 Double MISO Structure for Empirical Dynamics Model

The models were evaluated using the same techniques as for the shock absorbers. Plots of force vs. displacement, for the z-axis, show that the polynomial model predicts the 'backbone', but not the hysteresis (Figure 4.20), and that the FRF predicts the hysteresis but not the backbone (Figure 4.21). (Plots for the x-axis show similar results).

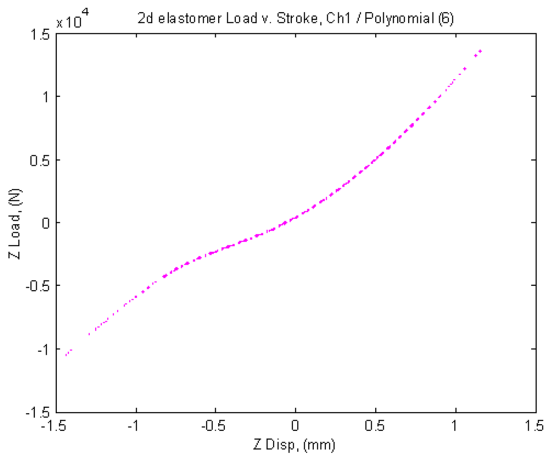


Figure 4.20 Bushing Polynomial Model, z axis

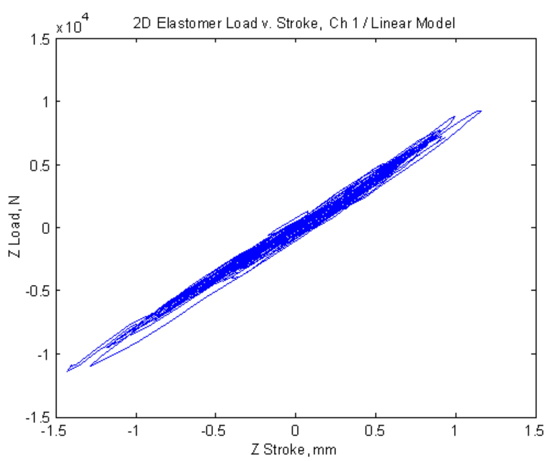


Figure 4.21 Bushing Force-Displacement, FRF Model, z axis

The ED model, however, gives a plot nearly identical to the original lab data (Figure 4.22).

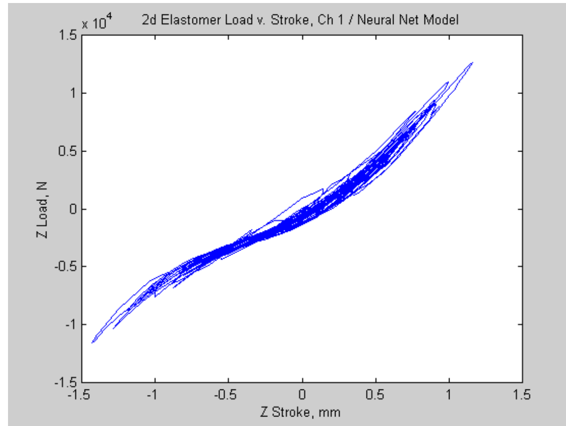


Figure 4.22 Bushing Force-Displacement, ED Model, z axis

Figure 4.15 is replicated here for direct comparison with Figure 4.22:

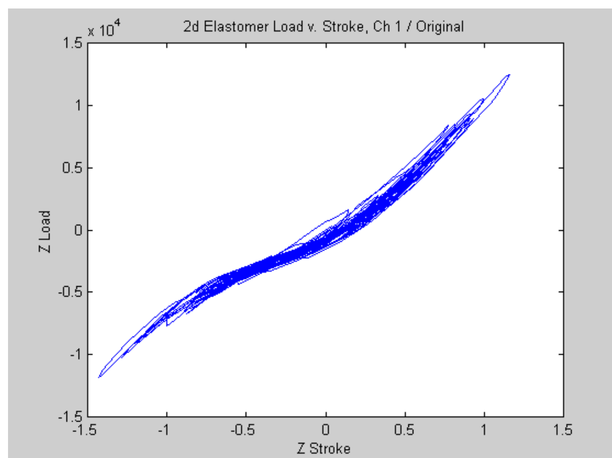


Figure 4.15 Bushing Force-Displacement, Original Lab Data, z axis

Time history and FFT plots likewise show that: the ED model has lowest errors in both domains (Figures 4.24-4.29; plots are shown for the z-axis only; the x-axis plots are similar).

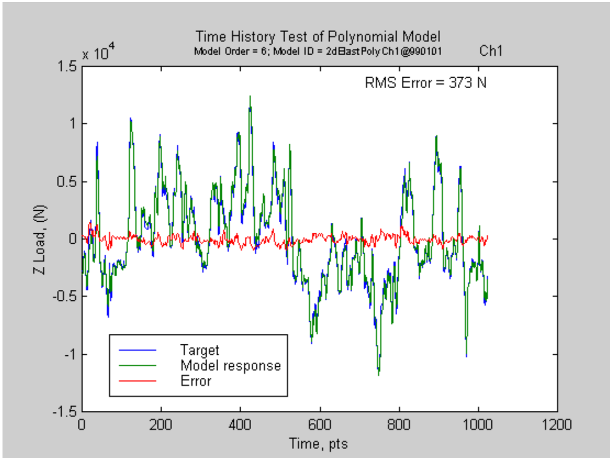


Figure 4.24 Bushing Time Histories, Polynomial Model, z axis

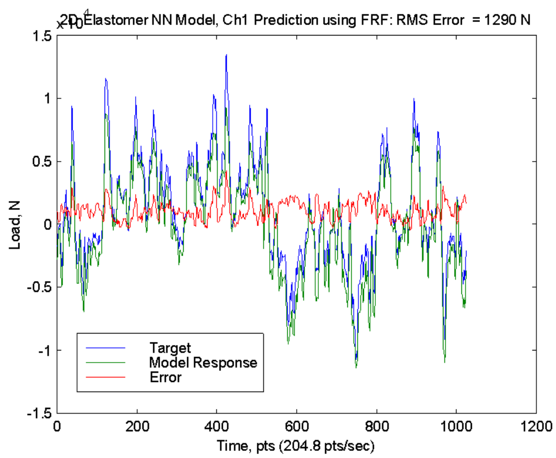


Figure 4.25 Bushing Time Histories, FRF Model, z axis

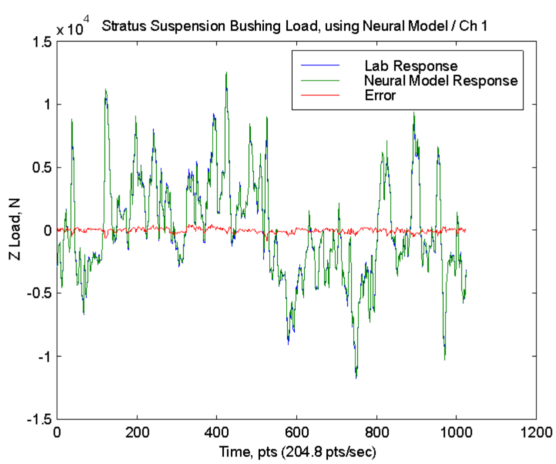


Figure 4.26 Bushing Time Histories, ED Model, z axis

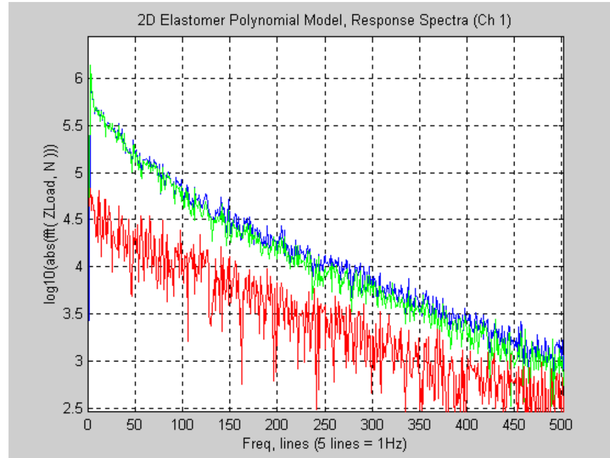


Figure 4.27 Bushing FFT's, Polynomial Model, z axis

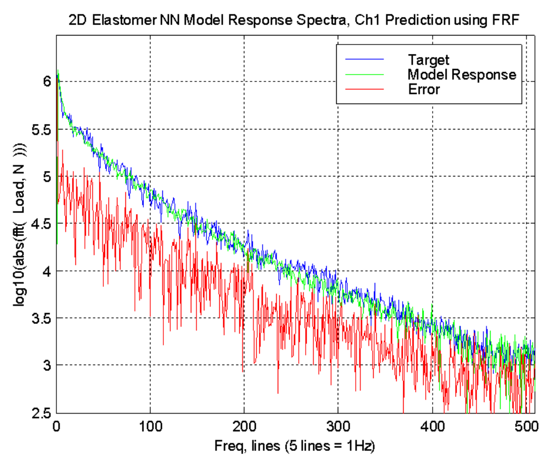


Figure 4.28 Bushing FFT's, FRF Model, z axis

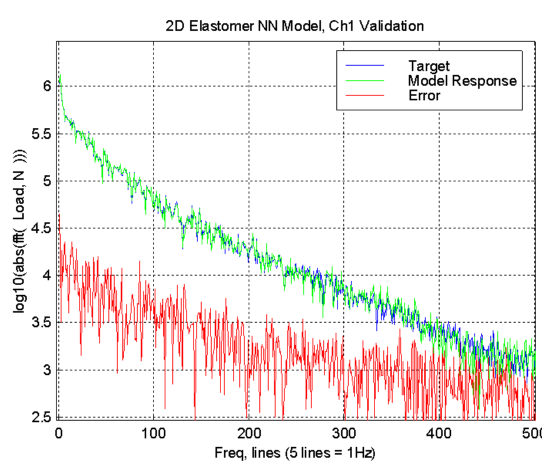


Figure 4.29 Bushing FFT's, ED Model, z axis

The preceding case studies have shown the power of the Empirical Dynamics technique for modeling vehicle components with one or two inputs. Other case studies (to be reported) demonstrate that this technique can be applied to 3 or more inputs. In one example, the lateral strain at a vehicle suspension ball joint, which is influenced by vertical, lateral, and longitudinal road inputs, was predicted using those three inputs, with error reduction of 40-50% over FRF models. The most significant aspect of this case study was that the specimen included not only the ball joint, or the suspension, but also the entire vehicle along with 12 channels of multi-axial road simulation test equipment (Figure 4.30).

From these examples, it seems reasonable that the technique may be extended to other types of vehicle components, including tires, engine mounts, exhaust system hangers, etc. Additional applications may include non-vehicle applications, including biomechanical systems, electrodynamic systems, and fluid dynamic systems.

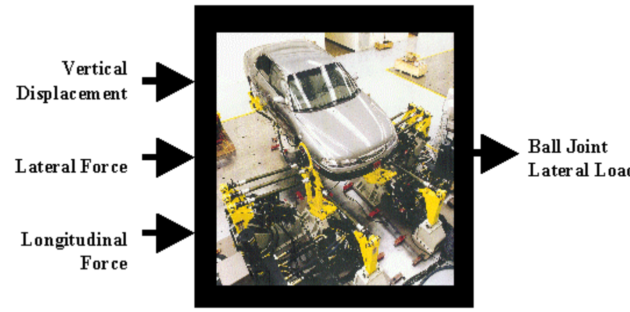


Figure 4.30 3x1 Blackbox Model for Ball Joint Strain

5 EDM Benefits and Limitations

The preceding Case Studies have shown by example how the Empirical Dynamics method yields models with higher accuracy than conventional blackbox methods. This section provides a summary of benefits and limitations of the Empirical Dynamics approach

Benefits of EDM include:

- Significant improvements in accuracy
- Ability to model a wider range of systems and components, with just one technique
- Simpler and faster modeling

Each of these is elaborated below.

- -

Benefit 1: Significant Improvements in Accuracy

Empirical Dynamics models are more accurate than conventional blackbox models. In many cases, Empirical Dynamics models will far outperform linear dynamic models and static-nonlinear models (e.g., polynomials). This depends of course on the degree of nonlinearity and/or memory in the system. Empirical Dynamics models are most beneficial when these effects are appreciable. (That is, if the system is already highly linear, EDM won't provide much improvement over an FRF). Evidence for this benefit was provided for numerous specimens in the Case Studies.

Empirical Dynamics models can be more accurate than whitebox models. Whitebox models often oversimplify complex physical phenomena. For example, a whitebox model for a 'friction element' may be represented in terms of a simple (static or dynamic) coefficient, obtained from a table in a design handbook. This value is subject to wide variability (related to surface finish, contamination, relative velocity, etc) and doesn't account for transitions from static to dynamic (e.g., stiction). By contrast, an Empirical Dynamics model of a system with physical friction can include these effects, in proper proportion. The result is a more accurate model overall.

Empirical Dynamics models are 'self-validating'. The accuracy of an Empirical Dynamics model is assessed as part of the model generation procedure. By splitting the data into training and validation sets, the performance of the model is easily assessed, without any need to return to the lab. Another important aspect of validation: since Empirical Dynamics models are generated and validated using random data, they automatically encompass a wide range of operating conditions. Whitebox models are often validated using simple waveforms (sinusoidal, step); these results may be difficult to extrapolate to more general conditions. Of course, whitebox models can also be validated using random signals, but this may require significant computational resources.

\- -

Benefit 2: A Wider Range of Systems, Components, and Signal Types

Simple components or complex systems can be handled with the same method. The only requirement is to have input and output signals from the specimen being modeled. Considerations re: actual complexity of the specimen, in terms of number of moving parts, degrees of freedom, etc., can be often be ignored.

The Empirical Dynamics approach may be used for a wide range of applications. That is, Empirical Dynamics modeling may be applied not just to mechanisms, but also to hydraulic, electrical, thermal, optical, digital/software, & other systems.

Empirical Dynamics models can be used when whitebox models aren't available. For many physical systems, there are no whitebox methods that can accurately predict dynamic behavior. For example, components that include elastomers/rubber bushings, fabric, biomaterials, etc., are sometimes modeled using the finite element method (FEM). The FEM models don't include provisions for modeling significant damping or hysteresis behavior in these components. Empirical Dynamics models may be the only choice in this case.

For other components, there may be special whitebox methods available, but only at prohibitive cost ($ or expertise). For example, fluid-filled or fluid-dynamic components may be modeled using a computational fluid dynamics (CFD) modeling package. If this tool is used only infrequently, the total cost to build a model may far exceed its capital outlay.

The Empirical Dynamics method allows use of alternative signal types: For example, dampers (shock absorbers) are often characterized in terms of force vs. velocity. With Empirical Dynamics modeling, accurate damper models may be generated in terms of force vs. displacement. Essentially, the Empirical Dynamics model includes the differentiator that converts displacement to velocity. This helps prevent errors that may be introduced by numerical differentiation schemes, or from velocity transducers which may be noise sensitive.

\- -

Benefit 3: Simpler and Faster Modeling

Empirical Dynamics modeling enables simpler model construction. In particular, generation of Empirical Dynamics models may be automated, in many cases. This happens in part because of the generic nature of Empirical Dynamics modeling, i.e., only input and output signals are required; internal specimen details can be ignored. Automation may be applied most beneficially in scenarios where all specimens are similar (for example, shock absorbers tested in an MTS shock absorber test rig).

Empirical Dynamics models may be constructed faster. In situations where a test rig, specimen, and related instrumentation are readily available, it may be more expedient to generate a model by the Empirical Dynamics approach, even if a whitebox method is available. Note that the test equipment and specimen are often available, to validate whitebox models. The benefit of Empirical Dynamics modeling is to consolidate validation and model generation into one procedure.

At this stage it is informative to present some timing benchmarks for the main Empirical Dynamics processes.

Specimen	**Lab Test Duration**	**Training Duration**
Shock absorber	1 minute	1 hour
Rubber bushing	2 minutes	2 hours for a 2x2 model

Some conditions: all calculations are based on using a 450 MHz Pentium II processor. Note that the lab test times do not include installation of the specimen, servo-loop tuning, amplitude adjustment, and other preliminary setup. Model calculation times do not include miscellaneous preliminary processing, including signal channel rearrangement, offset removal, and others. Most importantly, the calculation presumes the design for the neural net (including number of layers and number of neurons) has already been finalized. This last task may be the most time consuming part of the Empirical Dynamics modeling process; on the other hand, it may seldom be necessary, for many test scenarios. For example, when a series of Empirical Dynamics models is generated, for specimens of similar behavior, it will generally be necessary to design the network only once. Thus, network design may be pre-defined, for some specimen types.

Furthermore, a neural net design may be remarkably robust. Perhaps one of the most surprising discoveries in the development of the Empirical Dynamics method was that *the neural network designed for use with the shock absorbers could be used, without significant change, to the rubber bushing*. This is surprising because the two systems have significantly different mechanical behavior, as shown by their force-displacement plots.

<u>Empirical Dynamics models execute faster</u>. Empirical Dynamics models are usually represented as closed-form algebraic equations, which are quickly evaluated by 'turning the crank'. By contrast, whitebox models often yield differential and algebraic equations (DAE's), which are often algebraically complex and may require considerable iteration to solve.

At this time, benchmarks for EDM execution are not available, for operation with ADAMS. Standalone operation of EDM, where output signals are calculated directly from predefined inputs, has been measured. An ED shock absorber model, with a sampling rate of 204.8 Hz (giving model bandwidth ~ 80 Hz) has most recently been measured at 250 times faster than real-time. In other words, 10 seconds of input signal, equivalent to 2048 sample points, executes in approximately 0.04 seconds (with the CPU described above).

Note that for Empirical Dynamics models, <u>generation time may in many cases be offset by execution time</u>. For the shock absorber example, two hours of processing (1 for generation, 1 for execution) effectively buys 250 hours of simulation. (A direct comparison with whitebox methods cannot be made, as a whitebox shock absorber model has not been investigated. Clearly, this will be a worthwhile investigation).

- -

Limitations of the Empirical Dynamics approach include:

- Test rig & specimen are required
- Specimen parameters cannot be adjusted.
- The model requires a large number of coefficients.
- The number of inputs is limited.

- -

Limitation 1: A Test Rig & Specimen are Required.

Being a blackbox technique, Empirical Dynamics requires a specimen from which signals can be obtained. However, test rigs and specimens are often available, as needed for validation of whitebox models.

A future application of the Empirical Dynamics approach includes "whitebox substitution", where a time consuming whitebox subsystem model is replaced by an Empirical Dynamics equivalent. In this case, the Empirical Dynamics model is generated directly from the whitebox model, with no need for a test rig or specimen.

- -

Limitation 2: Specimen Parameters Cannot Be Readily Adjusted.

Parameters refer to specimen variables that may be adjusted as part of the specimen design process. For example, parameters of a tire might include:

- geometry (diameter, width, sidewall height, tread pattern)
- mechanical properties (viscoelasticity of the rubber, anisotropic flexibility of the rubber/cord matrix)
- pressure

Whitebox models are usually configured to allow adjustment of these parameters in the model.

Blackbox models, by ignoring the internal details of the physical system, preclude the ability to manipulate various parameters of that system. Nevertheless, there are multiple counterpoints that bear mention:

Complete adjustability is often not required: A dynamic vehicle model may be needed to study body loads incurred while traversing a rough road. Tire models for this vehicle may be whitebox or blackbox. For the purpose of modeling suspension-to-body loading, the tire parameters (listed above) are not needed. Instead, a single unchanging tire may be used; it can be modeled using a blackbox. The benefit of the blackbox is improved accuracy and execution speed.

Complete adjustability has serious economic ramifications: A whitebox model with full adjustability introduces excessive complexity into the modeling process, and with it much longer computation time.

Limited parameter adjustments are possible with Empirical Dynamics models. Two examples of these include 'extrinsic conditions' and 'model stretching'.

Extrinsic conditions refer to the use of multiple models, each corresponding to a different test condition, which is constant or slowly varying. For example, to account for thermally induced system variation, a series of Empirical Dynamics models is constructed, each corresponding to a separate temperature (Figure 5.1). These are combined into a single file structure (Figure 5.2). During execution, temperature is used as an input to this series of models, to select the correct model. Interpolation between temperatures could be performed with minimal difficulty (Figure 5.3). This same approach can be used for conditions such as pressure, preload, etc., but also for design variables. For example, a bushing could be measured using several diameters, rubber compositions, etc., and interpolation could be used to provide a wide range of parameter adjustment.

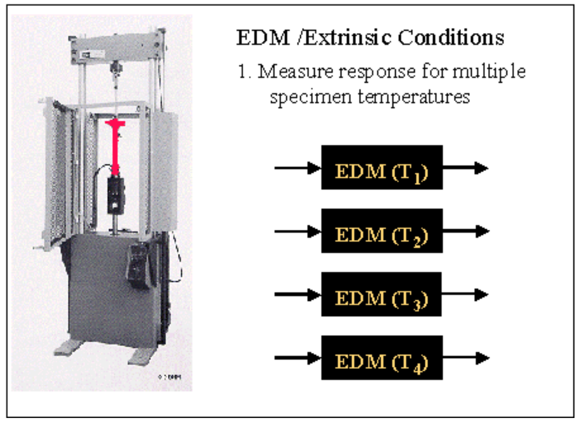


Figure 5.1 Extrinsic Condition, Measurement of Multiple Conditions

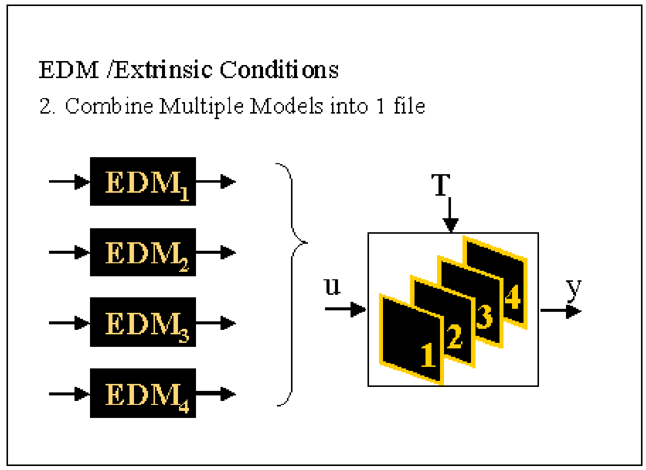


Figure 5.2 Extrinsic Conditions, Model Combination

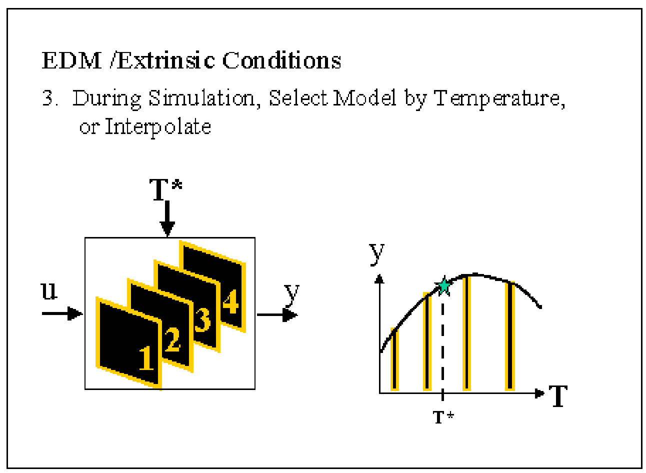


Figure 5.3 Extrinsic Conditions, Selection or Interpolation

Model stretching refers to the application of gain factors to input and output signals of a model, and to the time base as well. Whereas a measured input-output plot shows actual system behavior, this plot can be 'stretched' in the horizontal or vertical directions to perform a 'what if' evaluation. This may actually be more straightforward than adjusting physical/ whitebox parameters; for those systems: it's not always clear which parameter provides a stretching effect, and it's very likely that a physical parameter adjustment one way causes unintended side effects elsewhere in the system.

Moreover, model stretching adjustments may have actual physical counterparts. A case in point is that of a coil spring, with nonlinearity appearing when the coils 'bottom out' at large compressive displacement. As shown in Figure 5.4, this system may be modeled simply as a nonlinear force-displacement relation. By stretching the model along the displacement axis, the nonlinearity is encountered at a different displacement, but at the same force (Figure 5.5). This is equivalent to adding coils to the spring. If the model is stretched along the force axis, the nonlinear transition occurs at the same displacement (Figure 5.6). This is equivalent to changing the coil diameter (and leaving the wire diameter intact).

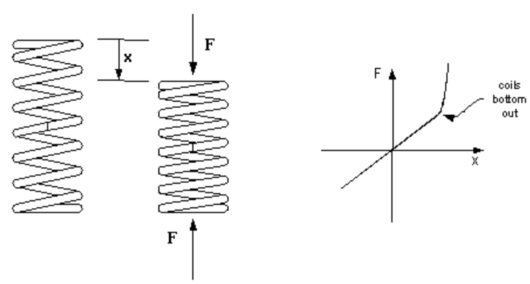


Figure 5.4 Coil Spring

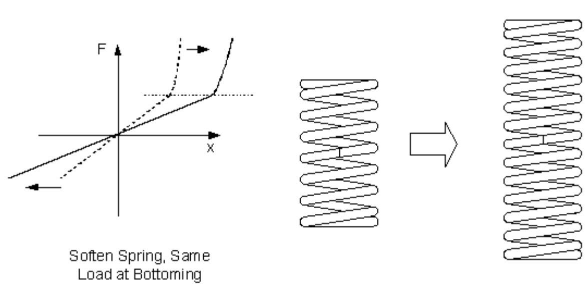


Figure 5.5 Coil Spring, Empirical Dynamics Model Stretching in Displacement

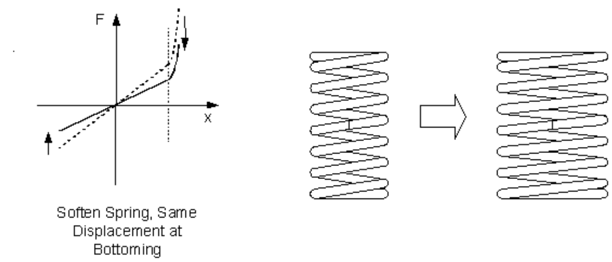


Figure 5.6 Coil Spring, Empirical Dynamics Model Stretching in Force

A similar capability is available by changing the time base of the Empirical Dynamics model (i.e., the sample rate used for the input and output histories). For a resonant mass-spring-damper system, changing the time base is equivalent to changing the natural frequency of the system, without changing the output/input gain. Note that in the whitebox realm, this would require modification of both the mass and the stiffness, to achieve the same effect. Another example is that of a damper: changing the time base is equivalent to changing the kinematic viscosity (assuming piston inertia is negligible), which can be further connected to thermal changes.

Empirical Dynamics modeling may be applied to dimensionless characterizations. An Empirical Dynamics shock absorber model doesn't allow direct adjustment of viscosity, orifice size, spring stiffness, etc., and so it isn't particularly useful for shock absorber design. However, even a whitebox damper model has limitations, particularly in the fluid dynamic coefficients, which are typically quasi-static 'cookbook' values. These may be inadequate for assessing the high frequency phenomena associated with closing valves. A possible improvement for the future is to use the Empirical Dynamics approach to model the only the fluid dynamic behavior, rather than the entire shock absorber. For example, a pressure coefficient may be modeled as a function of a 'dynamic' Reynolds number and valve position, normalized to some reference length, e.g., orifice size. This would allow the unsteady fluid dynamics to be properly modeled. Note that such a characterization would entail construction of a special fluid dynamic test rig. Similarly, an Empirical Dynamics model may be constructed for just the friction behavior. Once these 'subcomponent' Empirical Dynamics models are obtained, they can be integrated with whitebox spring and inertia components to provide improved accuracy, while retaining the adjustability needed for design optimization.

- -

Limitation 3. Empirical Dynamics Models May Require Many Coefficients

The number of coefficients represents the model size, i.e., the amount of information required to specify the model. Large model size has historically had economic ramifications (connected to disk & RAM capacities, for example). Likewise, large models may mean more complexity from a user perspective, especially if these numbers have to be interpreted, or manually determined.

With large numbers of neurons, an accurate Empirical Dynamics model may contain hundreds of coefficients for each combination of input and output.

However, <u>large is relative</u>. Compared to whitebox parameterizations, Empirical Dynamics models may seem large; compared to modern disk storage and RAM capacities, they are minute.
Similarly, <u>large is misleading</u>. The coefficients of a nonlinear blackbox model do not translate to additional complexity for a user (i.e., to identify, understand, or modify).

- -

Limitation 4. Empirical Dynamics Models are Limited to a Few Inputs

At this time, Empirical Dynamics models with three or fewer inputs have been studied. Model generation time for a three input system is excessive (> 2 days). Research and development efforts to reduce this duration are currently underway.

6 Interfacing Empirical Dynamics Models to the ADAMS Environment

The ADAMS Solver has been designed for optimal solution of equations in state variable form. This conflicts with several aspects of the Empirical Dynamics paradigm. Primary differences include:

- Empirical Dynamics models use a fixed sample rate, or time base. The ADAMS Solver typically uses variable time steps.
- Empirical Dynamics models require information about past inputs and outputs. The ADAMS Solver does not normally retain a history of such information; instead, it stores the necessary 'history' information as state variables.
- The ADAMS Solver supplies inputs to a component one at a time, and these may be trial values, whose convergence status is indeterminate. If past inputs and outputs are retained (in a buffer) for use with an Empirical Dynamics model, the trial values must be identified and excluded from the buffer.

To negotiate between these different environments, a prototype interface between ADAMS and Empirical Dynamics has been developed. Called the ADAMS/EDM Socket, this interface works within the ADAMS environment to perform the following functions

- sample rate conversion (via interpolation)
- determination of convergence status for past inputs
- buffering of past inputs and outputs.

Preliminary tests on the prototype socket have demonstrated its basic feasibility, and further tests are underway.

Future development plans for the Socket include provision for the following features:

Physical interfacing. This includes component specifications that aren't directly required for Empirical Dynamics model generation. For instance, the length of a shock absorber is needed within the ADAMS environment, but it isn't needed for Empirical Dynamics model generation. A related issue is matching of coordinate systems: the X,Y,Z axes used for a lab test must be properly identified and matched in the ADAMS environment.

Adjustments. As defined in the earlier Section, adjustments via extrinsic conditions and model stretching alleviate some of the limitations of the blackbox approach. These features best implemented at the socket level.

Limit checking: Empirical Dynamics models are only valid for the range of input data used to train and test them. If the ADAMS Solver supplies input outside of this range, it will be important to provide a warning to the User.

7 Summary

This paper has presented Empirical Dynamics modeling as an effective method to generate and use blackbox models for complex components. The complications of amplitude dependence, frequency dependence, arbitrary input, multiple inputs and outputs, and arbitrary system types, are all accommodated within the Empirical Dynamics framework, via use of neural networks and tapped delay structures for inputs and outputs. The capabilities of Empirical Dynamics modeling were demonstrated in case studies for shock absorbers and a rubber bushing from a passenger car. In general, Empirical Dynamics models provide a way to model complex components simpler and faster. The integration of EDM into the ADAMS environment is currently under development.

8 References

[1] Vladimir Cherkassky and Filip Mulier, *Learning from Data: Concepts, Theory, and Methods*, John Wiley & Sons, 1998.

[2] J.W. Fash, *Modeling of Shock Absorber Behavior using Artificial Neural Networks*, SAE Technical Paper 940248, 1994.

[3] Joseph Giacomin, *Neural Network Simulation of an Automotive Shock Absorber*, Engineering Applications of Artificial Intelligence, Vol. 4, No.1, pp. 59-64, 1991.

[4] Martin T. Hagan, Howard B. Demuth, Mark Beale, *Neural Network Design*, PWS Publishing Co., Boston, 1996.

[5] Lennart Ljung, *System Identification:Theory for the User*, 2nd Ed., PTR Prentice-Hall, Upper Saddle River, NJ, 1999.

[6] Kumpati S. Narendra and Kannan Parthasarathy, *Identification and Control of Dynamical Systems Using Neural Networks*, IEEE Trans. Neural Networks, Vol.1, No.1, March, 1990. pp. 4-27.

[7] Boaz Porat, *A Course in Digital Signal Processing*, John Wiley & Sons, 1996.

[8] Jonas Sjoberg and Lester S.H. Ngia, Neural Networks and Related Model Structures for Nonlinear System Identification, in *Nonlinear Modeling: Advanced Blackbox Techniques*, Johan A.K. Suykens and Joos Vandewalle (Eds.), Kluwer Academic Publishing, 1998, pp 1-28.

[9] Donald D. Weiner, John E. Spina, *Sinusoidal Analysis and Modeling of Weakly Nonlinear Circuits,* Van Nostrand Reinhold, 1980.

[10] Rodger E. Ziemer, William H. Tranter, D. Ronald Fannin*, Signals and Systems, Continuous and Discrete*, Macmillan Publishing, New York, 1983.