IFAC World Congress, Munich FRG, 27-31 July 1987. © IFAC

EXPERT-AIDED CONTROL ENGINEERING ENVIRONMENT FOR NONLINEAR SYSTEMS

James H. Taylor
Control Technology Branch
GE Corp. R & D
Schenectady, NY 12345 USA

John R. James
Dept. of Electrical Engg.
USMA
West Point, NY 10996 USA

Dean K. Frederick
ECSE Dept.
Rensselaer Polytechnic Inst.
Troy, NY 12181 USA

Abstract: We have been working to develop an **expert-system-based environment** for Computer-Aided Control Engineering (**CACE**). Our goal is to create a high-level user interface to conventional CACE tools, with substantial capabilities in areas that are either very complicated, or that require heuristic logic or specialized knowledge, or both. So far, we have developed rule bases for linear system diagnosis, specification development, lead/lag compensator synthesis, and design validation.

We report here on recent research in **expert-aided CACE for nonlinear systems**. Our approach is to perform extensive experimentation (using both numerical and symbolic processing) under the direction of a rule base containing "expertise" in and heuristic strategies for nonlinear CACE. The results and status of this effort are described in detail. This work represents the first phase in an iterative process; much is being learned that will be folded back into the rule base to improve the capabilities of the expert system. We should stress that our expert system is not particularly deep at this time - our present objective is to aid the user in making the most effective use of rather complicated procedures and conventional CACE software; in many instances, this simply involves "common sense".

Keywords: Computer-aided system design; expert systems; nonlinear systems; describing functions / Fourier analysis; artificial intelligence; computer interfaces for CAD; heuristic programming.

1. INTRODUCTION

There has been a major effort in recent years to develop very capable and sophisticated computer-aided control engineering (CACE) software environments. Some of the earlier and best-known major developments in this area are the UMIST Suite, CLADP (the "Cambridge Package"), KEDDC, the Lund Packages, CTRL-C, $MATRIX_X$, and the GE Federated System. These and other noteworthy examples may be found in the proceedings of the IFAC Symposia on CACSD (1982, 1985) and in ELCS (Rimvall, 1986).

As the number, capabilities, and complexity of this software increases, the possibility that any individual will be able to apply the available techniques and software effectively decreases. This problem is compounded in treating nonlinear systems: The number of approaches and the amount of expertise required to employ them successfully are both large. We believe that the **expert systems approach** provides an effective solution to this difficulty.

Our recent efforts (refer to Taylor and Frederick (1984); Taylor, Frederick, and James (1984); James, Frederick, and Taylor (1985); and James, Taylor and Frederick (1985)) have focussed on the use of expert systems to aid a control engineer in exploiting available software to carry out analyses and achieve acceptable designs. We use General Electric's DELPHI expert system shell in combination with the following conventional analysis and design software:

a. CLADP for analysis, design, and simulation of linear systems (mainly frequency domain; see Rimvall, 1986),

b. SIMNON for nonlinear simulation (see Rimvall, 1986),

c. extensions to SIMNON (Taylor, 1982, 1985), for determining equilibria, calculating conventional linearized models, and generating frequency-domain sinusoidal-input describing function characterizations of nonlinear system input / output relations, and

d. SFPACK, a MATLAB-based package for state-space analysis and design (Minto and Vidyasagar, 1985).

These packages are included in the GE Federated System (Spang, 1984); they were extended slightly to support the exchange of data with the expert system.

Integrating the numerical capabilities of the above software with an expert system results in a high-level combination of large-scale numerical processing with large-scale symbolic processing. We use DELPHI to invoke the various numerical routines, transform and exchange data required to carry out an analysis and design task, and interface with the user. For example, more than a hundred commands may be issued to CLADP by the expert system in the process of designing a single-input / single-output control system compensator (James, Frederick, and Taylor, 1985). Some of this activity is purely mechanical; other involves heuristic rules, e.g., for adjusting the parameters of a compensator to meet specifications. This software system thus achieves a transfer of much of the burden of the complexity of the control engineering problem from the shoulders of the design engineer to the expert system. Our most recent effort has been to extend this type of expert aiding from the linear domain to nonlinear systems.

Organization: In Section 2 we discuss the basic functions that belong in an expert's nonlinear control engineering tool kit, and the integration of these functions into high-level "expert procedures" is developed in Section 3. Other approaches to expert aiding are mentioned in Section 4. The operation of the expert system's nonlinear CACE knowledge base is illustrated in Section 5, where we apply the procedures of Section 3 to a position control problem.

2. CACE FUNCTIONS FOR NONLINEAR SYSTEMS

The following operations provide the "building blocks" that are integrated into expert-aided procedures for nonlinear system CACE in Section 3. We assume throughout that the user has a model of the nonlinear plant in the form

$$\dot{x} = f(x, u) \qquad \qquad (2.1)$$
$$y = h(x, u)$$

where x represents the state vector of dimension n, u denotes the input vector of dimension m, and y is the output or measurement vector of dimension p.

2.1 Direct Simulation

First explorations of a nonlinear system's behavior are usually carried out via simulation. By this, we mean writing the nonlinear equations (2.1) in a form suitable for numerical integration, specifying initial conditions on the states x_0 and the input function $u(t)$ appropriately, selecting a suitable numerical integration algorithm, and carrying out the integration to generate time-histories $x(t;\, x_0,\, u(t))$. This is done both to validate the model and to gain a better understanding of the behavior of the system.

2.2 Equilibrium Determination

A fundamental step in designing controls for a nonlinear plant is defining the operating point. Often this is an equilibrium corresponding to a given constant value of the input:

$$(2.2) \qquad u_0 \rightarrow x_0 : f(x_0, u_0) = 0$$

This is not the only way an operating point can be defined. For example, in dealing with flight control system design, the operating point is often a "trim condition":

$$x_0 \rightarrow u_0 : f(x_0, u_0) = 0$$

This is not a generic problem, as the user cannot specify x_0 arbitrarily. Instead, knowledge of the physical significance of the states must be used to generate a meaningful x_0 that can in fact correspond to trim. Since our objective is to provide expert-aiding for generic aspects of nonlinear CACE, the only operating points considered are equilibria.

2.3 Linearized Model Determination

Given an operating point (u_0, x_0), the formal definition of a linearized model is based on partial derivatives:

$$(2.3) \qquad \begin{aligned} \delta\dot{x} &= A\,\delta x + B\,\delta u \\ \delta y &= C\,\delta x + D\,\delta u \end{aligned}$$

where $\delta x \triangleq x - x_0$, $\delta y \triangleq y - y_0$, $\delta u \triangleq u - u_0$, and

$$(2.4) \qquad \begin{aligned} A &= \left[\partial f / \partial x\right]_{u_0, x_0}, \quad B = \left[\partial f / \partial u\right]_{u_0, x_0}, \\ C &= \left[\partial h / \partial x\right]_{u_0, x_0}, \quad D = \left[\partial h / \partial u\right]_{u_0, x_0} \end{aligned}$$

This process is often called small-signal linearization (SSL), to underscore that the model is only valid in the immediate vicinity of the operating point (u_0, x_0), assuming that the indicated partial derivatives exist.

Linearized models can be obtained by symbolic manipulation (e.g., MACSYMA); however, approximating these partial derivatives by finite differences is easier to implement numerically, usually faster in terms of computer execution time, and can be more informative in assessing the importance of nonlinear effects (cf. Step *e*, Section 3.1).

Using the scalar case for simple notation, the partial derivative of f with respect to x can be approximated by taking a central difference:

$$(2.5) \qquad \frac{\partial f}{\partial x} \cong \frac{f(x + \delta x) - f(x - \delta x)}{2\delta x}$$

One must be concerned with *round-off error* if δx in Eqn. (2.5) is too small, and *truncation error* due to the curvature of f if δx is too large. A *robust* automatic linearization method is used in our system (Taylor 1982) that evaluates central differences for δx and $2\delta x$, adjusts the perturbation δx based on the two estimates, and then combines them appropriately to minimize truncation error. This algorithm can also be used with a user-specified value of δx; one concern with the automatic algorithm is that discontinuities in $f(x)$ may defeat the adjustment, in which case manual selection of δx may be very useful (Section 3.1).

2.4 Diagnosis of Linearized Models

The diagnosis of a linearized model can provide a large body of useful information regarding the stability properties of the system in the vicinity of the operating point and other qualitative evaluations such as the controllability and observability of the system, "wrong-way response" due to right-half-plane zeroes, etc. Such diagnoses are well known and can be carried out using conventional software.

2.5 Distortion Analysis

One "measure of nonlinearity" of a process is the amount of distortion produced as it responds to a given type of input signal. The most common input in this context is the sinusoid, because stable linear systems excited by such an input produce a sinusoidal output in the steady state.

A standard approach to distortion analysis is to simulate the nonlinear system with sinusoidal inputs of amplitude Δu and perform a Fourier analysis of the system output signals to determine the amount of higher harmonic content. The usual distortion measures involve the second- and third-harmonic content of the output:

$$(2.6) \qquad \begin{aligned} \mu_{D,2} &= \sqrt{a_2^2 + b_2^2} \,/ \sqrt{a_1^2 + b_1^2} \\ \mu_{D,3} &= \sqrt{a_3^2 + b_3^2} \,/ \sqrt{a_1^2 + b_1^2} \end{aligned}$$

where a_i and b_i denote the Fourier coefficients (in-phase and quadrature) determined for the output of the system.

Fourier analysis of the process output can be carried out "off-line", i.e., by post processing simulation data from which the transient portion of the response has been eliminated, or more directly with a straightforward extension of software for nonlinear simulation (Taylor, 1985). In the latter approach, Fourier integrations are performed simultaneously with the integration of the system response to the sinusoidal input; a convergence test is made to determine when the simulation has reached steady state and the values of the Fourier integrals provide the harmonic content information needed for the distortion measures Eqn. (2.6).

2.6 Compensator Design

The most straightforward method for designing compensation for a nonlinear system involves choosing a "good" linearized model for the plant and using classical or modern techniques to design linear compensation. This is often adequate, and it is fair to say that most working control systems were designed in this way. Our present expert system for nonlinear system CACE is based on such an approach; however, we do intend to extend this system to include the nonlinear compensator synthesis methods of Taylor and Strobel (1984, 1985).

2.7 Design Validation

The minimum acceptable validation of a control system design is to simulate the closed-loop system (with the nonlinear plant model) for step inputs of amplitude Δu_{ref} selected to cover the expected range of operation (e.g., small, moderate, large Δu_{ref}). Other validation exercises might include robustness tests (simulations at various operating points or with parameter variations representative of their uncertainty), response tests for other types of inputs, more realistic emulations of the control system software and hardware, etc.

3. EXPERT-AIDED NONLINEAR CACE

The knowledge acquisition part of this effort is being approached using the "expert procedure modeling" technique described in Taylor and Frederick (1984): We pose "archetypical problems" and develop a conceptual framework and plan for their solution. This structures the knowledge and results in an architecture definition within which we implement the expert system. The resulting architecture for CACE consists of a supervisory rule base, which oversees the CACE activity at the highest functional level, and "worker" rule bases which deal with model diagnosis, constraint definition, specification development, control system design, and design validation.

We implement this approach by combining the basic operations outlined in Section 2 to realize our concept of an "expert approach" to nonlinear CACE. The areas where we believe nonlinear CACE is most substantially different from the linear case (in terms of "standard practice", not theory) are **modeling** and **control system design.** We thus focus exclusively on those areas and the associated rule bases.

3.1 Rule Base for Modeling Nonlinear Systems

The goal of modeling is to understand the nature of the plant and provide the basis for control system design; the output is a data base characterizing this process. Our plan for an "expert approach to nonlinear system modeling" is based on the fundamental requirement that the user be asked to provide only a **state-variable differential equation** (2.1) in a form acceptable by conventional software for simulation and analysis, and the **barest minimum** ***a priori*** information regarding the behavior of the nonlinear system. (We could probably produce a "smarter" system if we make the problem less generic; however, we are exploring the general case, since treating a system model as a black box seems to require considerable ingenuity and may provide a great deal of support for the user who is not a process expert.) We do presume that the system is **asymptotically stable** within its operating regime; otherwise many of the procedures below (e.g., determining frequency response via simulation) might have to be modified or eliminated.

In addition to the dynamic model, we believe that it is reasonable to ask what value of u_0 the user is considering (which defines the **operating point**, Sect. 2.2) and the approximate **maximum input excursions** Δu that are anticipated. The user should certainly be able to determine the answers to these questions by simulation, if not from experience. If the user is quite sure that the system model has no discontinuities, this is sufficient information; if discontinuities may be present, we also ask for an estimate of the system **rise time** T_r, for reasons outlined below. Given only this information, expert-aided nonlinear system modeling uses the following strategy:

a. Determine the equilibrium x_0 (Section 2.2) (if the equilibrium is not unique, then the user may have to provide an initial guess to obtain the desired point).

b. If there may be discontinuities, check for integrability: choose first-order Euler integration with step sizes $dt = 0.01{*}T_r,\ 0.02{*}T_r,\ 0.04{*}T_r$ and see if a reasonable error behavior is observed between the resulting time histories; if it is not, try doubling or halving the step size until a suitable error relation is obtained. An "optimal" dt is one such that the difference between that time history and one generated with $2{*}dt$ is within a specified ϵ while the next doubling ($4{*}dt$) yields unacceptable error. Then try a more sophisticated algorithm (e.g., Runge Kutta with variable step size) and see if the same accuracy can be obtained with less computer time. (If there are no discontinuities, the latter algorithm can usually be chosen directly. High-order predictor/corrector methods should not be used unless $f(x, u)$ and $u(t)$ are at least continuously differentiable.)

c. Determine a SSL linearized model for the operating point (u_0, x_0) using the automatic linearization algorithm (Section 2.3). This is a *provisional* model until the issue of discontinuity is resolved (Step e).

d. Determine the range of the state variables (Δx) and output variables (Δy) corresponding to the input signal range Δu: select a range of frequencies $[\underline{\omega}, \overline{\omega}]$, based on the eigenvalues of the linearized model if the user believes the system is continuous, or on the rise time T_r if this is not so; simulate the nonlinear system for sinusoidal inputs of amplitude Δu; and search for the corresponding maximum state amplitude over the frequency range. Note that the frequency range can be modified if the behavior so indicates; e.g., if the response is flat over $[\underline{\omega}, \overline{\omega}]$, then $\overline{\omega}$ should be increased, while if the response is continuously rolling off over $[\underline{\omega}, \overline{\omega}]$, then $\underline{\omega}$ should be decreased. It is important to determine $[\underline{\omega}, \overline{\omega}]$ so that "all the important action" (e.g., breakpoints) occurs over this range.

e. Check for discontinuities in the nonlinear system model at the operating point (u_0, x_0). This is done by using linearization (Section 2.3) with manually-selected perturbations based on the anticipated range of system variables determined in Step d, as follows: obtain two linearized models, one for $(0.02\Delta u, 0.02\Delta x)$ and the other for one-half these perturbations. Denote the resulting arrays $\{A^{(1)}, B^{(1)}, C^{(1)}, D^{(1)}\}$ and $\{A^{(2)}, B^{(2)}, C^{(2)}, D^{(2)}\}$, respectively. If any element(s) of $A^{(2)}$ are about twice as large as the same element(s) of $A^{(1)}$, then $f(x, u)$ is discontinuous with respect to x; if any element(s) of $B^{(2)}$ are about twice as large as the same element(s) of $B^{(1)}$, then $f(x, u)$ is discontinuous with respect to u; and similarly C, D can be inspected to find discontinuities in $h(x, u)$. If there are large disparities but not in the ratio 2:1 indicated, then halve the perturbation sizes and check again. Fig. 1 shows why this approach will effectively reveal discontinuities if carried out and interpreted correctly.

f. Check for multi-valued nonlinearities? (We have not yet been able to devise a general strategy to do this.)

g. Based on the results of Steps c and e, determine the *robust linear model* for the plant: if the system is continuous, then use the SSL model from Step c; if it is not, then replace the corresponding element(s) of the linear model arrays with the appropriate ***sinusoidal-input describing function*** (SIDF) gain(s). This is done as follows: assuming that the nonlinearity is

$$f(x) \cong m\,x + d\ sgn\,x,$$

for which the two derivative estimates in Step e are

$$[df/dx]^{(1)} = m + d/\delta x\,,\quad [df/dx]^{(2)} = m + 2d/\delta x$$

One can solve for m and d as

$$d = \delta x\{[df/dx]^{(2)} - [df/dx]^{(1)}\},$$

$$m = 2[df/dx]^{(1)} - [df/dx]^{(2)}.$$

Given m and d, the SIDF gain is $[m + 4d/\pi\Delta x]$, where Δx is the maximum amplitude of x (Step d). This new gain then replaces the invalid value in the SSL model; denote this model LRB_0.

h. Perform a linear diagnosis of this model (Section 2.4).

i. Diagnose the nonlinear model for distortion: Given sinusoidal inputs of amplitude $0.1 * \Delta u$, $0.5 * \Delta u$, and Δu and frequencies over the range $[\underline{\omega}, \overline{\omega}]$, determine the higher harmonic content in the system output (Section 2.5); distortion measures are given in Eqn. (2.6).

j. Diagnose the robust linear model for "fit": perform simulations of this model for sinusoidal inputs of amplitude Δu and frequencies over the range $[\underline{\omega}, \overline{\omega}]$, and compare the response amplitudes with those obtained in the step above (divide the response by 2 and 10 for the low-amplitude comparisons).

k. Determine linearized model sensitivity to perturbations in operating point: find equilibria for $u_0 + \frac{1}{2}\Delta u$ and $u_0 - \frac{1}{2}\Delta u$, denoted x_0^+ and x_0^-, respectively; then find robust linearized models at those equilibria (Step g) denoted LRB_0^+ and LRB_0^-; diagnose these models; quantify and compare:

- Eigenvalue sensitivity: Report this to the user in the form of root locus diagrams (characteristic loci in the multivariable case), where the changes in roots are due to operating point perturbations; quantify this in terms of percentage change.
- Transfer function sensitivity: Report this in the form of Bode plots of magnitude and phase (maximum and minimum singular values for the multivariable case); quantify this using H^2 or H^∞ norms.
- Step response sensitivity: Report this in the form of time-history plots; quantify this using integrated squared error divided by the integral of the nominal system response squared (normalized ISE); e.g., given $h_s(t)$ for the nominal operating point and $h_s^{(+)}(t)$ for the x_0^+ point,

$$(3.1)\qquad \mu_I = \int_0^T [h_s(t) - h_s^{(+)}(t)]^2\,dt \Big/ \int_0^T [h_s(t)]^2\,dt$$

  This measure can be obtained directly by expanding the overall model to include a subsystem that performs the required integrals during simulation.

Note that systems with odd symmetry about the operating point will yield perturbed models LRB_0^+ and LRB_0^- that are identical; the test $LRB_0^+ \cong LRB_0^-$ may be used as a check for this condition.

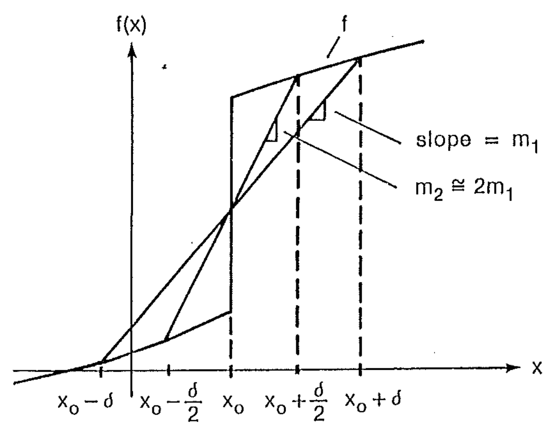

Figure 1. Detection of Discontinuities

3.2 Rule Base for Compensator Design

We have developed a lead/lag compensator synthesis rule base for linear control system design (James, Frederick, and Taylor, 1985). In our present approach to nonlinear control system design, we apply this rule base to the "most appropriate" linearized model of the plant. This strategy is straightforward, and conforms to common industrial practice for nonlinear systems; it generally works if linear compensation is adequate for the task. A more effective approach has been developed recently which synthesizes nonlinear compensators for highly nonlinear systems; see Taylor and Strobel (1984, 1985); however, we have not yet implemented these advanced techniques in our expert system. Therefore, we focus on selecting the most appropriate linearized model, which involves substantial judgement based on a modeling exercise as described in Section 3.1.

At the end of the modeling effort, we have an indication of the severity of the system nonlinearity and of where troubles might be encountered (e.g., for small input signals). In addition, there is a collection of linearized models that can be used as a set of candidates for the selection of the most suitable model for design and for direction in the design process. For example, these models may be used to quantify model uncertainty, and thus serve as the basis for a robust synthesis approach using singular values; or they may be used to define a conservative "worst-case" model (e.g., in the single-input / single-output case, the model with the least gain and/or phase margin) which is then used in conjunction with classical design techniques.

In our case, validating the linearized model to be used by the lead/lag compensator rule base is critically important. We perform this task automatically by making step- and sinusoidal-input response comparisons: We simulate both the nonlinear model and a candidate linearized model of the system with step or sinusoidal inputs of appropriate amplitude to generate output time-histories $y_N(t)$ and $y_L(t)$ and compare; the measure of linearized system modeling error is integrated squared error divided by the integrated square of the nonlinear system response (normalized ISE); cf. Eqn. (3.1). The input amplitudes used are based on modeling results (Section 3.1); the interpretation of the analysis and execution of further iterations if required are governed by the rule base. The output of this validation process (e.g., measures of goodness of fit) is used to select the best model for design, as a basis for control system performance specification, and to assess the likelihood of success in the linear compensator design process.

3.3 Interpretation and Heuristic Reasoning

The above items constitute a rather mechanistic statement of the overall procedure for expert-aided nonlinear modeling and design. The outcome of modeling should be a sound understanding of the behavior of the process and the information needed to perform a control system design with a good assurance of success. This information might include, for example, a basis for choosing performance specifications and the type of controller to be designed (e.g., lead/lag, PID, . . .; linear or nonlinear; we currently support only the first of these). Several points call for further discussion: the interpretation of the results, and the role of heuristic reasoning in the process. These considerations, in effect, justify implementing this process using an expert system rather than a conventional approach (e.g., setting up command files or macros to run the CACE software).

Interpretation: Some of the steps in Section 3.1 produce information that is of clear importance and meaning. For example, a diagnosis of a SSL system model (assuming the system is continuous) reveals a great deal of information regarding the behavior of the system in the vicinity of the operating point; see Section 2.4. Other steps have significance that is not so obvious:

- Step b: Integrability testing may provide early warnings of potential problems: for example, if the automatic algorithms take large amounts of computer time compared with the first test suggested, than the system is quite likely to be discontinuous or worse. The dynamics may also be stiff (have some states that are much faster than others). Either condition may be difficult to handle.
- Step d: the frequency range $[\underline{\omega}, \overline{\omega}]$ provides a measure of the intrinsic time-scale(s) of the plant which is very important if the system is not continuous. The ranges of the state and output variables in response to the user's definition of the range of the input excursion may also provide information regarding the suitability of the original Δu specification, and also serves to define what might be considered to be "small, medium and large" perturbations in other analyses (see, e.g., Steps e, i).
- Steps e, f: the importance of discontinuity in the system model at the operating point is well known. The main issues are: SSL models are not defined for systems without partial derivatives, so one must define an alternative model with considerable care (Step g); and discontinuities may lead to "chatter" or limit cycle phenomena in the closed-loop system unless it is designed to have infinite gain margin. Multi-valued nonlinearities in the system model are even more troublesome than discontinuities, so a simple, generic approach to detecting them would be very beneficial.
- Steps i, j, k: these analyses provide metrics for the severity of the nonlinearity of the system and quality of fit for the linearized model from Step g. These measures should be valuable to the user (e.g., for arriving at meaningful specifications), and may be used in the control system design rule base to aid in selecting the best linearized model to use as the basis for design.

In all of the above, there may also be a subjective aspect of interpretation. For example, conclusions of the form *"The system is very stiff."* or *"The system is severely nonlinear for small inputs."* may be critical in determining how to design the control system. Such statements might be reached by applying heuristic and/or fuzzy reasoning methods to the numerical results obtained in the diagnostic process.

Heuristic Reasoning: Executing Steps a through k in the modeling procedure is not as easy as might be inferred from the listing in Section 3.1. The expert system may have to reason and iterate in response to a number of eventualities:

- *Numerical problems:* In every case, an analysis must be set up and carried out with conventional software. Occasionally, the problem cannot be solved for numerical reasons, and another algorithm must be selected or a convergence parameter must be changed. Most of this information is in the form of heuristic rules known to expert users. Examples:
  - → Step b - if there are discontinuities in the system model, then do not use predictor/corrector methods for numerical integration (they use previous derivative values which become invalid at a discontinuity).
  - → Steps d, i, j - analyses of steady-state sinusoidal input response require careful handling of transient behavior and convergence (cf. Taylor, 1985).
- *Groping:* There are numerous areas in which the expert system has to use a rule of thumb to perform a trial, then perhaps modify the experiment and iterate until suitable information is obtained. Examples:
  - → Step b - iterate until a suitable integration method is found, as indicated.
  - → Step d - determine $[\underline{\omega}, \overline{\omega}]$: either the rise time estimate T_r or the eigenvalues of the SSL model provide a rough initial guess for this frequency range. For example, taking the minimum and maximum magnitudes of the SSL eigenvalues or taking $[0.3/T_r, 30/T_r]$ (since $\omega_n \cong 3/T_r$ for a second-order system) are reasonable starting values. Then iterate as indicated, using heuristics to ensure that "all the action" occurs in the final frequency range.
  - → Step e - checking for discontinuities may involve considerable trial-and-error, since the 2:1 rule only works for appropriate perturbation magnitudes.
  - → Steps i and j are based on the meaning of "small, medium, and large" input variations; the suggested amplitudes may or may not be the most significant cases to consider. For example, if the first-cut analysis with the multiples 0.1, 0.5, 1.0 shows that the system is severely nonlinear for small inputs,

then perhaps the analysis should be repeated for 0.02 and 0.05 to explore the problem further. Also, one may find that the number of frequencies selected in $[\underline{\omega}, \overline{\omega}]$ may need to be iterated if there are resonances or other need for detailed analysis.

→ Step k - quantifying linearized model sensitivity meaningfully is a difficult and subjective business. Quantifying this is not clear-cut (e.g., percent change of an eigenvalue near the origin may not be a good measure), and the significance of a sensitivity measure depends on context (e.g., the H^{∞} norm difference in frequency response has different significance near the desired system bandwidth than far above or below that point).

We believe this sort of trial-and-error study is required to produce definitive model diagnoses. An expert system can mechanize the many details and simple rules of thumb that make the difference between success and failure.

4. NON-GENERIC NONLINEAR CACE

As mentioned previously, there is a clear relation between the amount of *a priori* knowledge built into the expert system or supplied by the user and the depth of reasoning that can be performed. Several areas have come to mind in this context: Nonlinear CACE for a particular problem domain, and nonlinear CACE for systems that can be modeled using a particular formalism. There are undoubtedly other issues, and the following thoughts are only tentative.

4.1 Specific Problem Domains

The ability to go beyond a black-box treatment of a nonlinear process as detailed in Section 3 requires further knowledge of the process. Such information may provide a substantially better basis for modeling and control system design than can be obtained via the procedure in Section 3.

In the modeling area, for example, expert-aided diagnosis of a robot model could go into a number of issues that might be impossible in the generic case. One can conceive of simulation tests that can detect certain types of nonlinear friction (e.g., stiction), backlash, and torque motor saturation. Each of these effects can be made to reveal a distinctive signature if the suitable simulation is performed; the situations and signatures are well-known to robotics experts.

In addition to detecting the presence of such nonlinear effects, their severity can be assessed and it can be determined whether or not they need to be factored into the design of the control system and how. For example, given the information regarding the allowable input excursions Δu and an assessment of torque motor saturation, the need to accommodate integral wind-up can be determined. Similarly, the need for high gain for small inputs to correct for stiction and/or backlash can be inferred from performance specifications and diagnosis of these effects.

The above points illustrate how constraining the problem domain can lead to more definitive results. We believe that it is generally true that any discipline has a set of known problems, known ways for determining how important they are, and standard "fixes". These can be a valuable resource for expert-aided nonlinear CACE which can be folded into the generic rule bases already described.

4.2 Constrained Model Formulations

Certain nonlinear effects are difficult to detect in a black-box context via experimentation (e.g., multi-valued nonlinearities, Section 3.1). Another possibility is to use **lexical analysis** on the system model. The success of this approach would be inversely proportional to the freedom of the user in choosing the formulation of the model.

At the one extreme, if the user is allowed to supply the model in the form of Eqn. (2.1) in a relatively general, high-level, flexible language such as FORTRAN, then it is unlikely that allowing the expert system to inspect the model will be fruitful.

As an intermediate case, if the system is formulated in a constrained modeling language like that used in SIMNON, then dissecting the model may be very helpful. A SIMNON model is shown in Table 4.1: Note that the names of the primary variables are given, the possible nonlinear functions are well defined, and the use of "if - then - else" constructs to write piece-wise-linear, discontinuous, and multi-valued nonlinearities is quite transparent. Lexical analysis might not be simple, but something useful can clearly be done.

Table 4-1. Example SIMNON System Model

```
continuous system MOTORLD
" Saturating servo motor driving a load with
" Coulombic friction and linear plus cubic spring
state el x1 x2    " States are 'lag', theta, omega; theta =
"                   angular position, omega = angle rate
der del dx1 dx2   " Derivatives of states
input u           " Input to motor
output y          " Output = angular position
"    Smoothly saturate the input:
Tmot = G1 * sign(el) * (1.0 - exp(-B1*abs(el)))
"    Viscous plus Coulombic friction:
Tfrict = F1 * x2 + F2 * sign(x2)
"    Linear plus cubic spring:
Tspring = K1*x1 + K3*x1*x1*x1
"    Effective load torque:
Tel = Tmot - Tfrict - Tspring
del = pole * (u - el)
dx1 = x2              " theta dot = omega
dx2 = Tel / J         " omega dot = torque / MOI
y = x1                " output = theta
pole: 10.0 " pole for electrical lag
G1: 2.0   " -> Tmot saturates at u = 2.0 v;
B1: 0.5   " -> gain for small input = 1.0 Nm/v
F1: 0.1   " Viscous term; Nm-s/rad
F2: 0.3   " Coulombic coefficient; Nm
K1: 1.0   " Linear term; Nm/rad
K3: 0.5   " Cubic term; Nm/rad^3
J: 0.01   " Moment of inertia; kg-m**2
end
```

Lexical analysis becomes straightforward only if the user must formulate the process model in terms of components related via a precise interconnection formalism, and all components of the model are taken from a "library" of nonlinear elements. Some special-purpose simulation packages are structured in this fashion, so expert-aided modeling in this sense is possible in such environments.

5. EXAMPLE OF EXPERT-AIDED NONLINEAR CACE

The expert system defined above is currently being developed. We are coding specialized software (SIMNON systems, SIMNON and SFPACK macros, LISP functions) and rule-base modules to execute the steps outlined in Section 3; the overall strategy will be implemented once the modules are validated. We demonstrate several of these modules by applying them to an illustrative example:

The problem is position control system analysis and design. The plant model corresponds to an electro-mechanical positioning servo (motor plus load) with a first-order "electrical lag" and three nonlinear effects: Coulombic friction, a cubic spring term, and torque saturation. The equations and parameter values are given in Table 4-1.

Many of the nonlinear CACE modules outlined in Section 3 have been exercised on this problem. The starting knowledge was: $u_0 = 0.0$ volts, $\Delta u = 2.0$ volts, $T_r \cong 1/8$ sec. Sample results:

1. Integrability, Step b: We found that the "optimum" dt was in fact $T_r/100$, and that first-order Euler with fixed step size was the most effective algorithm. Fourth-order Runge-Kutta with variable step size used 3.9 times more computer time for acceptable accuracy.
2. State variable range, Step d: We used our SIDF-generating capability (Taylor, 1985) to obtain the maximum magnitudes Δel, Δx_1, and $\Delta x_2 = 2.0$, 0.94, and 7.03, respectively. In so doing, we first tried $\underline{\omega} = T_r/20$, $\overline{\omega} = 5T_r$ (see *Groping*, Section 3.3), but we found that $\underline{\omega} = 0.5$, $\overline{\omega} = 50.0$ was more appropriate in terms of covering the breakpoints.
3. Discontinuity check, Step e: We found that $a_{3,3}$ for δx_2

= 0.04 and 0.02 was -760 and -1510, respectively, clearly detecting the discontinuous Coulombic term.

4. Robust linear model, Steps c, g: We used the data from item 3 above to solve for $m = 10.0$ and $d = 30.0$; based on $\Delta x_2 = 7.03$, we calculated the corresponding SIDF gain to be $a_{3,3|SIDF} = 15.43$. The complete linear system LRB_0 was:

$$A = \begin{bmatrix} -10.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \\ 99.5 & -100.0 & -15.43 \end{bmatrix}; \quad B = \begin{bmatrix} 10.0 \\ 0.0 \\ 0.0 \end{bmatrix}$$

5. Distortion analysis, Step i: nine log-spaced frequencies were selected in $[\underline{\omega}, \overline{\omega}]$, and Fourier analysis was performed for input amplitude Δu to obtain negligible second-harmonic output and maximum third-harmonic content of 59% at $\omega = 1.6$ rad/sec. (This was done in the same run as item 2 above.)

6. Operating point sensitivity, Step k: the indicated analyses were conducted, and the frequency-domain behavior of LRB_0 and LRB_0^+ obtained - see Fig. 2.

7. Robust linear model fit (Section 3.2): the nonlinear model (Table 4-1) and LRB_0 were simulated for $u = \Delta u \sin(\omega\, t)$ and normalized ISE (Eqn. (3-1)) was calculated. A resulting pair of time-histories is shown in Fig. 3, indicating that LRB_0 is conservative. (The actual system response is smaller due to Coulombic friction, saturation, and the nonlinear spring.) The ISE obtained was 0.084.

8. Compensator design: We used the CACE-III lead/lag compensator design rule base with specifications of closed-loop bandwidth $\omega_{BW} = 45$ rad/sec, gain margin = 20dB, low-frequency gain = 40dB; and we obtained a compensator with two leads (pole/zero ratios of 20 and 2.4) and one lag (zero/pole ratio = 17.9).

9. Design validation: We checked the closed-loop system via step-response simulations for $\Delta u_{REF} = 0.2$, 0.5, and 1.6, and obtained the time histories shown in Fig. 4 (a simulation with the linear plant LRB_0 is also shown for comparison). This verifies that the choice of LRB_0 for compensator design was conservative: Coulombic friction and saturation make the response over-damped for small and large reference inputs, respectively. Saturation clearly limits performance for large inputs.

6. CONCLUSION

Handling a nonlinear control system design problem in a modern CACE environment can be very complicated, time-consuming, and involve using a large number of trial-and-error studies of the sort described in Section 3. Managing this process using an expert system to carry out at least a preliminary "reasonable" exploration of the properties of the plant and compensator design and validation exercise should provide a great deal of assistance and insurance to the user, whether novice or expert.

The approach to nonlinear CACE described above and the corresponding rule base that is under development represent Phase 1 of our attack on the problem. We emphasize that we expect that our understanding and software will both become deeper as we develop, test, and refine the system.

REFERENCES

1. James, J. R., D. K. Frederick, and J. H. Taylor (1985). On the application of expert systems programming techniques to the design of lead/lag precompensators. *Proc. Control 85*, Cambridge, UK; to appear in *IEE Proceedings D: Control Theory and Applications.*

2. James, J. R., Taylor, J. H., and Frederick, D. K. (1985). An expert system architecture for coping with complexity in computer-aided control engineering. *Proc. 3rd IFAC Symposium on CAD in Control and Engineering Systems,* Lyngby, Denmark.

3. Minto, K. D. and Vidyasagar, M. (1985). Design of reliable control systems: theory and computations. Tech. Rpt. No. UWEE85-03 (Ph. D. Thesis), Univ. of Waterloo, Waterloo, Ontario, Canada.

4. Rimvall, M. (1986). The extended list of control software (ELCS); contact M. Rimvall, Dept. of Automatic Control, Swiss Federal Institute of Technology (ETH), CH-8092 Zurich, Switzerland.

5. Spang, H. A. III (1984). The federated computer-aided control design system, *IEEE Proc., Vol. 72,* 1724-1731.

6. Taylor, J. H. (1982). Environment and methods for computer-aided control systems design for nonlinear plants. *Proc. 2nd IFAC Symposium: CAD of Multivariable Technological Systems,* West Lafayette, IN USA.

7. Taylor, J. H. (1985). Computer-aided control engineering environment for nonlinear systems analysis and design. *Proc. 3rd IFAC Symposium on CAD in Control and Engineering Systems,* Lyngby, Denmark.

8. Taylor, J. H. and Frederick, D. K. (1984). An expert system architecture for computer-aided control engineering. *IEEE Proceedings, Vol. 72,* 1795-1805.

9. Taylor, J. H., Frederick, D. K., and James, J. R. (1984). An expert system scenario for computer-aided control engineering. *Proc. American Control Conf.,* San Diego, CA USA.

10. Taylor, J. H. and Strobel, K. L. (1984). Applications of a nonlinear controller design approach based on quasilinear system models. *Proc. American Control Conf.,* San Diego, CA USA.

11. Taylor, J. H. and Strobel, K. L. (1985). Nonlinear compensator synthesis via sinusoidal-input describing functions. *Proc. American Control Conf.,* Boston, MA USA.

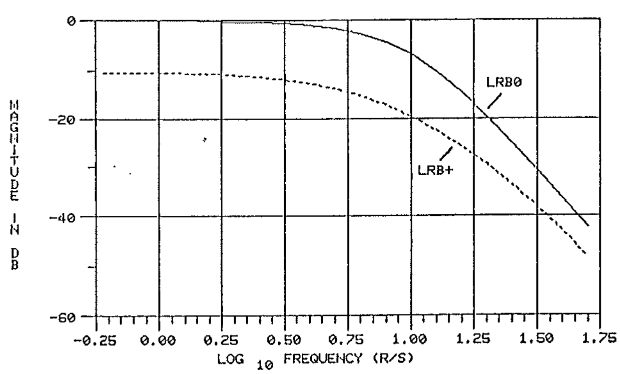

Figure 2. Bode Magnitude Plots of LRB_0 and LRB_0^+

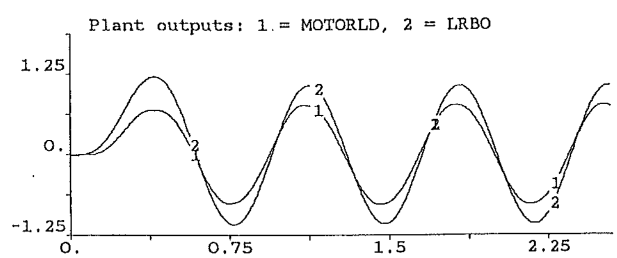

Figure 3. Time Responses of MOTORLD and LRB_0

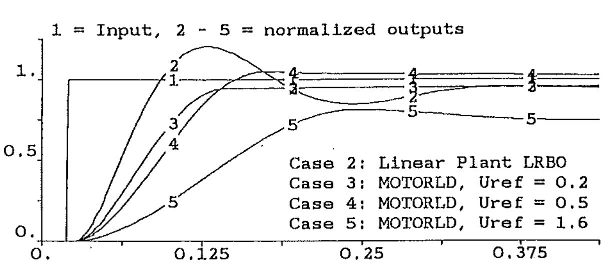

Figure 4. Compensator Design Validation