Proceedings of the
American Control Conference
San Francisco, California • June 1993

COMPUTER-AIDED CONTROL ENGINEERING ENVIRONMENT FOR THE SYNTHESIS OF NONLINEAR CONTROL SYSTEMS

James H. Taylor and Jin Lu
Odyssey Research Associates
Ithaca, New York 14850-1326
email: jim@oracorp.com

Abstract[1]

We describe a second-generation toolset for computer-aided nonlinear control system analysis and design based on sinusoidal-input describing function (SIDF) methods. There are two main elements in our CAD software: a simulation-based program for generating amplitude-dependent SIDF input/output models for nonlinear plants, and a frequency-domain nonlinear compensator design package. Both of these are described in detail. This software can treat very general nonlinear systems, with no restrictions as to system order, number of nonlinearities, configuration, or nonlinearity type. An overview of the application of this software to the ATB1000 Army test facility (comprising an electromechanical pointing system with flexible modes) will be provided, to illustrate the use and efficacy of these tools.

1 Introduction

The basis of this work has been established in earlier publications. An overview of SIDF models for nonlinear systems and the overall approach to controller design is provided in [3], and the methodology for multi-model nonlinear controller design by SIDF inversion is outlined and illustrated in [4]. A first generation toolset for computer-aided nonlinear control system design based on sinusoidal-input describing function (SIDF) method is described in [5].

This toolset implements the functions required to carry out the multi-model nonlinear controller design approach [4], based on SIDF modeling and SIDF inversion. This approach is based on the behavior of the nonlinear system for signals having various amplitudes that correspond to the actual anticipated operation (e.g., for "small", "medium" and "large" signals, as determined by the application). The amplitude dependence of a nonlinear system is a key characteristic that often must be considered in the analysis and design of robust nonlinear systems. This issue is distinct from the dependence of nonlinear system behavior on operating point, which can often be characterized by a family of standard linearized models about various operating points.

[1]Support for the research described herein has been provided by the Defense Advanced Research Project Agency through the U.S. Army AMCCOM at Picatinny Arsenal, NJ by Contract Number DAAA21-92-C-0013.

2 SIDF Modeling Tools

Software for generating amplitude-dependent frequency-domain models describing the input/output (I/O) behavior of nonlinear systems excited by sinusoidal inputs has been developed based on MATRIX$_X$ [6] and *dstool* [1].

2.1 A MATRIX$_X$-based SIDF Modeler

MATRIX$_X$ is a CAD software environment developed by ISI for system simulation and (linear) control design. We based the SIDF modeling on MATRIX$_X$ because we can use SystemBuild to easily model a nonlinear system using block diagram notation. The SIDF modeling program built on MATRIX$_X$ is called SIDFGEN; it has two components:

(1) **Integration**. To get the SIDF, a sinusoidal function $a\sin(\omega t)$ is used as input to the nonlinear system, where $a \in [a_l, a_u]$, $\omega \in [\omega_l, \omega_u]$. Here the range $[a_l, a_u]$ covers the input amplitude of interest, and $[\omega_l, \omega_u]$ covers the input frequency of interest. The output of the nonlinear system is integrated over the period of the sinusoidal function to get the SIDF:

$$Re(G_K(j\omega;a)) = \frac{\omega}{\pi a}\int_{t_0+KT}^{t_0+(K+1)T} y(t)\sin(\omega t)dt \quad (1)$$

$$Im(G_K(j\omega;a)) = \frac{\omega}{\pi a}\int_{t_0+KT}^{t_0+(K+1)T} y(t)\cos(\omega t)dt \quad (2)$$

where $Re(\cdot)$ and $Im(\cdot)$ are real and imaginary parts of the SIDF $G(j\omega;a)$, $T = 2\pi/\omega$, and $y(t)$ is the output of the nonlinear system. A SystemBuild model for doing the integration (1) and (2) is shown in Fig. 1. It contains both the nonlinear dynamics whose SIDF is to be found and the integrators for calculating (1) and (2).

Convergence testing. Convergence testing of the integrals (1) and (2) is a major concern in any attempt to automate I/O transfer function model generation, because transients generally occur in the simulation, and the integrals (1) and (2) are not meaningful until the contributions of these transients have decayed to become small compared to the steady-state response.

First, we have introduced error control parameters in order to provide the user with the required control over convergence. These are ϵ_m, ϵ_ϕ, and N_{cyc}. The first two of these

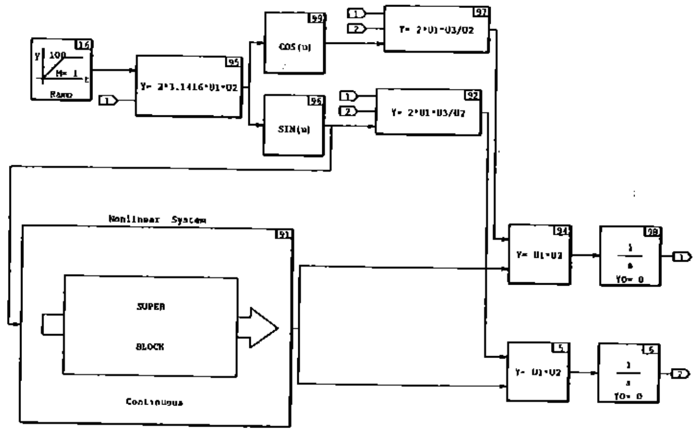

Fig. 1: A SystemBuild Model
(input: $1 \rightarrow f = \omega/2\pi, \quad 2 \rightarrow a$
output: $1 \rightarrow Im(G_K), \quad 2 \rightarrow Re(G_K)$)

are error bounds for the magnitude and phase. respectively; and N_{cyc} provides one mechanism to limit the number of cycles of the input sinusoid that will be used.

Somewhat loosely speaking, the integrals (1) and (2) are said to have converged when the magnitude and phase of G_K have converged in the sense that $M_K = |(G_K(j\omega; a)|$ and $\Phi_K = \angle G_K(j\omega; a)$ evaluated over the previous ((K-1)st) period of the simulation and the latest (Kth) period satisfy the following conditions:

$$err_m^K = \frac{| M_K - M_{K-1} |}{| M_K |} < \epsilon_m \qquad (3)$$

$$err_\phi^K = | \Phi_K - \Phi_{K-1} | < \epsilon_\phi \qquad (4)$$

The SIDF Modeling Tool then works as follows: For $a \in [a_l, a_u]$, $\omega \in [\omega_l, \omega_u]$, do the following;

1. Initialization: let $K = 0$, pick an initial time t_0;

2. Calculate $Re(G_K(j\omega; a))$ and $Im(G_K(j\omega; a))$ according to (1) and (2);

3. Calculate err_m^K and err_ϕ^K according to (3) and (4);

4. If $err_m^K < \epsilon_m$ and $err_\phi^K < \epsilon_\phi$ or $k = N_{cyc}$, then stop; otherwise let $k = k + 1$ and goto 2.

2.2 A *dstool*-Based SIDF Modeler

dstool is a tool for simulating and analyzing dynamic systems. One of its unique features is that it can determine limit cycles or other steady-state periodic solutions of a dynamic system. As a result, we can use *dstool* to generate the steady-state response of a nonlinear dynamic system driven by a sinusoidal function, then process (integrate over) that response to obtain the SIDF. We do not have to worry about the transient behavior of the system response or convergence testing.

Another feature of *dstool* is its capability of precisely simulating nonlinear systems involving discontinuities (e.g., a piece-wise-linear systems with relays, backlash etc.). Many existing simulation algorithms assume smooth nonlinear dynamics, so errors occur when simulating a discontinuous nonlinear system because an integration step may involve erratically calculating derivatives on either side of the discontinuity [1]. *dstool* incorporates an event-handling mechanism in its simulation algorithms to detect the moment when a trajectory crosses a discontinuity. Once a discontinuity is detected, the simulation algorithm can adjust its step size to avoid using different vector fields in one integration step.

The above features of *dstool* make it singularly appropriate for generating SIDF models of nonlinear systems. A *dstool*-based SIDF Modeling Tool is currently under development.

3 Nonlinear Controller Synthesis Tools

The basic idea [4] is to process a set of frequency-domain models for the nonlinear plant and synthesize a *nonlinear controller* so that the open-loop response of the resulting control system will be as insensitive to input (error signal) amplitude e as possible, where e takes on appropriate values $e_j, j = 1, 2, \ldots$ selected by the user, based on the study of the performance of the linear controller that lead to the decision to seek a nonlinear controller and on anticipated operating conditions.

This procedure involves seven steps:

1. select a set of plant operating regimes defined by input amplitudes a_k, $k = 1, 2, \ldots$ and generate the $G_k(j\omega;\ a_k)$ models using SIDFGEN, Section 2;
2. select one plant operating regime from this set, denoted by a^* and characterized by $G^*(j\omega, a^*)$;
3. design a linear PID controller denoted $C^*(j\omega)$ based on $G^*(j\omega;\ a^*)$ using some classical frequency-domain method (e.g., design to achieve good bandwidth and gain-margin);
4. use $C^*(j\omega)$ and $G^*(j\omega;\ a^*)$ to define an *achievable open-loop objective function* denoted $C\,G^*(j\omega)$;
5. take the set of models G_k and error-signal amplitude set $e_j, j = 1, 2, \ldots$ and for each j determine the compensator static gains $K_{P,j}$, $K_{I,j}$, $K_{D,j}$ required to force the frequency response of the PID followed by $G_k(j\omega;\ a_k)$ to fit the open-loop objective function $C\,G^*(j\omega)$ with minimum mean-square error;
6. pass the three gain sets $K_{P,j}(e_j)$, $K_{I,j}(e_j)$, $K_{D,j}(e_j)$ to the SIDF inversion routine for controller nonlinearity synthesis; and
7. validate the nonlinear controller design by evaluating its frequency response (via SIDFGEN) and by transient-response simulation.

This approach results in designing a nonlinear control system whose frequency-domain response has as little amplitude sensitivity as possible. This appears, on the basis of a number of applications, to minimize transient response sensitivity as well. Implementing this approach requires frequency-domain design methods (for which we have not designed tools), linear gain-set evaluation (the first tool below), and nonlinearity synthesis (second tool below).

3.1 Linear gain-set evaluation

This technique proceeds as follows: The input for this procedure is frequency response data for G_k and $CG^*(j\omega)$. For each error signal amplitude e_j, frequency ω_m and gain-set K_P, K_I, K_D the plant input amplitude is calculated, $a_j = |C(j\omega)| \cdot e_j$ and the corresponding plant frequency response is obtained by interpolating over the $G(j\omega, a_k)$ data ("G-surface"). An iterative minimization procedure is then used (MINPACK [2]) to adjust the linear gain-set until the objective $CG^*(j\omega)$ has been fitted as well as possible over the frequency range $[\omega_l, \omega_u]$ under a mean-square error criterion. The details are omitted for the sake of brevity; for further information see [4]. As mentioned previously, the output of this procedure is a set of error amplitude-gain pairs $K_{P,j}(e_j)$, $K_{I,j}(e_j)$ $K_{D,j}(e_j)$ from which the compensator nonlinearities are synthesized.

3.2 Controller Nonlinearity Synthesis

For each value of e_j there exists a linear compensator design that differs only in the gain values $K_{P,j}$, $K_{I,j}$, $K_{D,j}$. This information serves as the basis for nonlinear controller synthesis on an element-by-element basis (P, I, D), as follows: Given the gain versus amplitude relation of the form $K_j(e_j)$ that is to be achieved by a single nonlinearity, adjust the parameters of a specified piecewise-linear nonlinear function $f_{pwl}(e)$ so that the SIDF of f_{pwl} provides the best fit to the gain/amplitude relation $K_j(e_j)$ in the minimum-mean-square-error sense. We defined f_{pwl} in fairly general terms as shown in Fig. 2; the parameters to be adjusted are the slopes, S_1, S_2, the breakpoint, δ, and the step discontinuity, D.

The user is allowed to restrict the nonlinearity by fixing any of the parameters of f_{pwl}; only free parameters will be adjusted for mean square error minimization. For example, fixing

$$S_1 = 0.0, \quad S_2 = 0.0$$

restricts f_{pwl} to being a relay with deadzone, of arbitrary output level D and deadzone width 2δ. The parameters shown in Fig. 2 allow enough degrees of freedom that most reasonable gain/amplitude relations can be fit with decent accuracy; allowing the designer to restrict this freedom should permit the user to arrive at a practical design.

In summary, denoting the parameters that define $f_{pwl}(e)$ by

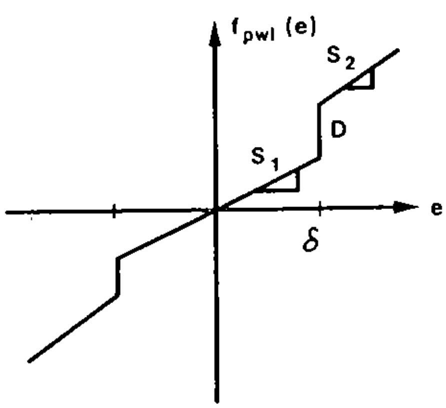

Fig. 2: Nonlinear Class for SIDF Inversion

the vector p,

$$p^T = [S_1, S_2, \delta, D] \tag{5}$$

we developed a routine for evaluating the SIDF of $f_{pwl}(e, p)$, and the parameter adjustment is done using MINPACK [2]. The output of MINPACK is the parameter set p^* for the nonlinearity in the controller, which completes its definition. It should be observed that the gain set $K_j(e_j)$ must be well-conditioned, in the sense that the values of e_j must cover a reasonable range of e (e.g. $e_j =$ 1.2, 3.6, 7.2, 10.8 rather than $e_j =$ 1.2, 1.35, 1.50, 1.65); the reason for this is that the SIDF of the nonlinearity f_{pwl} cannot change abruptly with small amplitude changes, so a closely-spaced gain set will usually be meaningless. Also, the user should be aware that quite different nonlinearities may have very similar SIDFs (e.g., the SIDF for a 3-level quantizer is almost identical to the SIDF of a limiter for all normalized input amplitudes greater than 1.5). Therefore, the user may have considerable latitude in choosing the nonlinearity to implement. This also means that there are local minima – the fitting process should be done interactively with careful consideration of the results and alertness for spurious results (e.g., δ being adjusted to be beyond the maximum e_j).

4 Nonlinear Control Synthesis for the ATB1000

In this section, we will illustrate the use of the software by designing a nonlinear control for the ATB1000, an Army test facility. The ATB1000 is a nonlinear system that involves hard nonlinearities (backlash, Coulomb friction) and flexible modes. This challenging system is a good benchmark problem for testing the nonlinear control design methodology discussed in this paper.

As shown in Fig. 3, the testbed consists of

- a drive sub-system, including a DC motor (with nonlinear friction), a gear chain (with backlash), and an elastic shaft;
- a wheel/barrel sub-system, including an inertia wheel (with nonlinear friction) and a flexible gun barrel.

Readers are referred to paper[7] for a detailed description of the testbed model.

The objective is to control the inertia wheel angle θ_i so that it will have a satisfactory response to input signals (see Fig. 4 for the control diagram).

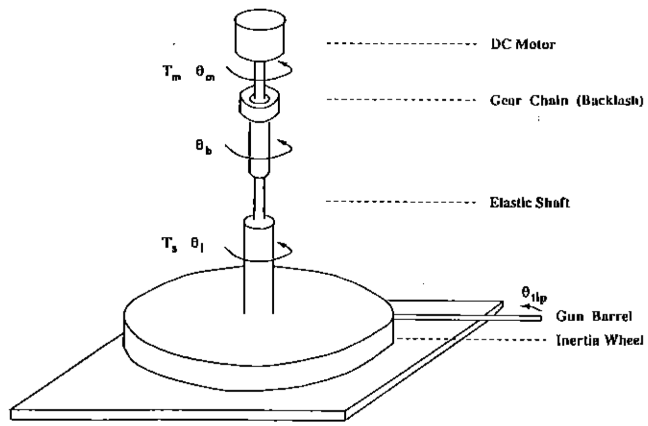

Fig. 3: Schematic of a Tank Turret System

1. Generating $G(j\omega;a)$.

First we select the ranges for the amplitudes and frequencies of the input sinusoidal functions. We found that when the amplitude of the input is smaller than 0.55 (volt), the system response is virtually zero; when the amplitude of the input is large than 5.5 (volt), the system response is close to a linear response ($G(j\omega;a)$ does not depend on the amplitude). Therefore, we choose the range for the input amplitude to be $\mathcal{A} = [0.55, 5.5]$. We chose the frequency range as $\omega \in \mathcal{W} = [3.1416, 94.25]$. The $G(j\omega;a)$ for selected a's $\in \mathcal{A}$ are plotted in Fig. 5.

2. Control Design.

We will give the results of designing a nonlinear control for ATB1000 by following the nonlinear control synthesis procedure in Section 3.

a. We select one of the I/O models $G^*(j\omega;a^*)$ $(a^* \in \mathcal{A})$ as a nominal model (see Fig. 6). This nominal model is chosen because the associated input amplitude is an intermediate value in $\mathcal{A}$.

b. Then, we design a PID compensator $C^*(j\omega)$ for the nominal model to achieve desired "shape" of the $CG^*(j\omega;e^*)$ in terms of gain and phase margins, where $CG^*(j\omega;e^*)$ is the SIDF I/O relation between the input to the compensator and the output of the plant that follows the compensator, and e^* is a value chosen to be consistent with a^* and the PID gain near the crossover frequency, denoted $|C_{co}|$, i.e., $e^* = a^*/|C_{co}|$. Plots of $CG^*(j\omega;e^*)$ are also shown in Fig. 6.

c. We next select a set of compensator input amplitudes $\{e_j\}$. Then we design a set of linear compensators $\{C_j(j\omega)\}$ so that the error

$$E(j\omega) = 1 - C_j(j\omega)G(j\omega;e_j|C_j(j\omega)|)/CG^*(j\omega,e^*) \quad (6)$$

over the frequency set $\omega \in \mathcal{W}$ is minimized in the mean square sense; this yields a set of PID parameters for each values of e_j, denoted $\{K_P(e_j)\}$, $\{K_I(e_j)\}$, $\{K_D(e_j)\}$ (see Fig. 7).

d. Use the sets of PID parameters $\{K_P(e_j)\}$, $\{K_I(e_j)\}$, $\{K_D(e_j)\}$ with SIDF inversion to synthesize the nonlinearities $\{f_P(e)\}$, $\{f_I(e)\}$, $\{f_D(e)\}$ (see Fig. 8).

e. Validate the nonlinear controller design via simulation. Fig. 9 shows the step responses of the closed loop system to step inputs with different amplitudes. To compare, the step responses of the closed loop system with a linear PID control to step inputs with the same amplitudes are shown in Fig. 10.

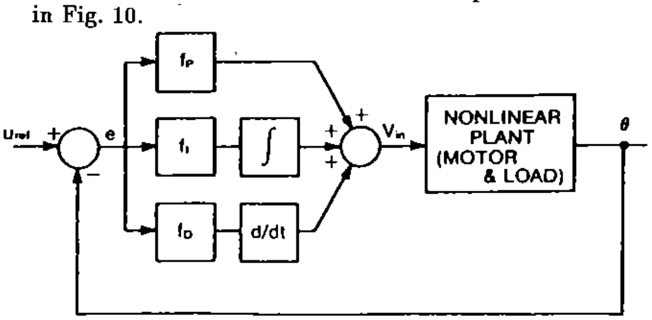

Fig. 4: Nonlinear Control System

5 Conclusion

We have described a second-generation toolset for computer-aided nonlinear control system analysis and design based on sinusoidal-input describing function (SIDF) methods. This software can treat very general nonlinear systems, with no restrictions as to system order, number of nonlinearities, configuration, or nonlinearity type. An overview of the application of this software to the ATB1000 Army test facility illustrated the use and efficacy of these tools.

References

[1] Guckenheimer, J. and A. Nerode. (1992). Simulation of Hybrid Systems and Nonlinear Control. *Proc. IEEE Conf. on Decision and Control*, Tucson, AZ, 2980-2981

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

[3] Taylor, J. H. (1983). A systematic nonlinear controller design approach based on quasilinear system models. *Proceedings of the American Control Conference*, San Francisco, CA, 141-145.

[4] Taylor, J. H., and K. L. Strobel. (1985). Nonlinear Compensator Synthesis via Sinusoidal-Input Describing Functions. *Proc. American Control Conference*, Boston MA, 1242-1247.

[5] Taylor, J. H. (1985). Computer-Aided Control Engineering Environment for Nonlinear Systems. *Proc. Third IFAC Symposium CAD in Control and Engineering Systems*, Lyngby, Denmark, 38-43.

[6] Walker, R. A., C. Z. Gregory Jr., and S. Shah. (1982). MATRIXx: A Data Analysis, System Identification, Control Design and Simulation Package. *IEEE Control Systems Magazine*, Vol. 2, pp. 30-37.

[7] Taylor, J. H. and Jin Lu, "Application of an SIDF-Based Nonlinear Control System Synthesis Method to an Electro-Mechanical Pointing System," *Proceeding of the 1993 American Control Conference*, San Francisco, CA, 1993.

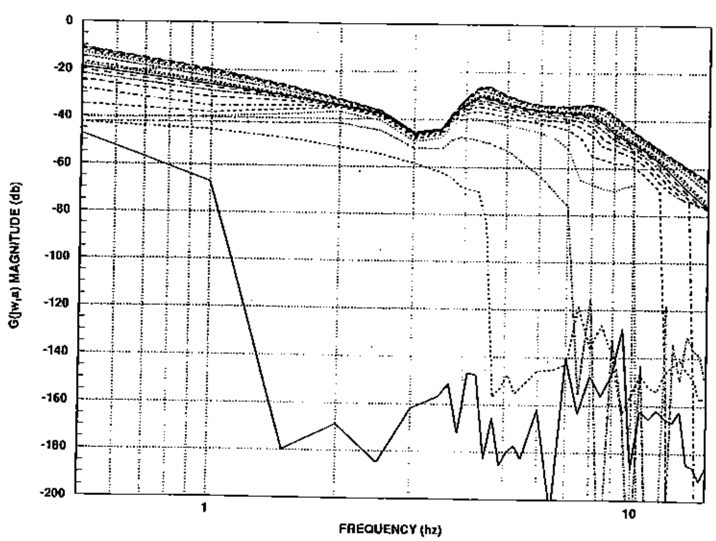

Fig. 5(a): $|G(j\omega,\ a)|$ Plots for Selected $a \in \mathcal{A}$.

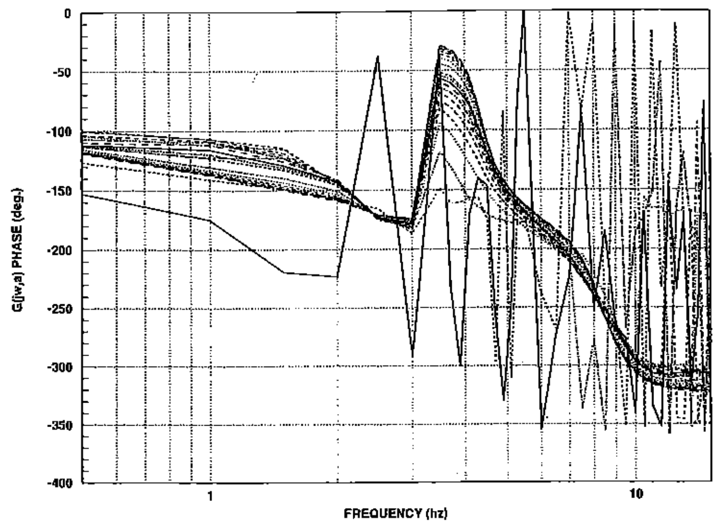

Fig. 5(b): $\angle G(j\omega,\ a)$ Plots for Selected $a \in \mathcal{A}$.

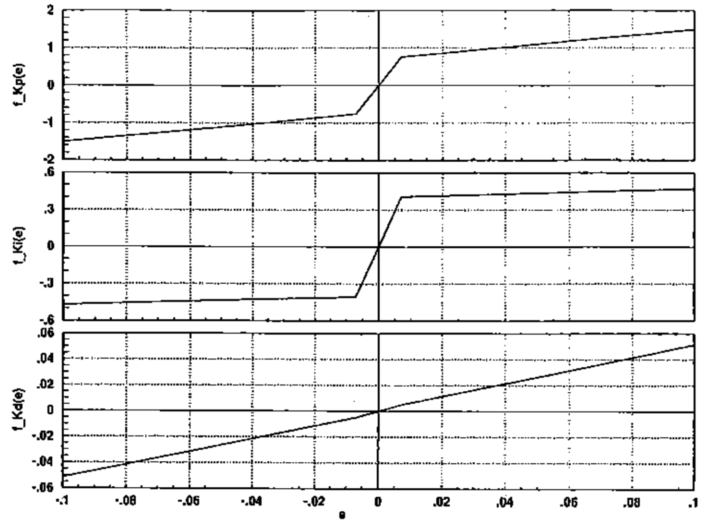

Fig. 8: Results of SIDF Inversion

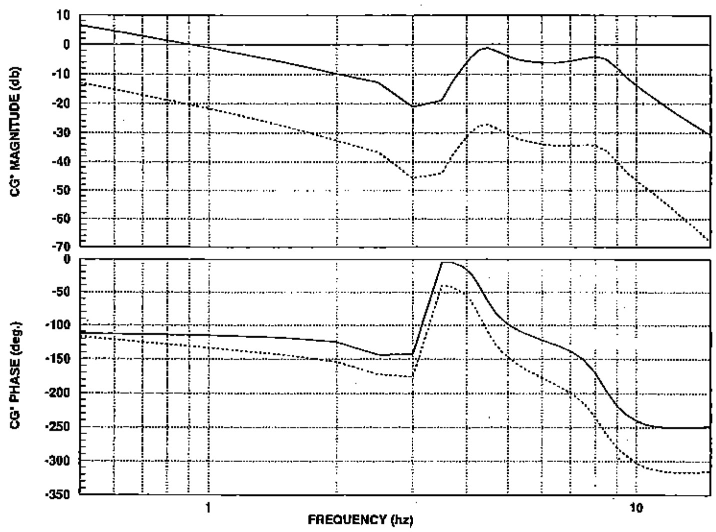

Fig. 6: —: CG^* ($K_P = 14, K_I = 4, K_D = 0.5$)
$\cdots$: $G^*(j\omega, a^*)$, $a^* = 3.5$

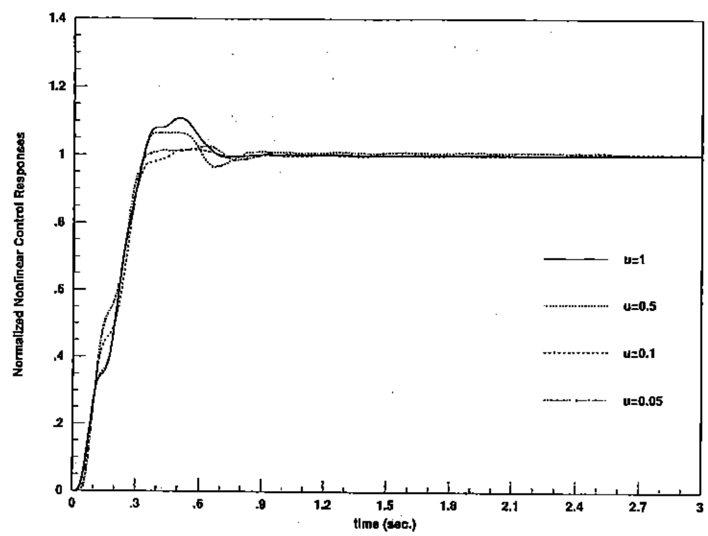

Fig. 9: Responses of Nonlinear PID Control to Step Inputs u.

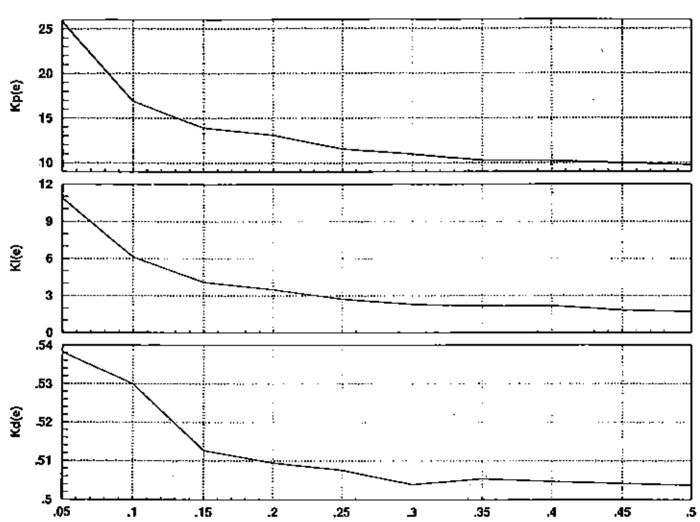

Fig. 7: Results of Fitting $C(e, K_P, K_I, K_D) * G(j\omega, a)$ to CG^*.

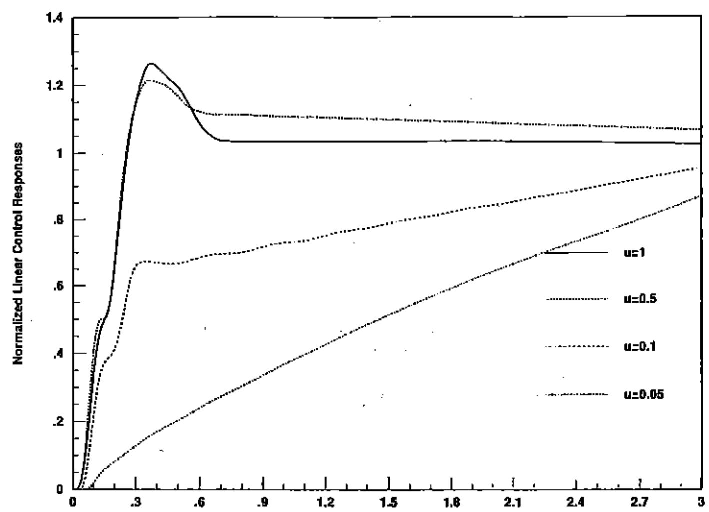

Fig. 10: Responses of Linear PID Control to Step Inputs u ($K_P = 14, K_I = 4, K_D = 0.5$).