Chapter 3

Experimental Identification

In the previous chapter, we derived the fixed points of the modulation equations. These equations give us a way to theoretically relate the input and output signals and the phase between them. Next, we need to be able to determine these variables (a_{base}, a, and γ) experimentally. This will allow us to find values for the parameters of the system that minimize the difference between the experimental and theoretical fixed points of the modulation equations. We will first discuss how to determine a_{base}, a, and γ experimentally. We will then present several curve-fitting algorithms that attempt to estimate the system parameters. As these algorithms evolve, we will eventually arrive at a reasonably fast algorithm that includes all known physical constraints. This algorithm will estimate the system parameters based on data from one amplitude sweep and one frequency sweep. To validate the algorithm, we use the parameters it gives to predict the force-response curves at two nearby frequencies.

3.1 Curve Fitting of Periodic Solutions by Use of Phase Lag

Equations (2.22) and (2.23) can be rewritten as

$$a_{base}\sin\gamma = \kappa_1 a + \kappa_2 a^2 \tag{3.1}$$

$$a_{base}\cos\gamma = \kappa_3 a + \kappa_4 a^3 \tag{3.2}$$

where

$$\kappa_1 = \frac{\Omega\mu_1}{\eta_e},\ \kappa_2 = \frac{4}{3\pi\eta_e}\mu_2\Omega,\ \kappa_3 = -\frac{\sigma}{2\eta_e},\ \kappa_4 = -\frac{1}{8\eta_e}(2\Omega^2\delta - 3\alpha) \tag{3.3}$$

By treating the products $a_{base}\sin\gamma$ and $a_{base}\cos\gamma$ as the dependent variables, we can determine the coefficients κ_n by applying a curve-fitting algorithm to Equations (3.1) and (3.2). In order to do this, we must be able to measure a, a_{base}, and γ.

The experimental setup is shown in Figure 3.1. The system consists of a cantilevered steel beam 33" long with a rectangular cross-section that is $\frac{5}{8}$"x$\frac{1}{32}$". It is oriented vertically and clamped at the bottom. The beam is excited by a 100-lb. shaker. The shaker is driven by an amplifier which is controlled by a signal analyzer (DSA). The input and output signals are monitored with the DSA and recorded using a PC.

An accelerometer attached to the shaker head is used to measure a_{base}. The response amplitude a can be determined from a strain gauge attached to the beam. The strain-gauge output voltage can be converted to displacement amplitude by converting the voltage intermediately to strain, from strain to stress, from stress to bending moment, and from bending moment to the displacement amplitude by solving the following equations sequentially:

$$e = \frac{4V_{out}}{V_{excite}GK_g} \tag{3.4}$$

$$\sigma_{11} = Ee \tag{3.5}$$

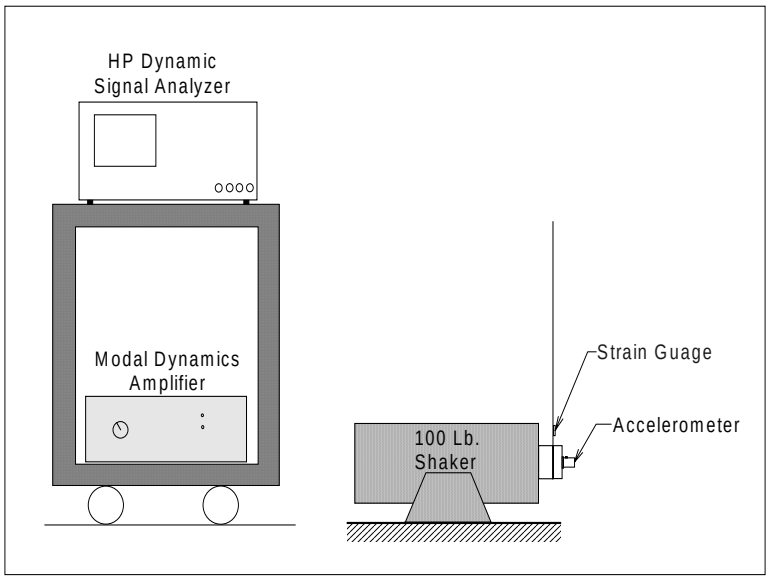

Figure 3.1: Diagram of the setup of the experiment.

$$M = \frac{\sigma_{11} c}{I} \tag{3.6}$$

$$M = \frac{EIa\phi''}{[1 + (a\phi')^2]^{\frac{3}{2}}} \tag{3.7}$$

where e is the strain, V_{out} and V_{excit} are the output and excitation voltages of the strain gauge, G is the gain of the strain-gauge signal conditioner, K_g is the gauge factor, σ_{11} is the stress, E is Young's Modulus, M is the fully nonlinear bending moment, c is one-half the thickness of the beam, I is the moment of inertia about the axis perpendicular to the plane of vibration, $\phi(s)$ is the normalized mode shape, and s is the arc length. Since ϕ' and ϕ'' are known at the location of the strain gauge, Equation (3.7) can be solved for a. Finally, the phase angle γ between these two signals can be found in several ways. This can be done easily in the frequency domain by either subtracting the respective phase angles from the complex-valued FFT's of the signals at the known input frequency or by taking the phase from the cross-power spectrum at that same frequency.

The data are collected using a sine-dwell test method. The shaker is driven by a

constant amplitude and frequency sine wave. We wait until the input and output signals reach steady state, record time traces of the data, and then increment either the forcing level or the excitation frequency, depending on the type of test we are doing. Once we have captured time traces over the desired range of the control parameter, we convert the time-domain signals to the frequency domain and extract the amplitudes of the input and output as well as the phase between them.

3.1.1 Problems in Phase-Lag Measurement

In practice, several problems arose in trying to determine γ. Phase shifts were introduced by low-pass filters. In theory, such filters should be usable as long as consideration is given to their effect on the data being recorded. Since we are concerned only with the difference in the phase, only relative shifts in the phases will cause problems. The largest problem encountered with our system was with using the filter on the strain-gauge signal conditioner, where only the output signal would be filtered. This introduced significant relative phase shifts. With the cutoff frequencies set high enough and with both signals being filtered, the shift of the input and output signals relative to one another may not be significant.

Another source of phase error considered was that introduced by other structural modes. The theoretical model assumes that, since the natural frequencies of the structure are widely spaced and no internal resonances are activated, the structure can be treated as a single-degree-of-freedom system within some range of the driving frequency around each natural frequency. The phase shift caused by other modes asymptotically decays to zero as the driving frequency moves away from the natural frequency for each mode. For a linear,

single-degree-of-freedom system, the phase lag between input and output is described by

$$\gamma = \tan^{-1}\left(\frac{2\zeta\beta}{1-\beta^2}\right) \tag{3.8}$$

where $\beta = \frac{\omega}{\Omega}$ is the ratio of the natural frequency to the driving frequency and ζ is the damping ratio. This model assumes that the phase lag will be practically zero far enough below resonance. That was not the case for this system. The effect of other modes on the phase could not be ignored. This was evident in the variation of the phase from that predicted by Equation (3.8), particularly at the lowest driving frequencies considered (i.e. those with the smallest value of β).

A final source of phase error considered was that caused by the time delay associated with multiplexing during sampling. If two channels are being recorded at 200 Hz, the data actually recorded by the computer is the same as if one channel were being sampled at 400 Hz. Subsequently, the second channel is sampled at one-half of a time step (i.e., $\frac{1}{400}$ of a second for this example) later than the first.

3.1.2 Development and Application of the Curve-Fitting Algorithm

If a_{base}, a, and γ can be measured experimentally, then Equations (3.1) and (3.2) can be used to extract the coefficients κ_1, κ_2, κ_3, and κ_4. The curve-fitting algorithm used must take into consideration that the product $a_{base}\sin\gamma$ is constrained to be a function only of a and a^2. Similarly, the product $a_{base}\cos\gamma$ is constrained to be a function only of a and a^3. A curve-fitting algorithm that insists on using the form $a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0$ will not work in this case. While the least-square error may be reduced by including, for example, a constant term, there is no physical basis for such a term.

Matlab contains a built-in function called **fmins** which can be used to this end. In

order to use this function, the user must define a function to minimize and tell Matlab which coefficients it can vary. **Fmins** will then return a vector containing the coefficients which provide a local minimum for the function specified near the initial guesses for the coefficients. Following Equations (3.1) and (3.2), we seek to minimize the error functions:

$$E1 = \sum_{i=1}^{N} \Big((a_{base})_i \sin\gamma_i - (\kappa_1 a_i + \kappa_2 a_i^2) \Big)^2 \tag{3.9}$$

$$E2 = \sum_{i=1}^{N} \Big((a_{base})_i \cos\gamma_i - (\kappa_3 a_i + \kappa_4 a_i^3) \Big)^2 \tag{3.10}$$

Determining κ_3 from the curve fit and knowing σ allows η_e to be computed. Knowing η_e allows the rest of Equations (3.3) to be solved for μ_1, μ_2, and the effective cubic nonlinearity coefficient $-\frac{1}{3}(2\delta\Omega^2 - 3\alpha)$.

The parameters found in this way can now be used in the theoretical model to generate curves for the variation of $a_{base}\sin\gamma$ and $a_{base}\cos\gamma$ with a. Figures 3.2 and 3.3 show that, while these curves seem to capture the qualitative trends in the data, the quantitative agreement is poor. There is an apparent discontinuity in the data before and after the jump in the response amplitude ($a < 3\text{x}10^{-3}$ versus $a > 3\text{x}10^{-3}$). This may be seen by visualizing a curve that fits smoothly through the small amplitude data of Figure 3.2. If this curve were projected out, the values for $a_{base}\sin\gamma$ near $a = 5\text{x}10^{-3}$ would be badly overpredicted.

Force- and frequency-response curves can be predicted by squaring and adding Equations (3.1) and (3.2). This will also allow the possible benefit of eliminating γ from the curve-fitting algorithm to be investigated. The predicted response curves are shown along with the experimental data in Figures 3.4 and 3.5. We are concerned not only with the accuracy of the predicted response amplitudes but also with the locations of the bifurcation points. Figures 3.4 and 3.5 show that the parameters found in this way give reasonable estimates for the jump points on the force-response curve as well as the jump-up point on

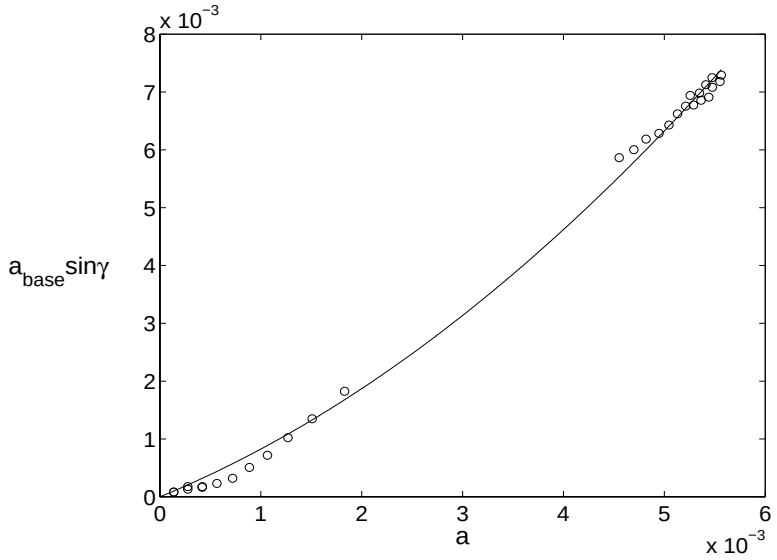

Figure 3.2: Experimental and theoretical fixed points: variation of $a_{base} \sin\gamma$ with a.

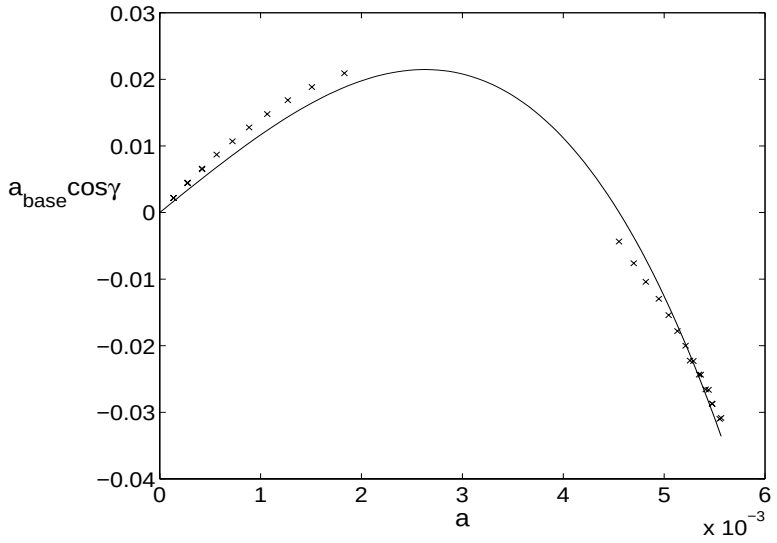

Figure 3.3: Experimental and theoretical fixed points: variation of $a_{base} \cos\gamma$ with a.

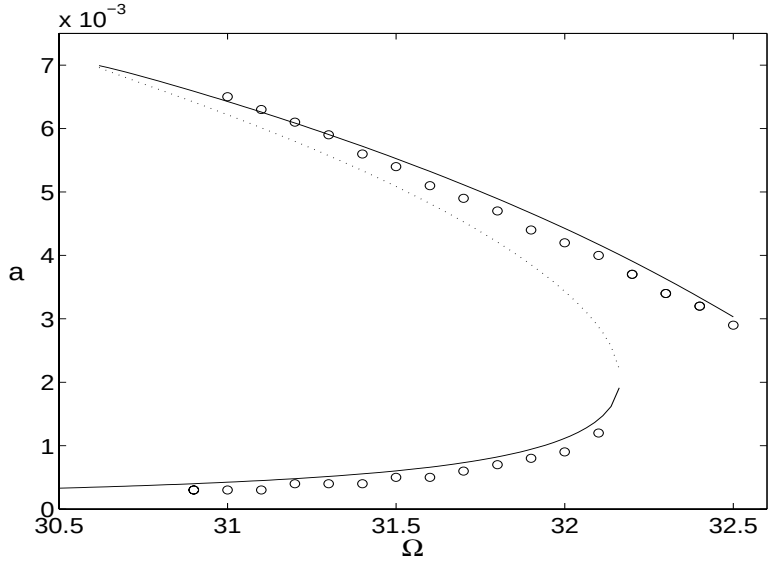

Figure 3.4: Experimental and theoretical frequency-response curves, $a_{base} = 1g$. The parameters of the theoretical curve were determined using the fixed points of the multiple-scales solution and the phase lag between the input and output.

the frequency-response curve. The jump-down point on the frequency-response curve is poorly estimated. The predicted amplitudes are generally good except at low amplitudes. The discrepancy at low amplitudes leads to an imprecise estimate of the linear damping, one that is twice the value determined from linear identification techniques.

Even after all of the known sources of phase error were removed or corrected for, there was still some uncertainty in the measurement, particularly at high response amplitudes. Since determining γ is not an explicit concern, a curve-fitting algorithm that does not involve γ would seem advantageous.

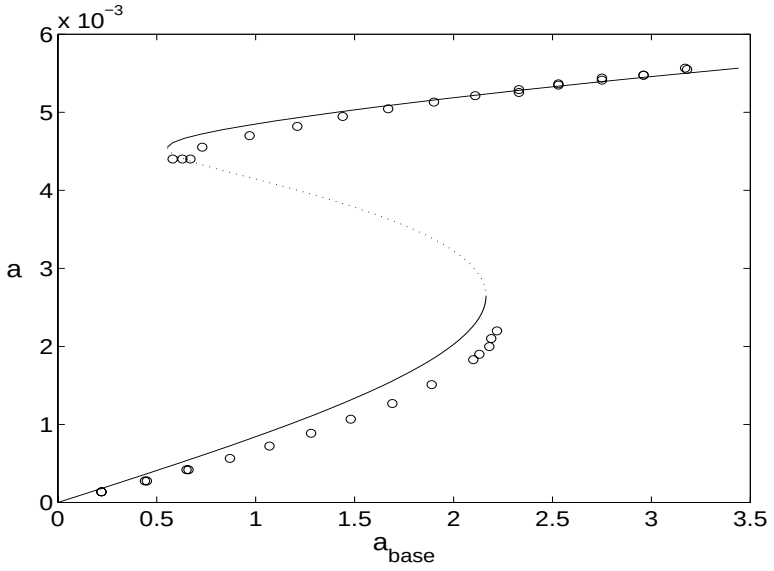

Figure 3.5: Experimental and theoretical force-response curves, Ω=31.8 Hz. The parameters of the theoretical curve were determined using the fixed points of the multiple-scales solution and the phase lag between the input and output.

3.2 Amplitude- and Frequency-Sweep Curve Fitting

In order to circumvent any problems or potential errors in determining γ, we eliminate the phase from the curve-fitting algorithm. We do this by finding the frequency-response equation from squaring and adding Equations (3.1) and (3.2). By holding Ω constant, we can predict the force-response curve by solving the frequency-response equation for a as a function of a_{base}. Similarly, by holding a_{base} constant and varying Ω, we can predict the frequency-response curve. Comparing these curves with experimental data will form the basis for the next group of curve-fitting algorithms.

3.2.1 Problems with the Least-Squared-Error-Only Criterion

A first attempt at a curve-fitting algorithm that does not involve γ can be made by finding the coefficients of

$$a_{base}^2 = (\kappa_1^2 + \kappa_3^2)a^2 + 2\kappa_1\kappa_2 a^3 + (\kappa_2^2 + 2\kappa_3\kappa_4)a^4 + \kappa_4^2 a^6 \tag{3.11}$$

such that the square of the error is minimized. This was originally done by applying the **fmins** algorithm to Equation (3.11). The coefficients found in this way were not physically realizable. For example κ_1 and κ_2 are known to be positive. Some of these kinds of problems can be avoided by writing Equation (3.11) in terms of physical parameters by substituting in Equations (3.3).

Two problems were encountered using this algorithm. The first was that the curve-fitting algorithm does not realize which branches of the amplitude- and frequency-sweep plots are stable. The algorithm is concerned only with the distance between an experimental point and the nearest point on the theoretical curve and does not consider whether the point it is using on the theoretical curve is stable or not. The second problem was that the algorithm does not realize that the range in which the bifurcation happens is bracketed by points on opposite sides of the jumps in the response amplitude.

3.2.2 Incorporating Branch Stability

It was not immediately apparent how to incorporate branch stability into the algorithm based on using **fmins** and Equation (3.11). The only way we saw to do that at the time was to solve for all three positive roots of Equation (3.11) for each value of the control parameter (Ω or a_{base}) for which multiple solutions exist. If all three possible values for a are known, the largest and smallest positive roots are stable and the second largest is

unstable. An algorithm was written in Matlab that used a_{base} and Ω as the independent variables. The unknown coefficients are varied in pursuit of the combination that gives the least-squared error. For each combination of the unknown coefficients, Equation (3.11) was solved for the six possible roots of a. The three negative roots were ignored, and the stability of the positive roots was determined. Eliminating the unstable branch from consideration by the algorithm left the algorithm with either one or two choices of response amplitude, depending on whether or not the value of Ω or a_{base} was such that there were two stable solutions for a. The error for each data point is the difference between the known experimental value of the response amplitude and the nearest stable predicted value. This approach uses the known experimental value to determine which branch of the theoretical curve to use. The error we seek to minimize is then the squared sum of the differences in the stable theoretical and the experimental response amplitudes. Figures 3.6 and 3.7 show predicted and experimental force- and frequency-response curves for an algorithm that uses the squared error as the only curve-fitting criterion. If we compare Figures 3.4 and 3.6, we see that the amplitude- and frequency-sweep algorithm leads to more accurate prediction of the small amplitude portion of the frequency-response curve. There is little difference in the jump-up point and the magnitude of the large-amplitude branch between the two figures. The estimate of the jump-down point is clearly worse with the amplitude- and frequency-sweep results. Making a similar comparison between Figures 3.5 and 3.7, we see that the results from the amplitude- and frequency-sweep algorithm predict the curve better at small amplitudes, but this is a trade off with accuracy in the large-amplitude region ($a > 4\text{x}10^{-3}$). The new algorithm leads to worse estimates of the jump points. Improvements will have to be made if this algorithm is to be considered better than the previous one.

A similar approach would be to follow Yasuda, Kamiya, and Komakine (1997) and

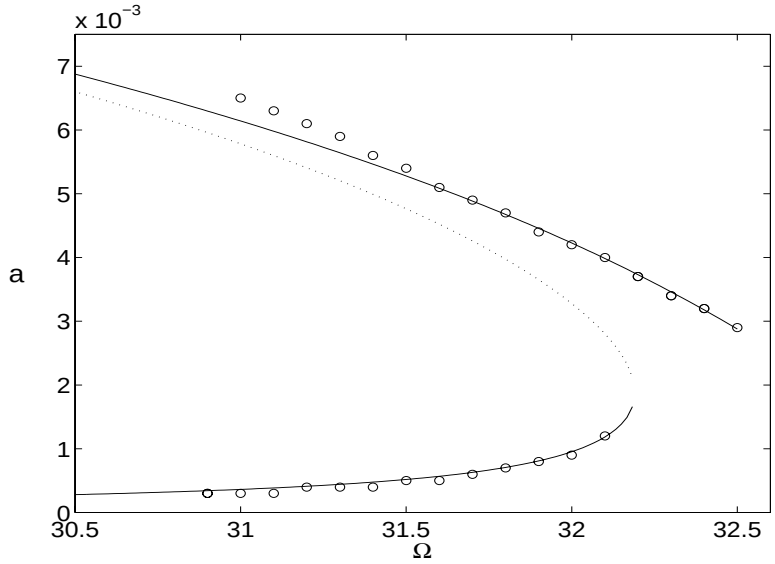

Figure 3.6: **Experimental and theoretical frequency-response curves, $a_{base} = 1g$. The theoretical curve uses parameters found by applying a least-squares algorithm to the amplitude of the multiple-scales approximate solution (the phase lag has been removed from the formulation).**

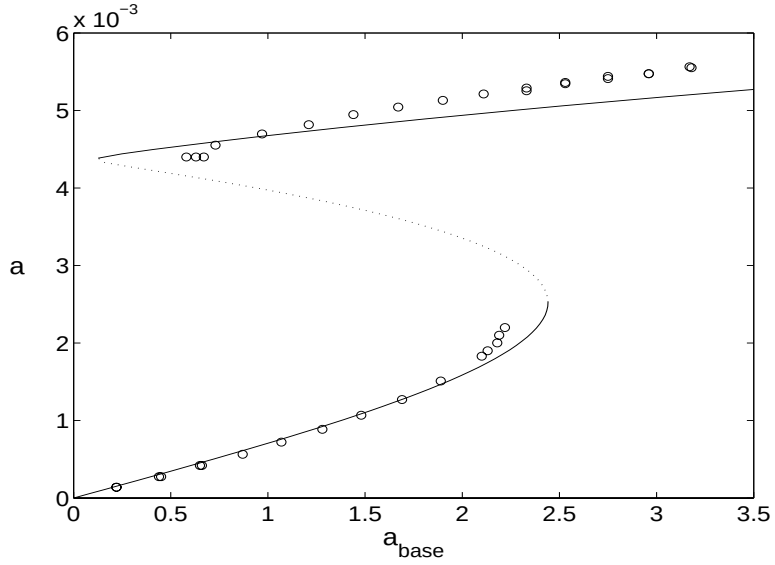

Figure 3.7: **Experimental and theoretical force-response curves, Ω=31.8 Hz. The theoretical curve uses parameters found by applying a least-squares algorithm to the amplitude of the multiple-scales approximate solution (the phase lag has been removed from the formulation).**

assume a harmonic-balance solution to the equation of motion

$$\ddot{u} + \omega^2 u + \delta(u\dot{u}^2 + u^2\ddot{u}) + \alpha u^3 + 2\mu_1\dot{u} + \mu_2\dot{u}|\dot{u}| = \eta_e a_{base}(e^{i\Omega t} + e^{-i\Omega t}) \tag{3.12}$$

in the form

$$a_{base} = a_0 + a_1\cos\Omega t + a_2\cos 2\Omega t + \ldots + a_1^*\sin\Omega t + a_2^*\sin 2\Omega t + \ldots$$

$$u = u_0 + u_1\cos\Omega t + u_2\cos 2\Omega t + \ldots + u_1^*\sin\Omega t + u_2^*\sin 2\Omega t + \ldots \tag{3.13}$$

The coefficients a_n, a_n^*, u_n, and u_n^* are found by applying an FFT algorithm to the respective signals. Knowing the Fourier coefficients of u allows the Fourier coefficients of the rest of the terms in Equation (3.12) to be determined. Substituting the Fourier expansions into Equation (3.12) and equating each of the coefficients of $\cos n\Omega t$ and $\sin n\Omega t$ on both sides gives a system of equations for the parameters we are trying to identify: ω, δ, μ_1, μ_2, and η_e. Applying this method at various values of Ω and a_{base} allows a least-square algorithm to be used to estimate the parameters. Figure 3.8 shows experimental and theoretical frequency-response curves for an algorithm that applies a harmonic-balance formulation to the frequency-sweep data only. Figure 3.9 shows experimental and theoretical force-response curves found using the same approach on amplitude- and frequency-sweep data. These plots show some of the same problems with predicting jump points that we saw in Figures 3.6 and 3.7.

Figures 3.6-3.9 show that the jump-down points in both the amplitude- and frequency-response curves are poorly predicted by algorithms that use the least-square error as the only criterion for the curve fit. We seek to correct this as the next step in the development of the curve-fitting algorithm.

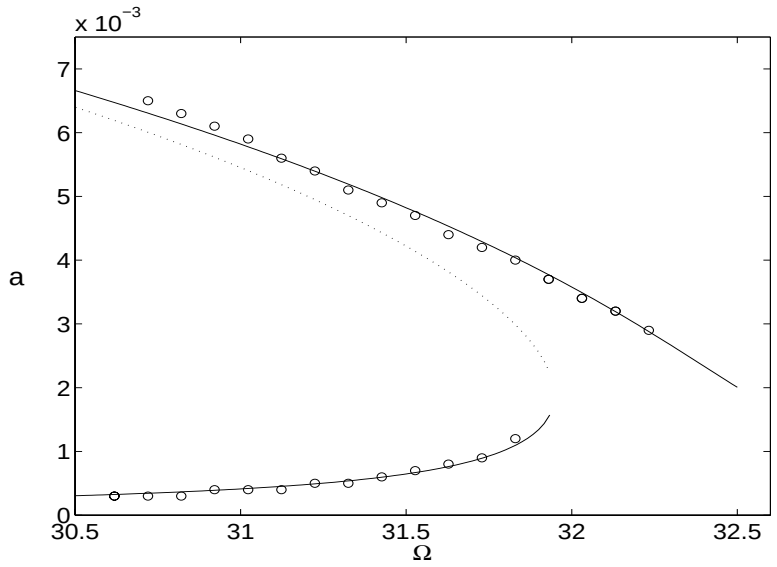

Figure 3.8: Experimental and theoretical frequency-response curves, $a_{base} = 1g$. The theoretical curve uses parameters found by applying a least-squares algorithm to a harmonic-balance approximate solution.

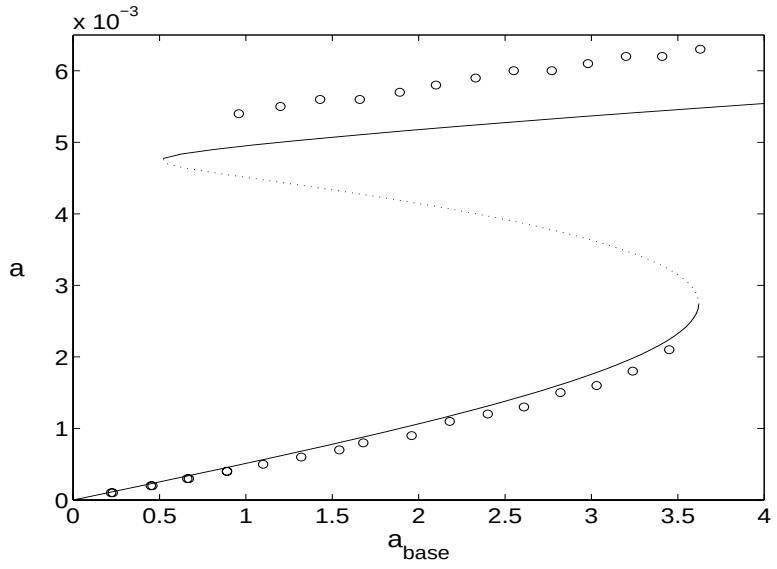

Figure 3.9: Experimental and theoretical force-response curves, $\Omega = 31.5$Hz. The theoretical curve uses parameters found by applying a least-squares algorithm to a harmonic-balance approximate solution.

3.2.3 Including Bifurcation Range Penalties

This inaccurate prediction of the location of the bifurcation points can be corrected by including, in the curve-fitting algorithm, the information that the bifurcation is known to occur within a small range of the control parameter. An example of this is shown in Figure 3.10. A saddle-node bifurcation is known to occur between the two vertical lines. The data is collected by performing sine-dwell tests beginning with the data point in the lower left corner ($a_{base} \approx 0.25$ and $a \approx 0.2\text{x}10^{-3}$). After waiting for the response to reach steady state and collecting the data, a_{base} is increased. With each increase in the control parameter, the steady-state response moves from one diamond to the next. Once the desired maximum value for a_{base} is reached, we begin decreasing the excitation level in a similar fashion. The data for this reverse sweep is shown with circles. When a_{base} is increased from approximately 2.3 to 2.4, the lower branch of the curve becomes unstable and the response amplitude jumps to the upper branch. Thus, the bifurcation range is bracketed by data points on opposite sides of the amplitude jump. The curve-fitting algorithm must be told that the data is organized in forward and reverse sweeps and that a bifurcation has occured in the range between the vertical lines. The most effective way to include this information in the algorithm is to require that the stable and unstable branches of the predicted curve meet and bifurcate within the known range.

The algorithm that finds all possible roots for a lends itself to enforcing this bifurcation restriction. Since all six roots are found at each value of the control parameter, what happens to each root can be easily monitored. When a saddle-node bifurcation occurs, the polynomial roots corresponding to the saddle and the node approach the same real value, meet, and then become complex. Since the bifurcation is known to occur within a given range, the bifurcation range can be enforced by requiring that the corresponding polynomial roots switch from real-valued to complex-valued somewhere within that same

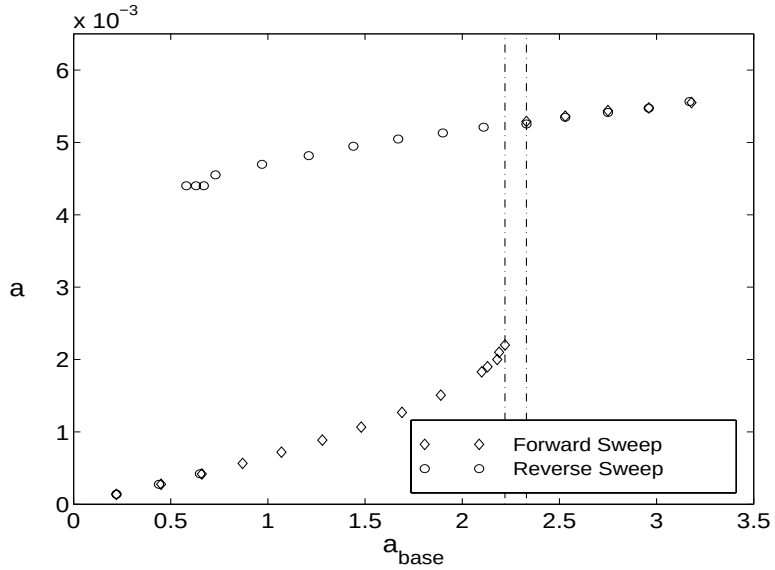

Figure 3.10: Experimental forcing-sweep data with the known bifurcation range shown. This physical constraint is included to improve the curve-fitting algorithm.

range. Numerically this requirement is enforced by adding a large penalty to the error if the bifurcation occurs outside of the known range.

This algorithm, based on minimizing the error while requiring that the data points be mapped onto stable branches and the bifurcations occur within experimentally known ranges, was originally applied to the amplitude- and frequency-sweep data separately. From these separate curve fits it became apparent that the force- and frequency-response curves are each particularly sensitive to different parameters. For example, μ_2 is best determined from the frequency-sweep data while η_e and the effective cubic nonlinearity coefficient are best determined from the amplitude-sweep data. From this information, the algorithm can be further improved by fitting the frequency and amplitude data simultaneously. While doing these simultaneous curve fits, it became apparent that enforcing the frequency jump-down and amplitude jump-up points causes the other jump points to be predicted accurately.

The final algorithm of this type simultaneously fits data from one frequency sweep and one amplitude sweep while requiring that the data points be mapped onto stable branches and that the forcing jump-up and frequency jump-down points occur within known ranges. The success of the algorithm is evaluated by using the parameters it finds to predict the force-response curves at two nearby input frequencies. Figures 3.11 and 3.12 show the predicted and experimental response curves for the data used by the algorithm. By comparing Figures 3.6 and 3.11, we see that enforcing the bifurcation range has had no appreciable effect on the accuracy of the rest of the frequency-response curve. This is also the case for the force-response curves (Figures 3.7 and 3.12). If we compare Figures 3.11 and 3.12 with Figures 3.4 and 3.5, we see that the amplitude- and frequency-sweep algorithm with the bifurcation ranges enforced leads to an improvement over the algorithm that used the phase lag between the input and output. Figures 3.13 and 3.14 show the predicted and experimental force-response curves used for evaluation. The jump points on these plots are well predicted and the accuracy of the magnitudes seems reasonable. We note that the low-amplitude predictions are better than before. This improvement at low amplitudes leads t a more precise estimate of the linear damping, one that is within 6% of the half-power linear estimate.

While this approach allowed incorporating these physical restrictions in an intuitive fashion, it also has several drawbacks. It is very sensitive to initial conditions. Enforcing the bifurcation ranges requires that the given initial guess satisfies all of the bifurcation restrictions. It also limits the curve fit to combinations of parameters that satisfy the restrictions. This approach also requires finding the roots of the sixth-order polynomial for each experimental data point and each combination of the parameters we seek to estimate. As a result, this algorithm is very computationally intensive and time consuming.

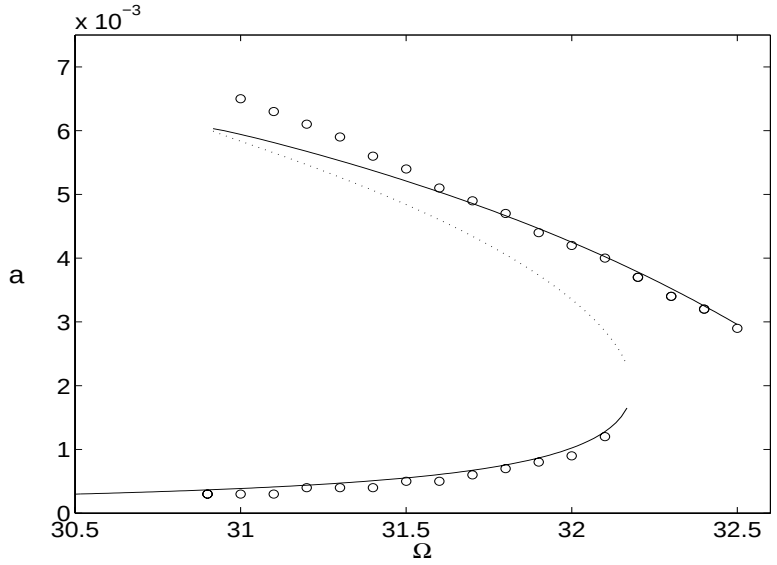

Figure 3.11: **Experimental and theoretical frequency-response curves, $a_{base} = 1g$. The theoretical curve uses parameters found by applying a least-squares algorithm that enforces known bifurcation ranges to the multiple-scales approximate solution.**

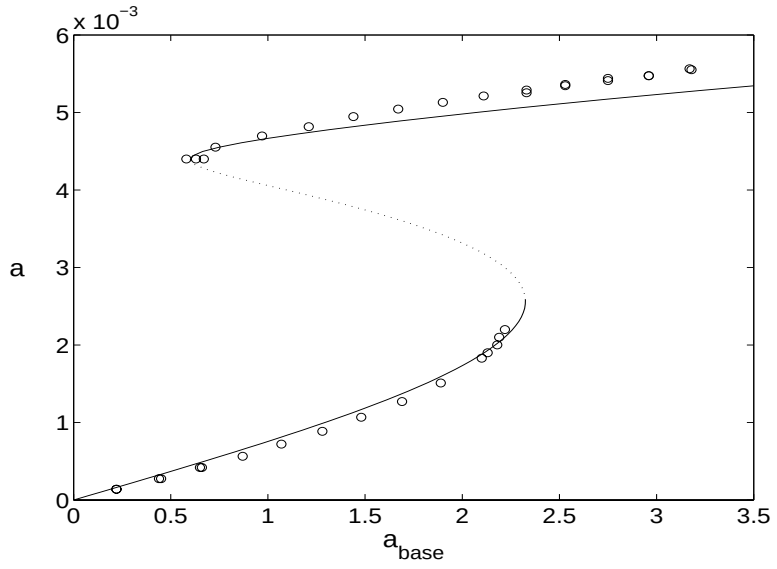

Figure 3.12: **Experimental and theoretical force-response curves, Ω=31.8 Hz. The theoretical curve uses parameters found by applying a least-squares algorithm that enforces known bifurcation ranges to the multiple-scales approximate solution.**

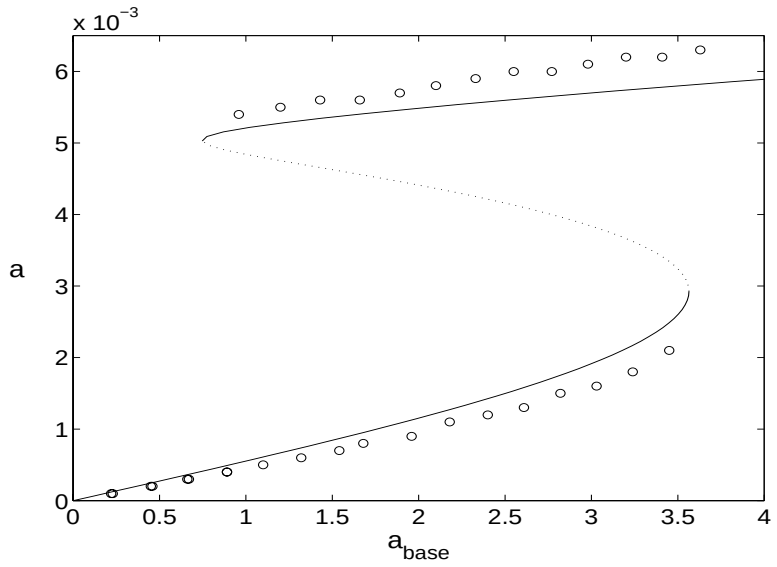

Figure 3.13: **Experimental and theoretical force-response curves, Ω=31.5 Hz. The theoretical curve is predicted using parameters extracted from a frequency sweep at $1g$ and a force sweep at 31.8 Hz.**

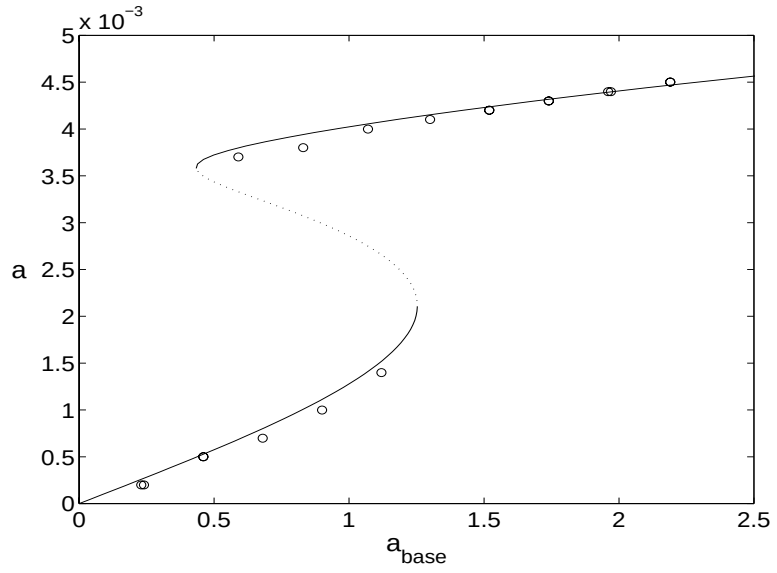

Figure 3.14: **Experimental and theoretical force-response curves, Ω=32.1 Hz. The theoretical curve is predicted using parameters extracted from a frequency sweep at $1g$ and a force sweep at 31.8 Hz.**

3.2.4 Development of a Faster Algorithm with Less Initial-Condition Sensitivity

While doing numerical simulations to validate the curve-fitting algorithm it became apparent that the initial-condition sensitivity and the degree of computational intensity were not workable. Finding a way to get back to using the **fmins** function in Matlab seemed extremely desirable. Much of the execution time for the previous algorithm was spent using a built-in Matlab function to find the roots of the sixth-order polynomial and then eliminate negative and complex roots and sort the positive ones in order of increasing magnitude. We sought a way to avoid as much of this process as possible. One way to do this is to go back to treating a as the independent variable. For the amplitude-response curve, using a as the independent variable means that finding the corresponding value of a_{base} requires only evaluating an expression that is one-to-one. For the frequency-response curve, solving for Ω for a given value of a means only solving a quadratic equation. This does force us to sort the experimental data according to which of the two solutions of the quadratic equation they correspond to. For example, if Ω is described by an equation of the form

$$a\Omega^4 + b\Omega^2 + c = 0 \tag{3.14}$$

then, below resonance

$$\Omega^2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \tag{3.15}$$

while, above resonance

$$\Omega^2 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \tag{3.16}$$

It should be noted that sorting the data in this way guarantees that, for the frequency-response case, only stable branches of the theoretical curve will be used in error calculations.

Ways to include the bifurcation range penalty and branch stability were still needed. For the bifurcation range, we want the penalty that the curve fit adds to the error to

be zero if the bifurcation occurs within the known range and then jump to a very large number if the bifurcation is outside the range. This is approximated by using a function that is zero if the bifurcation occurs in the middle of the range, remains small while the bifurcation is within the known range, and then grows rapidly as the theoretical location of the bifurcation leaves the experimentally known range. One function behaving in this way is

$$bifurcation\ penalty = \left(\frac{\alpha_{theoretical} - \alpha_{experimental}}{\alpha_{mean}} \right)^n \tag{3.17}$$

where α is the control parameter (either a_{base} or Ω) and n is an exponent used to tune the penalty function. The value of n was set to 10 in the final algorithm. A comparison of the ideal step-function penalty and the continuous polynomial suggested in Equation (3.17) is shown in Figure 3.15. In order to use Equation (3.17) to compute the bifurcation penalty, we need to be able to calculate $\alpha_{theoretical}$. Using expressions for the values of a_{base} and Ω at the bifurcations, we do two iterations of Newton's method to numerically approximate $\alpha_{theoretical}$. Having a numerical estimate for the theoretical value of the control parameter at the bifurcation allows the penalty to be calculated according to Equation (3.17).

Based on this same idea of using continuous functions to approximate step-function penalties, a method for enforcing the stability of branches of the curves was developed. If a is the independent variable, using only stable branches of the force-response curve for error calculation requires that the theoretical value of a at the bifurcation be less than the nearest experimental value. Ideally, we would like the penalty for this constraint to be zero if the theoretical value is less than the experimental value and very large if the theoretical value is greater than the experimental value; that is,

$$Stability\ Penalty = \begin{cases} 0, & \text{if } a_{theoretical} < a_{experimental}; \\ \infty, & \text{if } a_{theoretical} > a_{experimental}. \end{cases} \tag{3.18}$$

Numerically, this is accomplished by using the following function:

$$Stability\ Penalty = e^{\Lambda(a_{theoretical} - a_{experimental})} \tag{3.19}$$

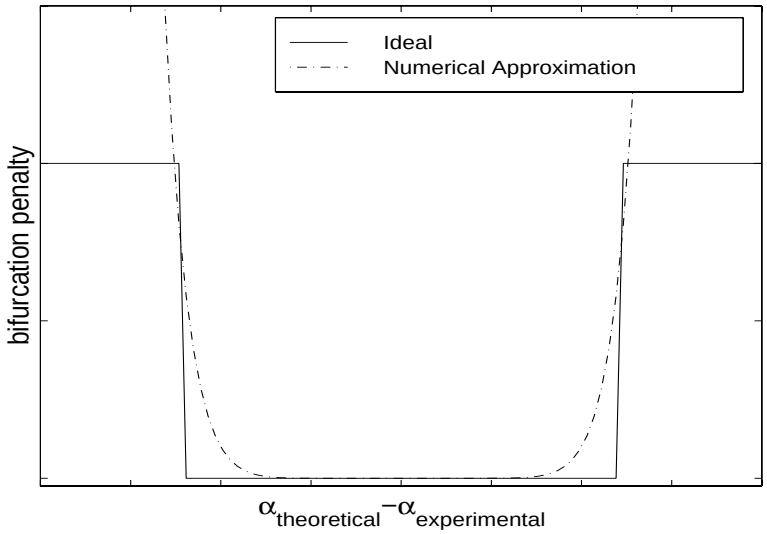

Figure 3.15: Ideal and approximate penalties imposed by the curve-fitting algorithm when the bifurcation does not occur within the theoretically known range.

where Λ is a large number.

A final modification to the algorithm was to return to a_{base} and Ω as the independent variables. From the perspective of the algorithm, making this change means switching the error used from the horizontal to the vertical distance between the experimental data point and the theoretical curve. These errors are labeled as h and v respectively in Figure 3.17. This change did not significantly impact the results of the curve fit. It does, however, allow the algorithm to use the same independent and dependent variables as the experiment. This change was implemented by guessing points near the branch of interest and using Newton's method to find the nearest root. We are essentially finding only one root of the frequency-response equation instead of the previous six. We avoid sorting the roots because only the root of interest is found (the stable point corresponding to the experimental data point). Branch stability for this method depends on guessing points near enough to the stable branch of interest so that the stable root will be found.

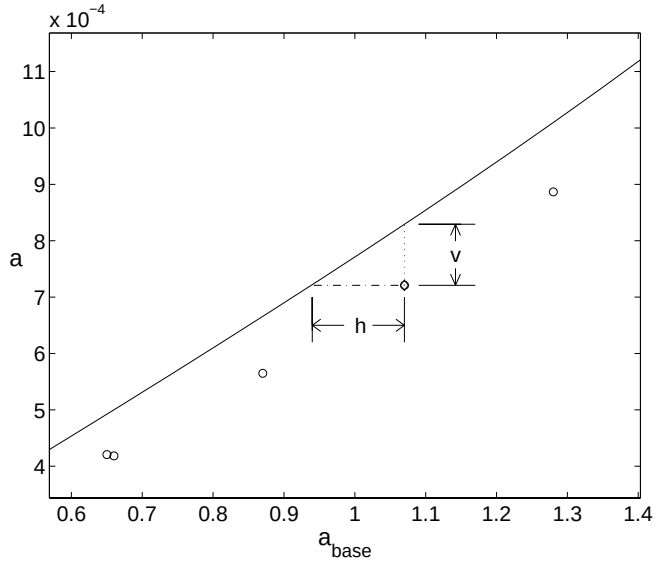

Figure 3.16: Experimental forcing-sweep data with the known bifurcation range shown. This physical constraint is included to improve the curve-fitting algorithm.

This algorithm based on using **fmins** is less sensitive to initial guesses and is much faster than the previous algorithm. These improvements in speed and initial-condition sensitivity have come without appreciable loss in accuracy. Results from each of the two curve-fitting techniques were compared to one another for both experimental data and data generated from numerical simulations.

Table 3.1 shows that, for the data generated by numerical simulation, the two algorithms give results that are both accurate and precise: they agree reasonably well with one another and give estimates of the parameters that are close to the known values used to produce the data. Table 3.2 demonstrates that the results for experimental data are precise as well. As can be seen, the estimates for α and δ are not in agreement with one another. However, there is good agreement in the estimate of the effective cubic nonlinearity coefficient $-\frac{1}{3}(2\delta\Omega^2 - 3\alpha)$. This indicates that, for the range of excitation frequencies used, the effects of α and δ cannot be separated because Ω does not vary significantly. Only the value of μ_1 can be checked for accuracy, since it is the only parameter whose value is

Table 3.1: Comparison of new (fmins) and old (computationally intensive) curve-fitting results for numerical data.

		New Algorithm		Old Algorithm	
Parameter	Known Value	Estimated Value	Percent Error	Estimated Value	Percent Error
η_e	100	99.993	0.007	99.999	0.001
μ_1	0.2	0.196	2.105	0.193	3.402
μ_2	300	302.52	0.841	303.50	1.165
δ	1.00e+04	9.83e+03	1.742	1.00e+04	0.365
α	9.00e+08	8.93e+08	0.778	9.01e+08	0.151
effective nonlinearity	-4.79e+08	-4.79e+08	0.070	-4.79e+08	0.037

experimentally known. The value found from linear vibration testing (half-power method) is 0.18, which is within 6% of the value found by the curve-fitting algorithm.

The inability of the new **fmins** curve-fitting routine to give back the exact values for the parameters used to generate the numerical data was disconcerting. The bifurcation penalty was turned off and the data was re-fitted. Table 3.3 shows that the curve-fitting routine without the penalty is able to fit the data exactly.

Making this same comparison between the new algorithm with and without the bifurcation ranges for experimental data shows significant differences. The difference in the estimate for δ is explained as before: the agreement in the estimates for the effective cubic nonlinearity coefficient is what is important. Turning off the bifurcation penalty for the old algorithm also leads to underestimation of μ_1 and μ_2. The value of μ_1 known from linear testing indicates that the estimate given by the algorithm with the bifurcation penalty enforced is the more accurate of the two. Similarly, underestimating μ_2 gives a frequency-response curve that is physically unrealizable. The curve is predicted to be stable well

Table 3.2: Comparison of new (fmins) and old (computationally intensive) curve-fitting results for experimental data.

	New Algorithm	Old Algorithm	
Parameter	Estimated Value	Estimated Value	Percent Comparison
η_e	81.9	80.4	1.85
μ_1	0.185	0.1747	5.87
μ_2	185	199	7.36
δ	1.04e+01	1.51e+04	200.
α	4.49e+08	1.05e+09	80.5
effective nonlinearity	-4.49e+08	-4.50e+08	0.213

Table 3.3: Comparison of the effect of enforcing the bifurcation penalty for the new (fmins) curve-fitting algorithm with numerical data.

		With Bifurcation Penalty		Without Bifurcation Penalty	
Parameter	Known Value	Estimated Value	Percent Error	Estimated Value	Percent Error
η_e	100	99.993	0.007	100.000	0
μ_1	0.2	0.196	2.105	0.200	0
μ_2	300	302.52	0.841	300.00	0
δ	1.00e+04	9.83e+03	1.742	1.00e+04	1.2e-07
α	9.00e+08	8.93e+08	0.778	9.00e+08	5.39e-08
effective nonlinearity	-4.79e+08	-4.79e+08	0.070	-4.79e+08	4.24e-09

Table 3.4: Comparison of the effect of enforcing the bifurcation penalty for the new (fmins) curve-fitting algorithm with experimental data.

	With Bifurcation Penalty	Without Bifurcation Penalty	
Parameter	Estimated Value	Estimated Value	Percent Comparison
η_e	81.9	78.6	4.06
μ_1	0.185	0.0752	84.5
μ_2	185	132	33.2
δ	1.04e+01	7.88e-01	272
α	4.49e+08	4.51e+08	0.395
effective nonlinearity	-4.49e+08	-4.51e+08	0.481

beyond the range where the bifurcation is known to occur. These two observations indicate that the curve-fitting algorithm with the bifurcation penalty included gives more accurate predictions of the damping coefficients than the same algorithm without that bifurcation penalty. This improvement in estimating the damping coefficients far out weighs any of the comparatively small errors introduced in estimates of the other parameters.