Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Sound and Vibration 281 (2005) 783–798

JOURNAL OF SOUND AND VIBRATION

www.elsevier.com/locate/jsvi

Dynamic response simulation for a nonlinear system

D.E. Roberts[a,*], N.C. Hay[b]

[a] *Centre for Mathematics and Statistics, Napier University, Edinburgh EH10 5DT, UK*
[b] *School of Engineering, Napier University, Edinburgh EH10 5DT, UK*

Received 8 September 2003; accepted 2 February 2004
Available online 2 October 2004

Abstract

Laboratory simulation testing has for many years contributed significantly to the durability and quality of motor vehicles. Most sophisticated test rigs use an iterative algorithm that generates the input drive files that reproduce service environments under laboratory conditions. Essentially the algorithm solves a nonlinear, multiple channel dynamic system. In this paper, the nonlinear problem is recast as a system of algebraic equations. This mathematical framework allows the application of alternative but well understood solution techniques. Using mathematical simulations, conclusions are drawn concerning the choice of iteration gain in the current algorithm and the better performance of alternative numerical solution procedures.
© 2004 Elsevier Ltd. All rights reserved.

1. Introduction

In laboratory simulation testing, a structure is mounted in a test rig and is excited in such a way that the service environment, as represented by a set of responses from transducers, is reproduced. It is believed that, when these responses are replicated, the complex stress field within the structure that occurs in service is also reproduced. The test rig and the test structure form a nonlinear dynamic system and the problem to be solved is to determine the input to this unknown multiple channel nonlinear system. The technology that achieves this was developed in the 1970s—see e.g. Ref. [1]—following the introduction of the hydraulic servo-valve, the construction of algorithms for quickly processing random data in terms of Fourier transforms, and of course the

*Corresponding author. Tel.: +44-131-455-2634; fax: +44-131-455-2651.
E-mail addresses: d.roberts@napier.ac.uk (D.E. Roberts), n.hay@napier.ac.uk (N.C. Hay).

0022-460X/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jsv.2004.02.016

development of more powerful computers. The technology is well summarised by Dodds and Plummer [2]. Generally, the procedure is that the system is treated as linear and measured using spectral analysis. An inverse system is then defined before an iterative algorithm determines the required drive files. Work to improve the performance of the iteration algorithm has been carried on over the years by, among others, Raath [3] who has developed a time-domain version of the algorithm as an alternative to the usual frequency domain implementation, and also by de Cuyper et al. [4] who examine improvements in the identification of the nonlinear system.

The work presented here reports on the realisation that the problem may be recast mathematically as a system of nonlinear algebraic equations. The conventional iteration algorithm is in fact an example of more general computational techniques for solving such systems. In the paper, these more general methods are introduced, and an application of them is then demonstrated in simulations using a single-degree-of freedom nonlinear mathematical model for the system, the Duffing equation. The new viewpoint involves both time- and frequency-domain considerations. Note that, for this paper, the single-degree-of-freedom system employed differs from the multiple channel physical laboratory simulation test system. Cost of equipment and control of parameters were considerations, but also using a single channel meant that the work could concentrate on the nonlinearity rather than interaction between channels. The latter will be studied at a later date.

Before introducing the new approach, the current algorithm is applied to the chosen simulation model, demonstrating the method and its characteristics in the face of various degrees of severity of nonlinearity. The situation is then studied mathematically and it is shown how discretisation leads to a system of nonlinear equations. After presenting some general methods for solving systems of nonlinear equations, the current algorithm is then shown to be a particular case. Finally, the feasibility of the more general approach is explored by comparing the success of the results of alternative solution methods.

2. Current algorithm

The current algorithm for achieving drive signals exists in several commercial software programs. For a description, the reader is referred to Ref. [2]. The procedure may be summarised as follows:

- Measurements of the response of the system are made during normal operation or specified operating conditions. These measurements are edited to provide a target response. In this paper, the target response is generated by exciting the system with band-limited white noise.
- The frequency response of the test rig and specimen is measured using spectral analysis.
- The validity of the frequency response measurement and the test rig design is then established using multiple and partial coherence functions, e.g. [5].
- An inverse frequency response function is computed and, from this, an initial drive file is derived using the target response.
- The drive file excites the system and produces a response, which is compared with the target response. The difference is then used to create a new drive file and the process continues as an iteration until an acceptable level of error is achieved.

The excitation data used for measuring the system consist of bandlimited white noise, represented by the components of a vector $\mathbf{x} := (x_0, x_1, ..., x_{N-1})$. The system response is sampled, yielding another vector $\mathbf{y} := (y_0, y_1, ..., y_{N-1})$, where, for signals of period T, $y_i := y(t_i)$ with $t_i := iT/N$ for $i = 0, 1, ..., N-1$. In the system measurement, spectral analysis uses the discrete Fourier transform of these signals, for which the kth components are denoted by X_k and Y_k, respectively, for $k = 0, 1, ..., N-1$, represented by the transform pairs

$$\mathbf{x} \leftrightarrow \mathbf{X}, \quad \mathbf{y} \leftrightarrow \mathbf{Y}. \tag{1}$$

The frequency response is based on the cross-spectral density estimate of the input and output signals as given by Bendat et al. [5, p. 138]

$$S_{yx}(\omega_k) := \lim_{T\to\infty} \frac{1}{T} \langle Y_k^* X_k \rangle, \tag{2}$$

where T is the period of duration of the signals and ω_k is the kth discrete frequency, and $\langle \cdots \rangle$ denotes an expectation value. The auto-power spectral estimates $S_{xx}(\omega_k), S_{yy}(\omega_k)$ are defined in a similar manner and the frequency response function is then given by

$$H_k := \frac{S_{yx}(\omega_k)}{S_{xx}(\omega_k)}. \tag{3}$$

In the simulations to be presented here, a target response signal $\mathbf{y}^D$ is determined by exciting the system using a sequence $\mathbf{x}^D$ of random numbers generated as bandlimited white noise. The iteration process is described more mathematically in Fig. 1. The fraction of the drive signal increment $\mathbf{p}^{(n)}$ which is fed back is stipulated by the iteration gain λ_n, a positive scalar quantity not greater than unity, which is chosen manually. In practice, the full drive signal is not normally used in determining the first drive file since the approximations in the estimate of the model may lead to the system being damaged. Similarly, the gain during the iteration is generally less than one to ensure convergence of the iteration and is again chosen manually.

3. Example of current iteration

The behaviour of the current algorithm is illustrated using a model of the Duffing equation constructed in MATLAB/Simulink. The system being simulated represents a mechanical single-degree-of-freedom, damped spring–mass system comprised of a mass m, a viscous damper with coefficient c, and a nonlinear spring. The stiffness of the spring increases with amplitude as described by a linear stiffness coefficient k, and a nonlinear factor kk'. Such systems are usually described in terms of natural frequency $(1/2\pi)\sqrt{k/m}$ Hz and damping ratio $c/(2\sqrt{km})$. The equation of the system being simulated is

$$m\,\frac{d^2y(t)}{dt^2} + c\,\frac{dy(t)}{dt} + ky(t)[1 + k'y(t)^2] = kx(t) \tag{4}$$

subject to the initial conditions $y(0) = \dot{y}(0) = 0$. The mass is taken to be 100 kg, the damping ratio $\zeta = 0.1$ and the natural frequency is normalised to unity. The right-hand side is chosen so that the input and output signals have similar magnitudes.

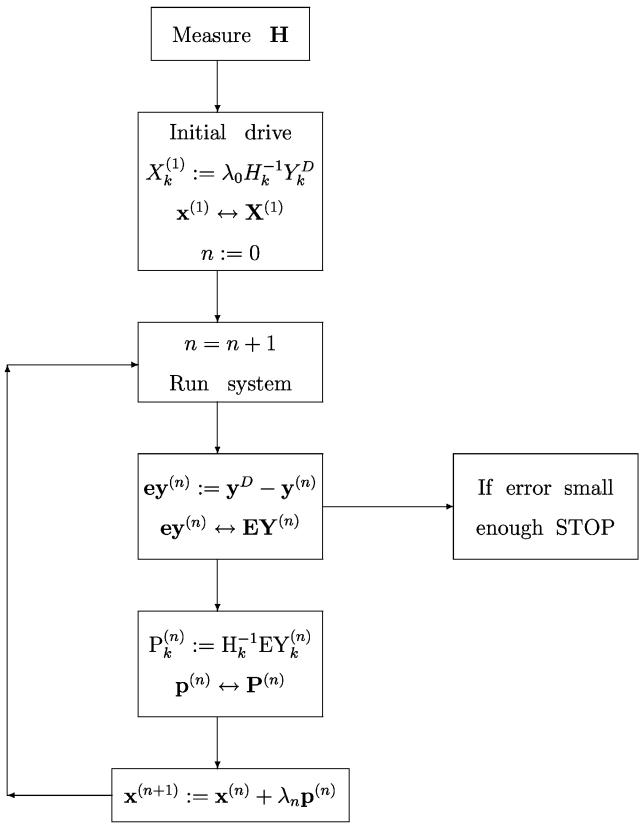
Measure $\mathbf{H}$
Initial drive
$X_k^{(1)} := \lambda_0 H_k^{-1} Y_k^D$
$\mathbf{x}^{(1)} \leftrightarrow \mathbf{X}^{(1)}$
$n := 0$
$n = n + 1$
Run system
$\mathbf{ey}^{(n)} := \mathbf{y}^D - \mathbf{y}^{(n)}$
$\mathbf{ey}^{(n)} \leftrightarrow \mathbf{EY}^{(n)}$
If error small enough STOP
$\mathrm{P}_k^{(n)} := \mathrm{H}_k^{-1}\mathrm{EY}_k^{(n)}$
$\mathbf{p}^{(n)} \leftrightarrow \mathbf{P}^{(n)}$
$\mathbf{x}^{(n+1)} := \mathbf{x}^{(n)} + \lambda_n \mathbf{p}^{(n)}$

Fig. 1. Current iteration algorithm—setting $\mathbf{x}^{(0)} := \mathbf{0}$.

When identifying the physical system, the normal practice is to use a large number of averages to improve the expectation value of Eq. (3) and achieve a smooth frequency response function. Here, a small number of averages are taken, but the function is smoothed using a least-squares fit. Numerical experiments suggest that the least-squares fitting is as good as employing a large number of averages. Fig. 2 illustrates the magnitude of the measured frequency response function—averaged over ten records—and a smoothed version obtained from a least-squares fit to produce a rational function which has as numerator a linear polynomial and as denominator a quadratic polynomial in frequency. In addition, the frequency response function corresponding to the linear part of Eq. (4) is also shown for comparison.

In these estimates, randomised drive signals with similar standard deviation to the desired drive input were used and the corresponding responses were determined. The drive signal, $\mathbf{x}^D$, is generated as a bandlimited random time series of $N = 1024$ points, over a frame length $T = 102.4$ s.

A sequence of experiments is conducted with the nonlinear coefficient, k', taking values from 0.15 to 0.45 in steps of 0.05. For a given value, the corresponding response $\mathbf{y}^D$ is computed by solving Eq. (4), using Simulink in MATLAB. Parts of these signals are shown in Fig. 3.

The iteration algorithm is applied with $\lambda_n = 0.5,\ n = 0, 1, ...$, to produce a sequence of response vectors $\mathbf{y}^{(n)},\ n = 0, 1, ...$ which converge to $\mathbf{y}^D$. The results are summarised in Fig. 9. The algorithm

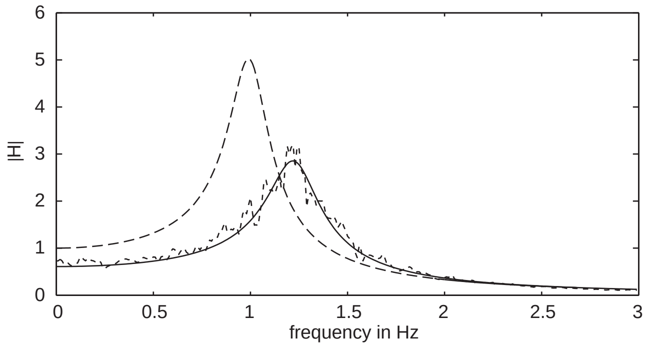

Fig. 2. Frequency response functions for $k' = 0.2$. The measured values are shown by the dotted line, while the least-squares fit to these is indicated by the solid line. For comparison, the function corresponding to the linear part of the system is given by the dashed line.

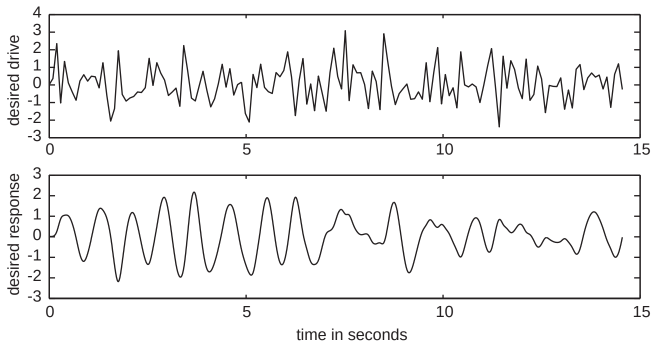

Fig. 3. Part of drive and response signals against time for $k' = 0.2$.

stops if the fractional Euclidean norm of the error vector

$$\mathbf{ey}^{(n)} := \mathbf{y}^D - \mathbf{y}^{(n)}, \tag{5}$$

i.e.

$$|\mathbf{ey}^{(n)}|/|\mathbf{y}^D| \tag{6}$$

falls below 5%. The error in the response achieved is shown in Fig. 4 as a function of time.

In practice, when the iterations fail to converge, the operator is free to adjust the iteration gain. For example, at the higher nonlinearity of $k' = 0.25$, the gain would be reduced and the iteration restarted, at the expense of slowing the convergence.

4. System of algebraic equations

In this section the problem is restated in terms of a system of algebraic equations. This opens up the possibility of applying well-known numerical techniques for solving such systems. In addition,

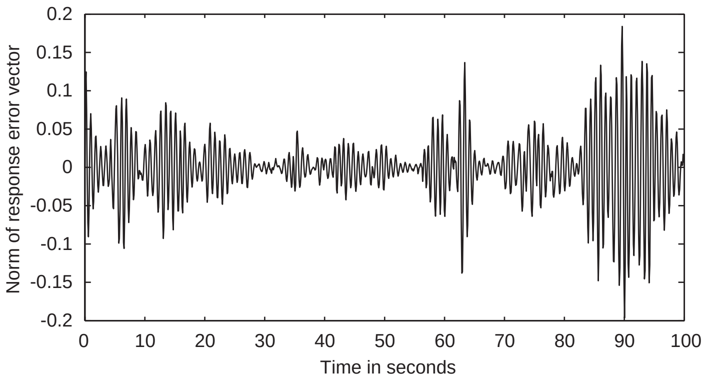

Fig. 4. Response error against time using the current algorithm, for $k' = 0.2$.

it is shown how the conventional approach appears as a particular case. One such computational strategy is applied to the simulation introduced in the previous section.

The point of view proposed in this paper is to note that the sampled response vector $\mathbf{y}$ is a function of the input signal $\mathbf{x}$ as symbolised in Fig. 5:

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) \tag{7}$$

or, in component form

$$\left.\begin{aligned} y_0 &= f_0(x_0, x_1, ..., x_{N-1}) \\ y_1 &= f_1(x_0, x_1, ..., x_{N-1}) \\ &\vdots \\ y_{N-1} &= f_{N-1}(x_0, x_1, ..., x_{N-1}) \end{aligned}\right\}. \tag{8}$$

To illustrate this, the model problem of the previous section is considered. Eq. (4) is discretised to produce a system of equations, thus providing explicit information about the vector-valued function $\mathbf{f}$ and the corresponding Jacobian.

First of all, consider the linear system obtained by setting $k' = 0$ in Eq. (4). The response is related to the input by a convolution in the time domain

$$\mathbf{y} = \mathbf{h} * \mathbf{x}, \tag{9}$$

where the discretised impulse response $\mathbf{h} := (h_0, h_1, ..., h_{N-1})$, is the inverse discrete Fourier transform of the frequency response function, $\mathbf{H} := (H_0, H_1, ..., H_{N-1})$, i.e.

$$\mathbf{h} \leftrightarrow \mathbf{H}. \tag{10}$$

This convolution may be written as a matrix product

$$\mathbf{y} = C_h \mathbf{x} \tag{11}$$

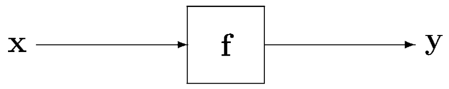

Fig. 5. Discretised system.

in which the $N \times N$ circulant matrix C_h has (i,j) component h_{i-j}

$$C_h = \begin{bmatrix} h_0 & h_{N-1} & h_{N-2} & \dots & h_1 \\ h_1 & h_0 & h_{N-1} & \dots & h_2 \\ \vdots & & & & \\ h_{N-1} & h_{N-2} & h_{N-3} & \dots & h_0 \end{bmatrix}. \tag{12}$$

The ith component of the vector Eq. (11) yields the approximate value of the response $y(t)$ at $t = t_i$. Eq. (11) may be rewritten

$$\mathbf{x} = [C_h]^{-1}\mathbf{y} \tag{13}$$

which may be regarded as a discretisation of Eq. (4) with $k' = 0$.

This process is extended to approximate the whole of the left-hand side of Eq. (4) at $t = t_i$ for non-zero k':

$$[m\ddot{y} + c\dot{y} + ky(1 + k'y^2)]_{t=t_i} \approx k[C\mathbf{y}]_i + kk'y_i^3 \tag{14}$$

for some appropriate circulant matrix C, such that the nth component of the DFT of the vector $[C\mathbf{y}]$ is given by $(1/k)(-\omega_n^2 m + \mathrm{j}\omega_n c + k)Y_n$.

The discretisation of Eq. (4), after division by k, may now be expressed as a vector equation:

$$\mathbf{x} = C\mathbf{y} + \mathbf{g}(\mathbf{y}) = \mathbf{f}^{-1}(\mathbf{y}), \tag{15}$$

where $[\mathbf{g}(\mathbf{y})]_i := k'y_i^3$, thus yielding an explicit form for the *function inverse* of $\mathbf{f}$ in Eq. (7).

The mathematical problem may be stated as follows: given a vector $\mathbf{y}^D = (y_0^D, y_1^D, \dots, y_{N-1}^D)$ and a particular function $\mathbf{f}$, determine a vector $\mathbf{x}$, such that

$$\mathbf{f}(\mathbf{x}) - \mathbf{y}^D = \mathbf{0}. \tag{16}$$

This is a system of nonlinear algebraic equations for which the solution is readily seen to be $\mathbf{f}^{-1}(\mathbf{y}^D)$. In practice, the explicit form of $\mathbf{f}$ is not known, but for a given vector $\mathbf{x}$, the value of $\mathbf{y} = \mathbf{f}(\mathbf{x})$ may be obtained by "running the system".

4.1. Iterative solutions

This type of problem is common, and there are well-known computational techniques for solving Eq. (16). For a survey of practical algorithms which may be used to solve systems of nonlinear algebraic equations, the reader is referred to a review by Martinez [6]. All the methods considered are iterative. Starting from some initial approximation $\mathbf{x}^{(0)}$, a sequence of iterates, $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots$, is generated which converge, ideally, to the desired solution.

In order to understand these techniques, a brief account of Newton's method for systems of nonlinear equations is given. This algorithm follows from the Taylor expansion in several variables of $\mathbf{f}(\mathbf{x})$ about the current approximation $\mathbf{x}^{(n)}$,

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{x}^{(n)}) + [J_{\mathbf{f}}(\mathbf{x}^{(n)})](\mathbf{x} - \mathbf{x}^{(n)}) + O(|\mathbf{x} - \mathbf{x}^{(n)}|^2), \tag{17}$$

where $J_{\mathbf{f}}(\mathbf{x})$ denotes the Jacobian matrix of order $N \times N$ for the vector-valued function $\mathbf{f}(\mathbf{x})$ in Eq. (7)

$$[J_{\mathbf{f}}(\mathbf{x})]_{i,j} := \frac{\partial y_i}{\partial x_j} \quad \text{for } i,j = 0, 1, \ldots, N-1 \tag{18}$$

the partial derivatives being evaluated at $\mathbf{x}$. In the context of matrix algebra, vectors are considered as column matrices.

The Jacobian for the model problem may be constructed from Eq. (15)

$$J_{\mathbf{f}^{-1}}(\mathbf{y}) := \left[\frac{\partial x_i}{\partial y_j}\right] = C + \mathbf{g}'(\mathbf{y}), \tag{19}$$

where

$$[\mathbf{g}'(\mathbf{y})]_i := 3k' y_i^2. \tag{20}$$

The dependence of the Jacobian on $\mathbf{y}$ and, therefore, on $\mathbf{x}$ is clear. We note that

$$[J_{\mathbf{f}}(\mathbf{x})]^{-1} = J_{\mathbf{f}^{-1}}(\mathbf{y}), \tag{21}$$

where $\mathbf{x}$ and $\mathbf{y}$ are related by Eq. (15).

For the linear system (9), it may be seen, from its definition, that the Jacobian is given by

$$J_{\mathbf{f}}(\mathbf{x}) = C_h, \tag{22}$$

i.e. a *constant* matrix.

For the general nonlinear system, if $\mathbf{x}^D$ is a solution to Eq. (16), then, setting $\mathbf{y} = \mathbf{y}^D$ in Eq. (17),

$$\mathbf{y}^D - \mathbf{y}^{(n)} = [J_{\mathbf{f}}(\mathbf{x}^{(n)})](\mathbf{x}^D - \mathbf{x}^{(n)}) + O(|\mathbf{x}^D - \mathbf{x}^{(n)}|^2), \tag{23}$$

where $\mathbf{y}^{(n)} = \mathbf{f}(\mathbf{x}^{(n)})$. Ignoring the error term in Eq. (23) leads to the following iteration scheme:

$$\mathbf{x}^{(n+1)} := \mathbf{x}^{(n)} + [J_{\mathbf{f}}(\mathbf{x}^{(n)})]^{-1}\mathbf{ey}^{(n)} \tag{24}$$

provided the Jacobian is non-singular at $\mathbf{x}^{(n)}$. This is Newton's method which is locally quadratically convergent—see e.g. Ref. [6]. It may be rewritten as

$$\mathbf{x}^{(n+1)} := \mathbf{x}^{(n)} + \mathbf{p}^{(n)}, \tag{25}$$

where

$$\mathbf{p}^{(n)} := [J_{\mathbf{f}}(\mathbf{x}^{(n)})]^{-1}\mathbf{ey}^{(n)}. \tag{26}$$

However, since $\mathbf{f}$ is not known explicitly, the Jacobian matrix cannot be constructed. Hence, "quasi-Newton" methods are considered which generalise Eq. (24) to

$$\mathbf{x}^{(n+1)} := \mathbf{x}^{(n)} + [B_n]^{-1}\mathbf{ey}^{(n)} \tag{27}$$

in which the matrix B_n plays the role of $J_{\mathbf{f}}(\mathbf{x}^{(n)})$. This may be recast as Eq. (25), where

$$\mathbf{p}^{(n)} := [B_n]^{-1}\mathbf{ey}^{(n)}. \tag{28}$$

The idea is that, starting from some initial estimate of the Jacobian, B_0, this is then updated using a simple formula. A very common approach is based on a version of the secant method and was suggested by Broyden [7], in which the updated inverse matrix $[B_{n+1}]^{-1}$ may be expressed in terms of $[B_n]^{-1}$, thus enabling a computationally efficient implementation of Eq. (27), provided we can readily compute B_0^{-1}:

$$[B_{n+1}]^{-1} := [B_n]^{-1} - (\delta\mathbf{x}^{(n)} + [B_n]^{-1}\delta\mathbf{ey}^{(n)})\frac{[\delta\mathbf{x}^{(n)}]^{\mathrm{T}}[B_n]^{-1}}{[\delta\mathbf{x}^{(n)}]^{\mathrm{T}}[B_n]^{-1}[\delta\mathbf{ey}^{(n)}]}, \tag{29}$$

where $\delta\mathbf{x}^{(n)} := \mathbf{x}^{(n+1)} - \mathbf{x}^{(n)}$, and $\delta\mathbf{ey}^{(n)} := \mathbf{ey}^{(n+1)} - \mathbf{ey}^{(n)}$.

4.2. *Alternative iteration schemes*

There are many variations of the basic iteration (27)—for other, similar, approaches see Martinez [6]. One possibility, which is relevant to our interests, is to keep B_n *constant* at some value B_0. The conventional approach discussed earlier—which treats the system as if it were linear—fits into this scheme

$$B_n := C_h, \quad n = 0, 1, 2, \ldots \tag{30}$$

in which the matrix C_h is based on the vector $\mathbf{h}$, the impulse response as in Eq. (12). This impulse response corresponds to the *measured* frequency response function.

Another possibility consists of using C_h as an *initial* approximation to the Jacobian in the nonlinear system:

$$B_0 := C_h \tag{31}$$

and then using Eq. (29) to produce the updates for Eq. (28). It may be noted here, that, for example, in the computation of $\mathbf{x}$ in Eq. (11), the fast Fourier transform may be employed, i.e. there is NO matrix multiplication performed. Indeed, all matrix multiplications are avoided by implementing the algorithm described in Ref. [8]. This algorithm is a memory-efficient approach which only requires scalar products of vectors to be computed.

However, these techniques are only locally convergent. That is, the *initial* approximations to the solution *and* the Jacobian must be *good enough* for convergence to follow. Even Newton's method may fail to converge for cases where there is a unique solution.

5. Improving global convergence

It was noted earlier, that, in the current algorithm, the iteration gain is reduced if the iterations diverge. In fact, a search can be conducted to determine a suitable iteration gain. In an attempt to achieve *global* convergence the basic iteration (25) is modified to allow a variable step in the search direction:

$$\mathbf{x}^{(n+1)} := \mathbf{x}^{(n)} + \lambda_n\mathbf{p}^{(n)}, \tag{32}$$

where λ_n, for $n = 0, 1, \ldots$ are real numbers lying between 0 and 1. The general iteration process is shown in Fig. 6. The values of λ_n may be constant, or, depending on the situation, the operator may vary them, e.g. some circumstances may warrant a moderate step reduction ($\lambda \approx 0.5$), while others may require larger reductions ($\lambda \ll 0.5$). The value of λ yielding the minimum error may be estimated using a backward line search. To do this a merit function is defined as follows:

$$\phi(\lambda) := \frac{||\mathbf{ey}(\mathbf{x} + \lambda\mathbf{p})||}{||\mathbf{y}^D||} \tag{33}$$

or, to avoid a square root, a common choice is

$$\psi(\lambda) := \tfrac{1}{2}[\phi(\lambda)]^2 = \frac{1}{2}\frac{[\mathbf{ey}]^{\mathrm{T}}[\mathbf{ey}]}{[\mathbf{y}^D]^{\mathrm{T}}[\mathbf{y}^D]}, \tag{34}$$

where $\mathbf{x}$ is the estimated drive at the last iteration, and $\mathbf{p}$ is the current search direction. The vector $\mathbf{ey}$ is the error in the response to the input $\mathbf{x} + \lambda\mathbf{p}$.

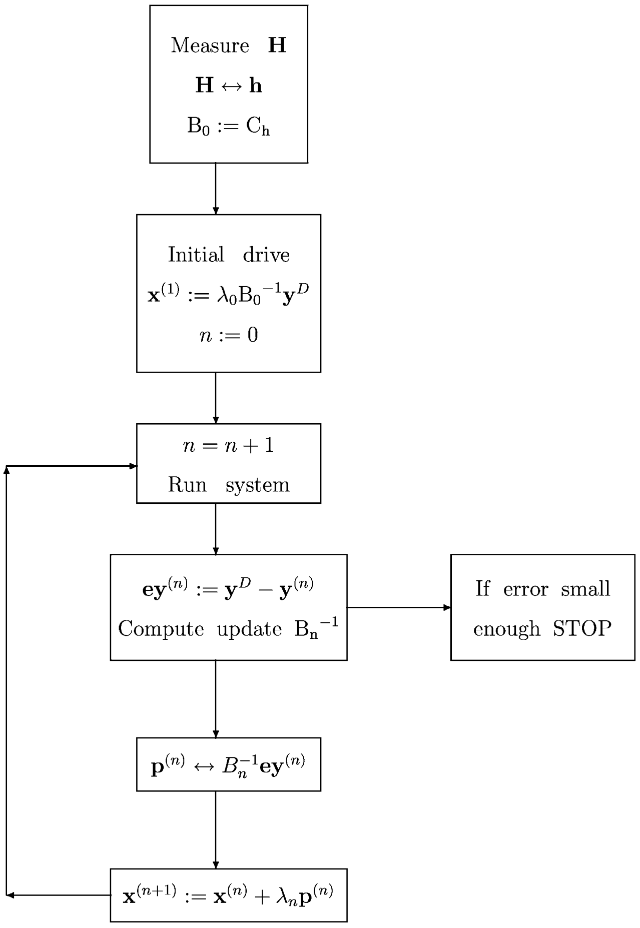
Measure $\mathbf{H}$
$\mathbf{H} \leftrightarrow \mathbf{h}$
$\mathrm{B}_0 := \mathrm{C}_\mathrm{h}$
Initial drive
$\mathbf{x}^{(1)} := \lambda_0 \mathrm{B}_0{}^{-1}\mathbf{y}^D$
$n := 0$
$n = n + 1$
Run system
$\mathbf{ey}^{(n)} := \mathbf{y}^D - \mathbf{y}^{(n)}$
Compute update $\mathrm{B_n}^{-1}$
If error small enough STOP
$\mathbf{p}^{(n)} \leftrightarrow B_n^{-1}\mathbf{ey}^{(n)}$
$\mathbf{x}^{(n+1)} := \mathbf{x}^{(n)} + \lambda_n \mathbf{p}^{(n)}$

Fig. 6. Scheme for alternative iteration algorithms—setting $\mathbf{x}^{(0)} := \mathbf{0}$.

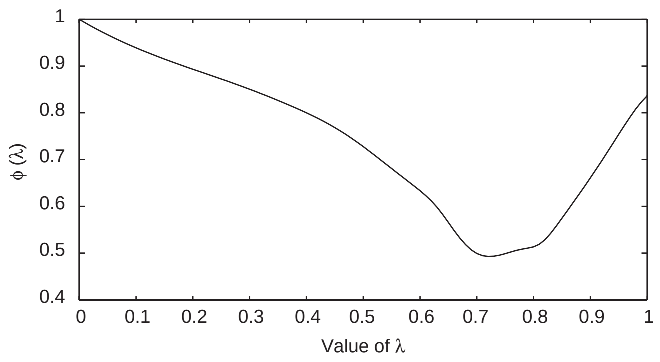

Fig. 7. Behaviour of ϕ as a function of λ for $k' = 0.3$ at the start.

The idea behind a backward line search is to model the merit function using a polynomial—typically a quadratic or a cubic. Quadratic interpolation is employed in this paper.

In the simulations presented later, the merit function of Eq. (33) is used to start off the process, i.e. to compute λ_0, and hence $\mathbf{x}^{(0)}$. A search is made for a value of λ that minimises the merit function. As an illustration of the behaviour of the error, ϕ is plotted as a function of λ for $k' = 0.30$ in Fig. 7.

The alternative merit function, Eq. (34) is employed during the iteration. Again, for illustration, Fig. 8 is a plot of ψ as a function of λ for the case of $k' = 0.3$ in the fourth iteration of Broyden's method.

For a full explanation of these and other search algorithms the reader is referred to Dennis et al. [9], Scales [10] and Numerical Recipes [11].

Whichever merit function is adopted, the price to be paid is that of "running the system" more often within a given iteration. The effect on the convergence behaviour will be demonstrated in Section 6.

6. Comparison of alternative iteration schemes

The alternative methods of iteration are now examined. The validity of the mathematical methods is established and their performance is compared. There are alternative choices for parameters and so, in the simulations presented here, each one uses the same Duffing model, the same desired solution and the same starting point. The same seven levels of nonlinearity are chosen for each alternative iteration method, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40 and 0.45. In each case, the iteration is stopped when the response is within 5% of the target response, or after a specified number of iterations.

For reasons of clarity of presentation, only four cases are plotted. However, all cases are represented in the tables.

The first method to be tested is the basic conventional algorithm as shown in Fig. 9.

The graphs figure shows that the conventional algorithm fails to converge at levels of nonlinearity 0.25 and greater. In industrial practice, the engineer would reduce the iteration gain

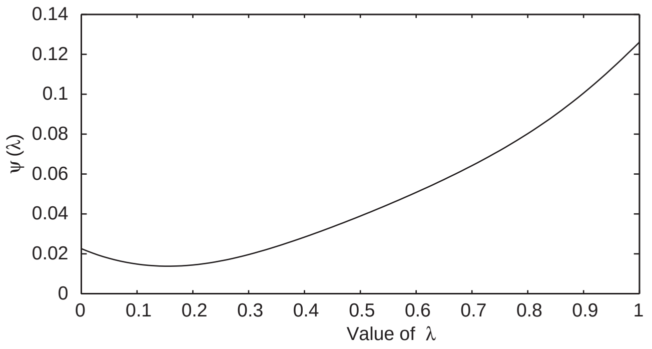

Fig. 8. Behaviour of ψ as a function of λ for $k' = 0.3$ on the fourth iteration using Broyden's method.

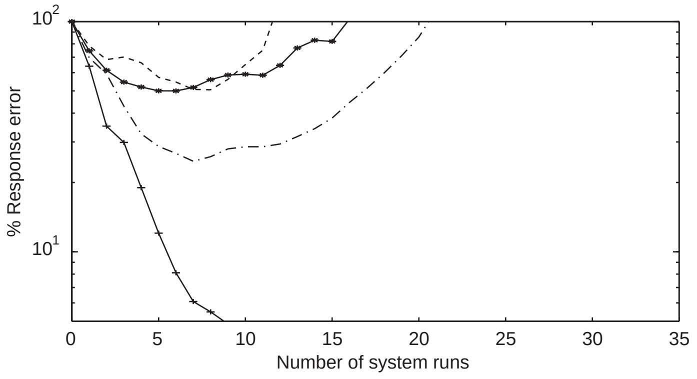

Fig. 9. Progress of conventional iteration for various levels of nonlinearity and an iteration gain of 0.5. For $k' = 0.15$, the iterations are shown by '+', for $k' = 0.25$ by a dash-dot line, for $k' = 0.35$ by '$*$', and for $k' = 0.45$ as a dashed line.

at the expense of running the system more often and also would examine the spectral densities in which troublesome frequencies may be detected.

Fig. 10 and Table 1 demonstrate the validity of manually reducing the iteration gain to 0.2. Convergence is achieved, for all bar the most nonlinear case, at the expense of a slower rate of convergence. The choice of iteration gain is generally left to the experience of the operators of industrial systems.

An early conclusion of considering the algorithm in the context of solving a system of algebraic equations, was that an appropriate iteration gain might be computed from the progress of the iteration, using a search for an appropriate iteration gain. This has been implemented and the results are presented in Fig. 11 and Table 2.

The divergent behaviour of Fig. 9 and 10 are eliminated, although manually choosing a small iteration gain initially performs better. However, even at a reduced gain, the convergence of the iteration stagnates for $k' = 0.25$ and 0.45, with a response error of about 11% after 35 iterations (about 100 system runs). The same behaviour is also observed for $k' = 0.30$ as indicated in Table 2.

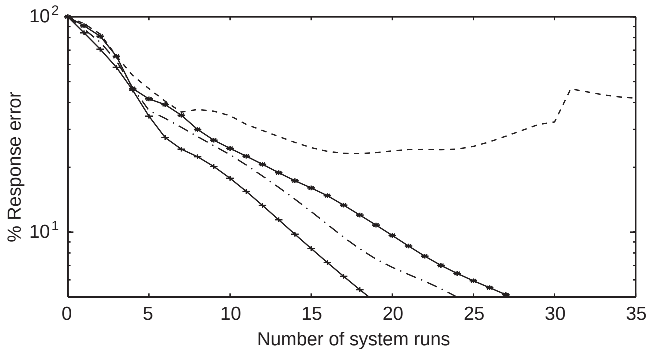

Fig. 10. Progress of conventional iteration for various levels of nonlinearity and an iteration gain of 0.2. For $k' = 0.15$, the iterations are shown by '+', for $k' = 0.25$ by a dash-dot line, for $k' = 0.35$ by '$*$', and for $k' = 0.45$ as a dashed line.

Table 1
Table of percentage error in the response for various levels of nonlinearity, for 35 runs, with $\lambda = 0.2$

Nonlinear coefficient k'	Percentage error	
	Constant Jacobian	Broyden's update
0.15	0.8	0.03
0.20	0.8	0.04
0.25	1.2	0.15
0.30	1.8	0.37
0.35	2.0	0.83
0.40	2.8	3.6
0.45	—	7.4

The above results used the conventional algorithm, and the conventional algorithm with search. These use a constant approximation to the Jacobian. The effect of updating the approximation, using Broyden's method, is now considered. Fig. 12 illustrates the results of this approach without a search.

The method works well for low levels of nonlinearity where it produces faster convergence. There is also convergence for levels of nonlinearity that failed to converge using conventional iteration. At higher levels of nonlinearity, the method still fails to converge.

The progress of Broyden's method improves when the iteration gain is reduced, Fig. 13, but has no great advantage over the conventional method as measured by the number of runs required to achieve a tolerance of 5%. However, it was noted that, as the number of runs were increased the error dropped faster for the updated technique—as indicated in Table 1.

The last of the sets of simulations presents, in Fig. 14, Broyden's method with a search.

The method is successful in achieving convergence at all the levels of nonlinearity that were considered but at the expense of running the system more often.

Table 2 compares the results for the conventional algorithm (constant Jacobian approximation) and Broyden's method (updated Jacobian approximation) using backward line searches for all the cases of nonlinearity. It indicates that the Broyden update has an advantage over the use of a

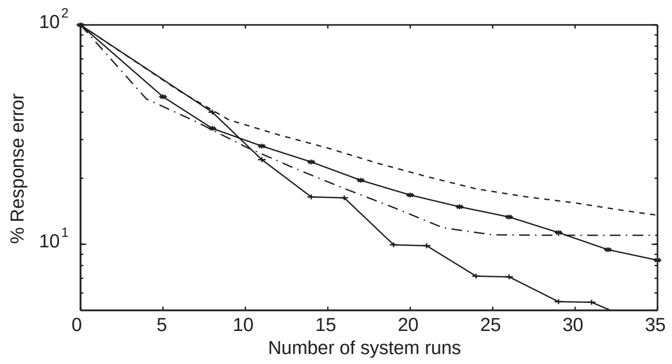

Fig. 11. Progress of iteration for various levels of nonlinearity—conventional algorithm with search. For $k' = 0.15$ the iterations are shown by '+', for $k' = 0.25$ by a dash-dot line, for $k' = 0.35$ by '$*$', and for $k' = 0.45$ as a dashed line.

Table 2
Table of number of system runs to achieve an error of 5%, for various levels of nonlinearity, using a search

Nonlinear coefficient k'	Number of system runs (iterations)	
	Constant Jacobian	Broyden's update
0.15	34 (10)	26 (6)
0.20	47 (15)	33 (9)
0.25	— (35)	31 (9)
0.30	— (35)	38 (11)
0.35	52 (16)	38 (11)
0.40	64 (20)	44 (13)
0.45	— (35)	57 (17)

The number of iterations for each method is in brackets. —, indicates that convergence was not achieved after 35 iterations.

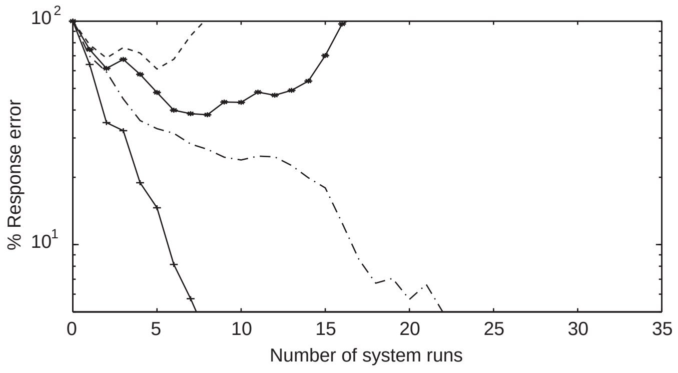

Fig. 12. Progress of iteration for various levels of nonlinearity—Broyden's method with an iteration gain of 0.5. For $k' = 0.15$ the iterations are shown by '+', for $k' = 0.25$ by a dash-dot line, for $k' = 0.35$ by '$*$', and for $k' = 0.45$ as a dashed line.

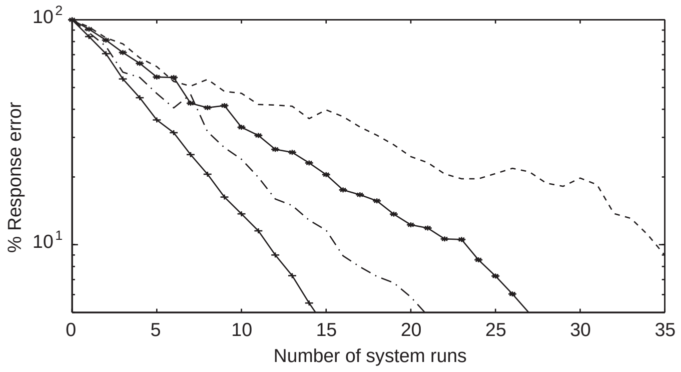

Fig. 13. Progress of iteration for various levels of nonlinearity—Broyden's method with an iteration gain of 0.2. For $k' = 0.15$ the iterations are shown by '+', for $k' = 0.25$ by a dash-dot line, for $k' = 0.35$ by '$*$', and for $k' = 0.45$ as a dashed line.

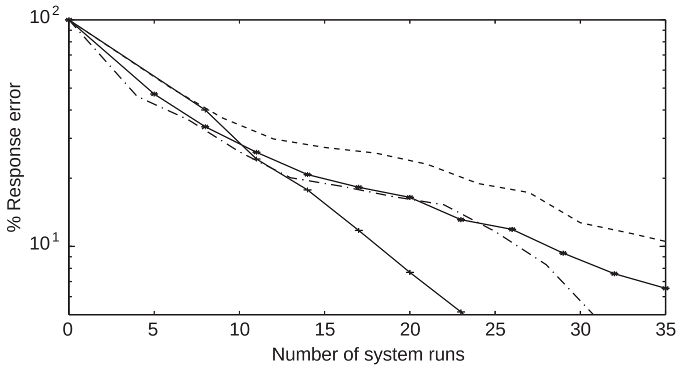

Fig. 14. Progress of iteration for various levels of nonlinearity—Broyden's method with search. For $k' = 0.15$ the iterations are shown by '+', for $k' = 0.25$ by a dash-dot line, for $k' = 0.35$ by '$*$', and for $k' = 0.45$ as a dashed line.

constant approximation to the Jacobian, by showing a faster convergence, and also by achieving convergence when the conventional algorithm fails.

The performance of search routines depends on chosen parameters and this requires further study. For example, the work presented does not use restarts, nor does it consider the effect of the many different forms of line search. In addition, there are many other types of update—including updating the frequency response function itself—for others see, e.g. Ref. [6]. Other approaches take advantage of the particular structure of the Jacobian. The aim of this work is to indicate that the particular point of view presented can be advantageous, and that recourse can be made to an arsenal of tried and tested techniques.

7. Conclusions

A new mathematical framework for the derivation of drive files for laboratory simulation test systems is demonstrated. The conventional algorithm is shown to be part of a broad mathematical area for which established mathematical techniques are available. This approach can achieve convergence in systems that do not readily converge with the conventional algorithm. It has also been shown that there is potential for improving the convergence of the latter using a backward line search.

Thus, this paper reports on a beginning, not a completion, of an investigation. The authors regard the work as the opening up of an area for further research. Consideration will be given to various solution techniques and the sensitivity of these to measurement errors. Systems with multiple channels, physical systems and alternative models of nonlinear behaviour will also be investigated.

References

[1] B.W. Cryer, P.E. Nawrocki, R.A. Lund, A road simulation system for heavy duty vehicles, Society of Automotive Engineers, SAE 760361, 1976.

[2] C.J. Dodds, A.R. Plummer, Laboratory road simulation for full vehicle testing—a review, *Symposium of International Automotive Technology*, Pune, India, SAE 2001-01-0047, January 2001.

[3] A.D. Raath, C.C. Van Waveren, Time domain approach to load reconstruction for durability testing, *Engineering Failure Analysis* 5 (1998) 113–119.

[4] J. De Cuyper, D. Coppens, C. Liefooghe, J. Debille, Advanced system identification methods for improved service load simulation on multi axial test rigs, *European Journal of Mechanical & Environmental Engineering* M 44 (1999) 27–39.

[5] J.S. Bendat, A.G. Piersol, *Random Data Analysis and Measurement Procedures*, third ed., Wiley Interscience, New York, 2000.

[6] J.M. Martinez, Practical quasi-Newton methods for solving nonlinear systems, *Journal of Computational and Applied Mathematics* 124 (2000) 97–121.

[7] C.G. Broyden, A class of methods for solving nonlinear simultaneous equations, *Mathematics of Computation* 19 (1965) 577–593.

[8] C.T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, SIAM, Philadelphia, PA, 1995.

[9] J.E. Dennis Jr., R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Non-linear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[10] L.E. Scales, *Introduction to Non-linear Optimization*, MacMillan, New York, 1985.

[11] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Numerical Recipes*, Cambridge University Press, Cambridge, 1992.