Multiple Force Identification for Complex Structures

by Robert Adams and James F. Doyle

ABSTRACT—This paper presents a method for determining force histories using experimentally measured responses. It is based on a recursive reformulation of the governing equations in conjunction with a general finite element program, this latter aspect making it applicable to complex structures. It can determine multiple isolated (uncorrelated) force histories as well as distributed pressures and tractions and allows for the data collected to be of dissimilar type. As a demonstration of the method and of its scalability, force reconstructions for an impacted shell and an impacted plate are determined using accelerometer and strain gage data.

KEY WORDS—Finite element method, force identification, regularization, structures

Introduction

The response $u(x, t)$ of a general linear structural system to a single excitation load $P(t)$ can be written as

$$u(x,t) = \int_0^t G(x, t-\tau)P(\tau)\,d\tau$$

where $G(x, t)$ is the system response function. The force identification problem is usually posed as: given some measurements $u(t)$ (perhaps imperfectly), and knowledge of the system function $G(x, t)$ (perhaps imperfectly), determine the load $P(t)$. This problem is difficult for two quite separate reasons. The first is that it is a highly ill-conditioned problem which means that small errors in the measurements or the modeling can cause very large variances in the identified forces. Second, the need to know the system function in analytical form has meant that most structural systems analyzed have been relatively simple.

Reference 1 gives an excellent summary of the literature on force identification as well as an overview of the subject itself; more recent citations can be found in Ref. 2. A variety of methods have been used, but the most common solution scheme is some form of deconvolution. Reference 3 summarizes some of the main deconvolution methods, although these are not directly applied to the force identification problem. References 2, 4, 5, and 6 used Fourier methods because of the relative simplicity of the inversion and because it is suitably matched to the spectral element approach. Deriving analytical response functions is a formidable challenge; Ref. 7 and Refs. 8 and 9 looked at half-space and half-plane problems, respectively; general frame structures were solved by the introduction of spectral elements in Ref. 2; folded plate structures were solved using plate spectral elements.[10] Each of these are quite limited, in that they are restricted to relatively simple geometries and did not address the question of multiple or distributed forces.

The beginnings of a deconvolution method coupled with a general finite element approach was given in Ref. 11 which used an FEM program to generate wavelet solutions which could be synthesized in the inverse solution. A quite different approach was pioneered in Refs. 12–14 for heat conduction problems and Refs. 15 and 16 for force identification problem. This approach uses ideas from dynamic programming, but what sets it apart from the above references is that it deals with the discretized form of the governing equations. This makes it suitable for coupling with a finite element modeling of the structure. However, as pointed out in Ref. 15, "one of the disadvantages of the method is that the amount of computations increases dramatically as the order of the model increases." Indeed, this necessitated the authors to introduce a modal reduction scheme such that the final size of the problem studied in Refs. 15 and 16 was nine degrees of freedom. More details and applications can be found in Ref. 17.

It is the goal of this paper to develop a method for force identification that is robust enough to be applied to complex structures, but whose computational costs scale as for a forward finite element analysis. Furthermore, we wish to be able to determine multiple isolated forces as well as distributed pressures and tractions. After developing the main ingredients of the method, we illustrate its attributes using experimental data from the impact of a 3-D shell and for the impact of a plate with a hole.

The Essential Difficulty

Before developing the specifics of the actual method, we first sketch the outlines of a naive method of force identification. This will also serve to introduce some of the main notations.

Using the discretization afforded by the finite element method,[18,19] the governing dynamic equations for a general complex (but linear) structure can be discretized in space as

$$[M]\{\ddot{u}\} + [C]\{\dot{u}\} + [K]\{u\} = \{P\} = [B_g]\{g\}. \quad (1)$$

In this, $\{u\}$ is the vector of all the free degrees of freedom and is of size $\{m_u \times 1\}$, $[K]$ is the $[m_u \times m_u]$ stiffness matrix, $[C]$ is the $[m_u \times m_u]$ damping matrix, $\{P\}$ is the $\{m_u \times 1\}$ vector of all applied loads (some of which are zero), $\{g\}$ is the $\{m_g \times 1\}$ subset of $\{P\}$ vector of non-zero applied loads, and $[B_g]$ is the $[m_u \times m_g]$ matrix that associates these loads

Robert Adams is a Professor and James F. Doyle is a Graduate Student, School of Aeronautics & Astronautics, Purdue University, West Lafayette, Indiana 47907.

Original manuscript submitted: January 7, 2000.
Final manuscript received: October 15, 2001.

with the degrees of freedom. Let these governing equations be discretized in time as the recurrence relations

$$\{u\}_{n+1} = [A]\{u\}_n + [B]\{g\}_n. \tag{2}$$

In this, n is the subscript over the discretized time; $\{u\}$ now contains the state vectors (at least displacement and velocity) and is of size $m \geq 2m_u$; the vector of forcing terms, $\{g\}$, is of size $\{m_g \times 1\}$; and $[B]$ is the $[m \times m_g]$ matrix that associates the forces with the DoF. Some specific forms for $[A]$ and $[B]$ are given in Appendix I.

In the forward problem, given the initial conditions as $\{u\}_1 =$ known, and the applied load histories $\{g\}$, we can solve for the response recursively from Eq (2). In the inverse problem of interest here, the applied loads are unknown but we know some information about the responses; we wish to use this information to determine the applied loads. In particular, assume we have a vector of measurements $\{d\}$ of size $\{m_d \times 1\}$ which are related to the structural DoF according to

$$\{d\}_n \Leftrightarrow [Q]\{u\}_n.$$

Note that the $[m_d \times m_u]$ matrix $[Q]$ could be a difference relation as would be the case for strains. We want to find the forces $\{g\}$ that make the system best match the measurements. Consider the general least squares error given by

$$E(u, g) = \sum_{n=1}^{N} \left[\{d - Qu\}_n^T \lceil W \rfloor \{d - Qu\}_n \right],$$

$$\{u\}_{n+1} = [A]\{u\}_n + [B]\{g\}_n,$$

where $\lceil W \rfloor$ could be a general weighting array on the data although we will take it as diagonal. Our objective is to find the set of forces $\{g\}$ that minimize this error functional.

It is possible to establish a global system of simultaneous equations by the usual procedures for minimizing the least squares. That is, we arrange Eq (2) to form

$$[\bar{A}]\{\bar{u}\} = [\bar{B}]\{\bar{g}\}, \quad \{\bar{u}\} = \{\{u\}_1, \cdots, \{u\}_N\}^T,$$

$$\{\bar{g}\} = \{\{g\}_1, \cdots, \{g\}_N\}^T.$$

Solve for $\{\bar{u}\}$ in terms of $\{\bar{g}\}$. That is, first decompose $[\bar{A}]$ and then solve $[G]$ as the series of back-substitutions

$$[\bar{A}][G] = [L^T DU][G] = [\bar{B}].$$

The matrix $[G]$ is the collection of forward solutions for unit loads applied for each of the unknown forces. We therefore can write the actual forward solution as

$$\{\bar{u}\} = [G]\{\bar{g}\}.$$

Substitute this into the error equation and minimize with respect to $\{g\}$. The resulting system is

$$[G^T Q^T W Q G]\{\bar{g}\} = [QG]^T [W]\{\bar{d}\}.$$

This naive scheme is not practical for two very important reasons.

Consider a dynamic problem where there are m_u degrees of freedom at N time steps; this would lead to the very large system array $[\bar{A}]$ of size $[(m_u \times N) \times (m_u \times N)]$. A system with 10,000 DoF over 2000 time steps has nearly 100×10^6 unknowns and a system size of $[\bar{A}]$ which is the square of that. This would lead to a RAM memory requirement of 3×10^9 MB. Furthermore, solving the system of equations would cost approximately N^3 operations, which on a 1 GFlop supercomputer would take 10^{13} seconds or 300000 years! Clearly, this approach to solving the problem does not scale very well and the procedure is restricted to very small problems only. Even if advantage is taken of the special nature of $[\bar{A}]$ as given in Appendix I, the numbers would still be outrageously large.

There is another reason why the above scheme will not work even if the computational cost could be afforded: the established system of equations is notoriously ill-conditioned. That is, any small perturbations in the input data or system parameters will cause significant changes in the estimated parameters.

The two key factors that must be addressed are therefore size and ill-conditioning. With this in mind, some of the attributes the method should have are:

- Be able to analyze structures with complexities as usually found in forward problems and handled by the finite element method.
- Be able to determine many unknown force histories forming unknown shape distributions as well as acting independent of each other.
- Be able to utilize many sensors, of different types, distributed through-out the structure and not placed in any particularly optimized positions.
- From a computational cost aspect, be scalable similar to forward problems using the finite element method.

In the following, the ill-conditioning is handled by the introduction of regularization terms.[3] We cope with the problem of size by using an algorithm adapted from Bellman's ideas in dynamic programming;[20] this scheme essentially allows a recursive (in time) solution of the problem.

Recursive Formulation of the Problem

Consider the general least squares error given by

$$E(u, g) = \sum_{n=1}^{N} \Big[\{d - Qu\}_n^T \lceil W \rfloor \{d - Qu\}_n + \lambda \{g\}_n^T [H]\{g\}_n \Big] \tag{3}$$

$$\{u\}_{n+1} = [A]\{u\}_n + [B]\{g\}_n \tag{4}$$

where the summation is over all the time steps as discussed before and the second term on the right of Eq (3) is the regularization term, as discussed in Appendix II. The main idea to be developed is to perform the minimization recursively rather than globally, working from the last time step. The original form of this algorithm was given in Refs. 20 and 21 and has much in common with the Kalman filter.[22–24]

Ricatti Equation

We will not give the detailed derivation but just hint at its essentials; a full derivation is given in Ref. 17.

Consider the arbitrary time step n; the optimized partial error sum is

$$E_n^*(u, g) \equiv \underset{g_n}{min} E_n(u, g)$$

$$= \sum_{j=n}^{N} \Big[\{d - Qu\}_j^T \lceil W \rfloor \{d - Qu\}_j + \lambda \{g\}_j^T [H] \{g\}_j \Big]$$

where E_n means the summation begins at $j = n$. The key to the algorithm is to rewrite the partial sum in terms of the solution to a previously optimized problem. Thus, at the earlier time

$$E_{n-1}^*(u, g) \equiv \underset{g_{n-1}}{min} \Big[\{Qu - d\}_{n-1}^T \lceil W \rfloor \{Qu - d\}_{n-1} + \lambda \{g\}_{n-1}^T [H] \{g\}_{n-1} \quad E_n^*(u_n = Au_{n-1} + Bg_{n-1}) \Big]$$

where the parenthesis of E_n^* means "function of". It can be shown that E_n^* is quadratic in $\{u\}_n$ for any n and therefore can be written as the recursion relation

$$E_n^*(u) = \{u\}_n^T [R_n] \{u\}_n + \{u\}_n^T \{S_n\} + C_n$$

where $[R_n]$, $\{S_n\}$ and C_n are coefficients. It is for these coefficients that a recurrence relation is established. Defining the inverse term

$$[D_n] \equiv \Big[2\lambda [H] + 2[B]^T [R_n] [B] \Big]^{-1}$$

then (after some manipulation) we get the recurrence relations

$$[R_{n-1}] = [Q]^T \lceil W \rfloor [Q] + [A]^T [R_n - 2R_n B D_n B^T R_n] [A]$$

$$\{S_{n-1}\} = -2[Q]^T \lceil W \rfloor \{d\}_{n-1} + [A]^T [I - 2R_n B D_n B^T] \{S_n\}. \quad (5)$$

This is a form of the Ricatti equation.[25] There is no need to compute the coefficient C_n since it disappears on differentiation. The starter values for the recursion are obtained by looking at the optimized error function at the end point $n = N$. This gives

$$[R_N] = [Q]^T \lceil W \rfloor [Q], \quad \{S_N\} = -2[Q]^T \lceil W \rfloor \{d\}_N. \quad (6)$$

In this way, starting with the end values, we recursively determine and store (either internally or to disk) the quantities $[D_n][2R_n B]^T [A]$ and $[D_n][B]^T \{S_n\}$. This is called the backward sweep. During the forward sweep, we compute

$$\{g^*\}_n = -[D_{n+1}][2RB]_{n+1}^T [A] \{u\}_n - [D_{n+1}][B]^T \{S_{n+1}\}$$

$$\{u\}_{n+1} = [A]\{u\}_n + [B]\{g\}_n. \quad (7)$$

As discussed in Appendix I, the manipulations with the large array $[A]$ are done in an efficient manner.

These are the form of the equations used in Refs. 15 and 16 for force identification and Refs. 12–14 for heat conduction problems. As discussed next, these equations are still unsuited for scaling to large sized finite element problems.

The arrays introduced with this form of the inverse problem are of size

$\{d\}$	=	$[m_d \times 1]$	
$\lceil W \rfloor$	=	$\{m_d \times 1\}$	diagonal
$[Q]$	=	$[m_d \times m]$	
$[R_n]$	=	$[m \times m]$	symmetric
$\{S_n\}$	=	$\{m \times 1\}$	
$[D_n^{-1}]$	=	$[m_g \times m_g]$	symmetric
$[D_n]$	=	$[m_g \times m_g]$	symmetric.

These arrays show a huge improvement in storage requirements as compared to the naive method. However, they are still not in a good form for scaling to large size problems. In particular, the square, fully populated array $[R_n]$ is of size $[m \times m]$ and recall that $m > 2m_u$. For our target system of 10000 DoF this is on the order of 1000 Mbytes (when stored in double precision). Hence, any manipulations involving it will be computationally intensive as well as requiring large storage. Furthermore, the system requirements are then orders of magnitude greater than needed for the corresponding forward analysis via FEM.

There is another reason for avoiding the use of the Ricatti form of the equations for large systems: as pointed out in Ref. 26, they are prone to a numerical instability unless special precautions are taken to preserve the positive definiteness of $[R_n]$. For small systems, enforcing the symmetry of $[R_n]$ by setting $[R_n] = \frac{1}{2}[R_n + R_n^T]$ is sufficient, but it is not clear that this is sufficient for large systems.

The equations will now be modified with the intention of avoiding having to form the matrix $[R_n]$ altogether and thus avoid any drawbacks associated with it.

Time Invariant Systems

For most structural problems of interest, the system is time invariant (the matrix $[A]$ is constant). The essence of the approach to take advantage of this is to realize that when the system does not change many of the intermediate matrices do not change between time steps and that the recurrence relations can be written in terms of their differences. This was first introduced in the series of papers by Kailath[26,27] who discusses a number of variations of the basic algorithm; derivations can also be found in Refs. 17 and 28.

Consider the symmetric array $[R_n]$ at two times in the recursion given by

$$[R_{n-1}] = [Q^T W Q] + [A^T][R_n][I - 2BD_n B^T][R_n][A]$$

$$[R_{n-2}] = [Q^T W Q] + [A^T][R_{n-1}][I - 2BD_{n-1} B^T][R_{n-1}][A].$$

There is a good deal of duplication in going from one level to the next; the only recursive term needed at the second level is $[D_n]$ which is also obtained from $[R_n]$ as the inverse of

$$[D_n^{-1}] = 2\lambda [H] + 2[B^T R_n B].$$

This motivates expressing the quantities in terms of differences

$$[\Delta R_n] \equiv [R_n] - [R_{n-1}].$$

The recursion relations are written for the differences.

The full sequence of recursion relations is

$$[D_{n-1}^{-1}] = [D_n^{-1}] - 2[B^T Y_n][L_n][Y_n^T B]$$
$$[D_{n-1}] = [D_{n-1}^{-1}]^{-1}$$
$$[L_{n-1}] = [L_n] + 2[L_n Y_n^T B][D_{n-1}][B^T Y_n L_n]$$
$$[Y_{n-1}] = [A^T]\left[Y_n - 2K_n B^T Y_n\right]$$
$$[K_{n-1}] = [K_n] - \left[Y_n - 2K_n B^T Y_n\right][L_n Y_n^T B][D_{n-1}]$$
$$\{S_{n-1}\} = -2[Q^T W]\{d\}_{n-1} + [A^T]\left\{\{S_n\} - 2[K_n B^T]\{S_n\}\right\}.$$

It is clear that these computations can be sequenced so that many of the partial products can be re-used. When doing this, the carry-over arrays should be chosen to be the small ones. One such possible scheme is given in Ref. 28. The initial conditions for the recursive variables are

$$[D_n^{-1}] = 2\lambda[H] + 2[B^T Q^T]\lceil W \rfloor [QB]$$
$$[D_N] = [D_N^{-1}]^{-1}$$
$$[Y_N] = [A^T][Q^T]$$
$$[K_N] = [Q^T W][QB][D_N]$$
$$[L_N] = -\lceil W \rfloor + 2\lceil W \rfloor [QB][D_N][B^T Q]^T \lceil W \rfloor$$
$$\{S_N\} = -2[Q^T W]\{d\}_N.$$

During the backward phase of the computations, the quantities

$$[D_n B^T]\{S_n\}, \quad [K_n^T]$$

are stored either to disk or in-core. These can then be used in the forward calculations. The forward stage of the computations is given by

$$\begin{aligned} \{g\}_n &= -[K_{n+1}^T][A]\{u\}_n - [D_{n+1} B^T]\{S_{n+1}\} \\ \{u\}_{n+1} &= [A]\{u\}_n + [B]\{g\}_n. \end{aligned} \tag{8}$$

This rearrangement of the equations should give exactly the same results as when using the Ricatti form. The primary restriction incurred is that it is assumed that the system matrix $[A]$ does not change between time increments.

This new form of the recursive relations deals with arrays of size

$$\begin{aligned} [D_n^{-1}] &= [m_g \times m_g] \quad \text{symmetric} \\ [D_n] &= [m_g \times m_g] \quad \text{symmetric} \\ [L_n] &= [m_d \times m_d] \quad \text{symmetric} \\ [Y_n] &= [m \times m_d] \\ [K_n] &= [m \times m_g] \\ \{S_n\} &= [m \times 1]. \end{aligned}$$

The largest arrays are of size $[m \times m_d]$ and $[m \times m_g]$, and since both m_d and m_g (number of sensors and number of forces, respectively) are significantly less than m, there is a huge reduction in the storage requirements.

The derived equations have the further advantage that they are more stable than the Ricatti form.[26,29]

The Computer Program `Inverse`

Both the Ricatti and time invariant equations are incorporated into a computer program called `Inverse`. Implicit and explicit versions of the time integrations are selectable.

Existing finite element code to produce the stiffness, mass, and damping arrays of 3-D thin-walled shell structures with reinforcements was modified and incorporated into `Inverse`; consequently it can handle the same problems as the original finite element program. The model of the structure is created using the mesh generating program associated with the finite element program.

Through the matrix $[Q]$, any number of mixed sensors can be used. In the case of the implicit scheme, the accelerometer data can be used directly; in the explicit scheme it needs to be integrated once to be put in the form of velocity.

Through the matrix $[B_g]$, any number of forces and moments can be determined. As seen in the equations of Appendix I, we actually determine the rate of force and not the force itself. This has the effect of adding regularization in the time direction; this is the only time regularization implemented. The choice of zero, first, and second order space regularization is implemented.

Note that if we store in-core then we need $[(m \times m_g) \times N]$ memory, where N is the number of time steps. For example, if the system size is 10000, with 20 unknown forces, and 2000 time steps, this can lead to over 3 GB of memory. Hence it is only for simple problems that in-core storage is performed. More details like this, and on the program in general, can be found in Ref. 28.

In order to exercise the program and determine some of the metrics, we now apply it to a cylinder problem using synthetic data. This will be used to demonstrate the ability of the program to handle large problems. It will also demonstrate the necessity of regularization.

This example problem, shown in Fig. 1, presents a complex problem, in that it is three dimensional with a large number of degrees of freedom. The elements used are such that there are six degrees of freedom at each node: three displacements, and three rotations. The total number of DoF for the mesh shown is 3456 and this translates into a total system size (state vector plus unknown forces) for the inverse problem as

implicit:	Size = 10375,	$\Delta t = 5.0\,\mu s$,	$N = 400$
explicit:	Size = 6375,	$\Delta t = 0.4\,\mu s$,	$N = 2000$.

These numbers are on the order of those presented in the introductory discussion.

The distributed load along the west edge is comprised of seven unknown forces with a pulse-like history. The sensors are not distributed in any particular or optimized manner for calculating these forces — indeed we purposely, for the results to be shown, did not use any sensors in the immediate location of the unknown forces. Also, the synthetic data were contaminated with Gaussian noise having a standard deviation of 10% of the average velocity.

Two sets of results are shown in Fig. 2 using different amounts of regularization. Regularization is seen to be an essential ingredient. The timings for the solutions are:

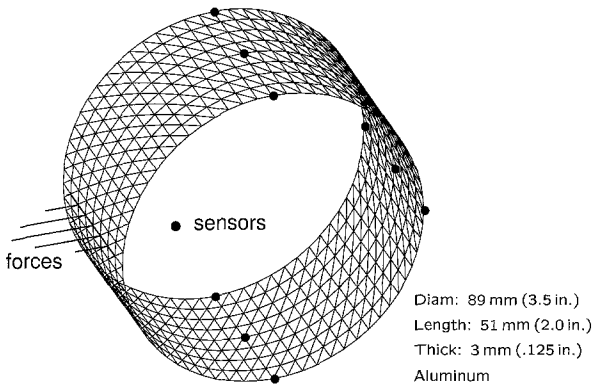


Fig. 1—Shell mesh with parameters

implicit:	53 minutes
explicit:	10.6 hours.

The relative cost between the backward and forward portions of the solution is approximately 8:1 for both integration schemes. Both sets of timings are significantly smaller than the naive scheme presented in the introduction. The difference between the two schemes is that the implicit method can use a time step related to the frequency content of the excitations, whereas the step size for the explicit method is dictated by the element size. In this instance, the advantage lies with the implicit method.

Experiment I: 3-D Cylindrical Shell

The first experiment is on an impacted shell. While only a single force is identified, it demonstrates some of the practical issues associated with using experimental data.

Experimental Setup

The set-up for the experiment is presented in Fig. 3. A 51 mm (2 in) long cylinder with 3 mm (1/8 in) wall thickness was constructed from aluminum. Accelerometers were placed on the top (90°), and 180° from the impact site, as shown in Fig. 3, and attached using bees' wax. The specimen was suspended by strings and impacted midway 25 mm (1 in) along the length.

Data for the experiments were collected by means of an Omega Instruments DAS-58 data acquisition card; this is a 12-bit card capable of 1 MHz sampling rate and storing up to 10^6 data points in the on-board memory. A total of 8 channels can be sampled; however, increasing the number of channels decreases the fastest possible sampling rate. For this experiment, four channels were sampled at 250 kHz, or a 4 μs time step.

Accelerometers (PCB 309A) and a modified force transducer (PCB 200A05) were used to collect the data. The connections between the accelerometers and the computer were through the use of an Omega Instruments BNC-58 multiplexing unit.

The accelerometer outputs were integrated according to the trapezoidal rule

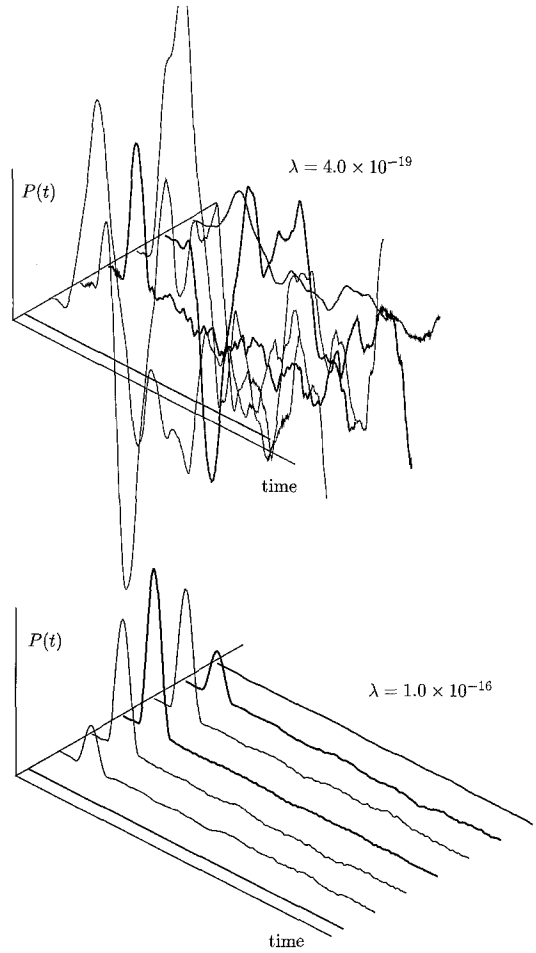


Fig. 2—Seven force reconstructions for the shell using different amounts of regularization

$$v_{n+1} = v_n + \tfrac{1}{2}(a_n + a_{n+1})\Delta t$$

to determine the velocities. Because of slight offsets in the acceleration voltages, a trend is sometimes observed in the velocity. This trend is estimated and subtracted; we refer to the detrended data as the raw data. The explicit integration option in `Inverse` requires a small Δt, and so the values of the input data are obtained from the measured data by linear interpolation. The implicit integration option does not require interpolation and can use the acceleration data directly.

The recorded force and (processed) velocity data are shown in Fig. 4. Note that the impact is a double impact on a very small time scale.

Forward Problem

A good inverse analysis is predicated on a high fidelity finite element model of the structure being analyzed. That is, this model must correspond accurately to the real experimental situation in terms of material properties (Young's modulus, mass density), dimensions (diameter, thickness), and sensor locations. Therefore, before any solution to an inverse problem is attempted, the modeling should be verified by doing a forward problem.

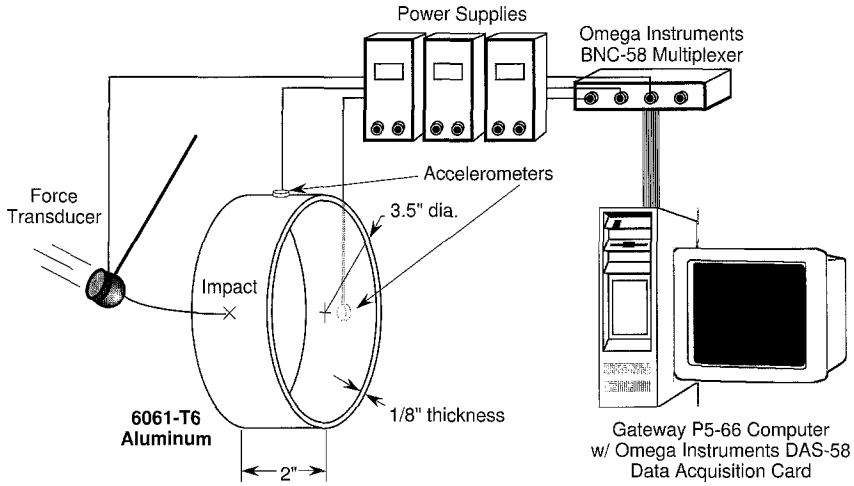


Fig. 3—Experimental set-up

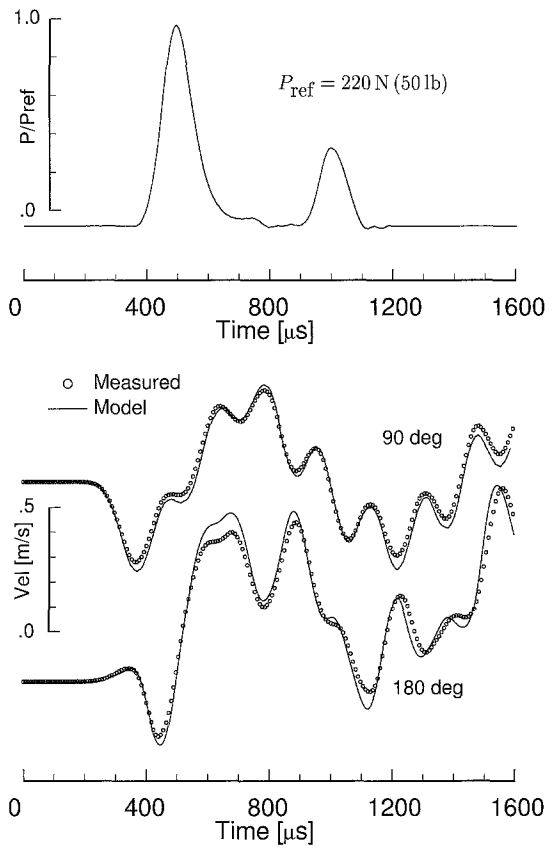


Fig. 4—Recorded force and velocity traces for the shell. Also shown is a comparison with the forward problem results

A convergence study was performed to determine the proper element size for the FEM model; the data from the force transducer provided the input force history for this study. From the results, it was determined that 3 mm (1/8 in) element modules would be an acceptable size for the `Inverse` analysis. A model was then created similar to that of Fig. 1 using 1408 folded plate elements. This creates a problem with a system size of 4486 degrees of freedom.

The results for the forward problem can be seen in Fig. 4. The two data sets are almost in phase with each other. To achieve these results, the effective diameter of the cylinder was taken at the average radius.

Inverse Study

The force reconstructions from the experimental velocity inputs are shown in Fig. 5. It is pleasing to note that the double impact is detected. Now, a word about the choice of regularization is in order. For transient problems, it can always be arranged that there is an initial zero force header, or an initial period of quiescence. On reconstruction with perfect data, this would be zero; with noisy data this would oscillate about zero; and with poor parameters this will show a trend. We tweak the regularization to give the best looking header.

Accelerometer #2 on its own does a good job but the same is not true for Accelerometer #1. At first sight, this might appear to indicate that somehow these data are corrupted. However, the fact that using both accelerometers together gives somewhat improved results does not bear this out. If accelerometer #1 was indeed contaminated, then mixing both accelerometer data would also show the contamination. A more reasonable conjecture is that position #1 is very sensitive, in an ill-conditioning sense, to slight errors in the data.

Our conjecture has important implications for inverse methods, so we decided to further investigate the nature of the discrepancy by manipulating synthetic data in a couple of ways and comparing the performance of positions #1 and #2.

First, the sensitivity due to the positioning of the accelerometer in relation to the node used in the `Inverse` analysis was examined. These results are shown in Fig. 6 and demonstrate that, for a given misalignment, the 90° position is more sensitive than the 180° position. However, the symptom of the error (a constant frequency superposition) does not reflect the symptom shown in the experimental results.

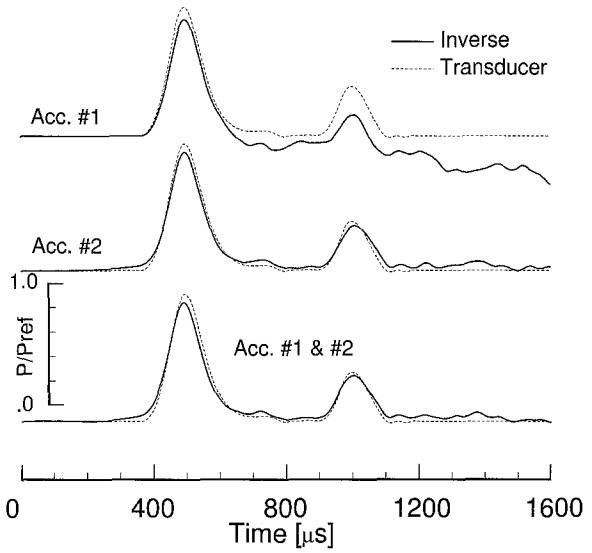


Fig. 5—Comparison of reconstructed forces from `Inverse` and force transducer data for the shell experiment

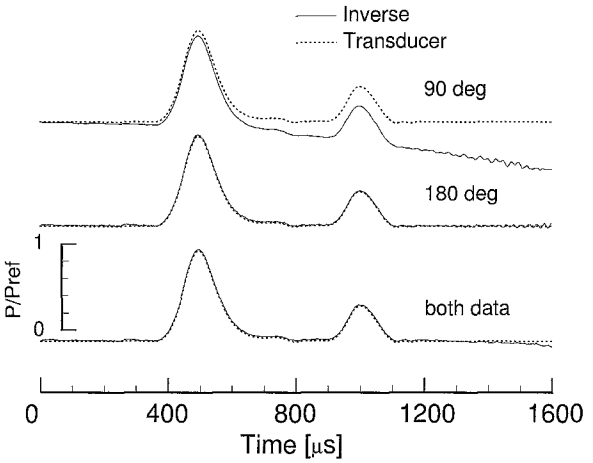


Fig. 7—Effect on force reconstructions of adding drift to velocity inputs

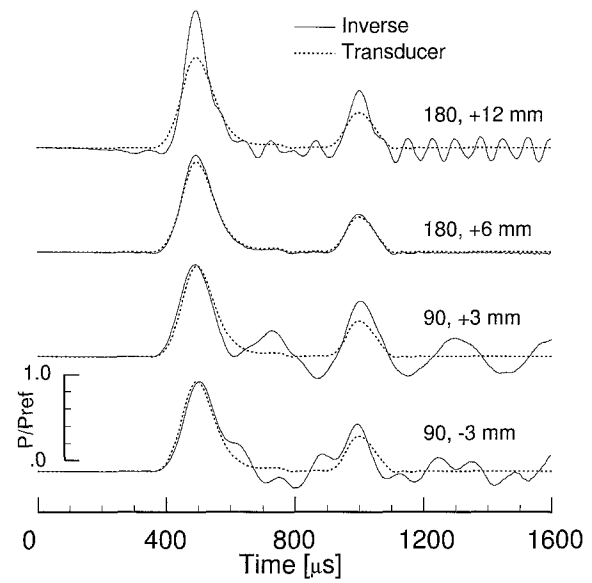


Fig. 6—Effect on force reconstructions of sensor mispositioning in the hoop direction

As a second study, approximately the same amount of drift seen in the experimental velocities was added to the velocities generated synthetically. This amount was determined by examining the raw data from the accelerometers and estimating the digitizing level of the voltages recorded. This value was found to be around 15 m/s (600 in/s). These adjusted velocities were used to reconstruct the force from the response at the two nominal 90° and 180° positions and the results are presented in Fig. 7. Again, the 90° position is more sensitive to slight deviations in the input data. This time the symptom is similar to that of the experimental results. It is noted that using both accelerometers does not give a simple average but gives a result that is often better than the individual results.

These studies indicate a concern that is quite crucial in inverse studies and is different from the experimental issues of position accuracy and signal fidelity. When doing an inverse analysis, the location of the sensor(s) and not just their accuracy is very important. This is not something that can always be decided in advance; for example, if the impact was at the 270° location then the roles of the accelerometers would be reversed. The implication is that multiple sensors must be used, and that they be combined through the use of regularization. Both of these aspects are part of the program `Inverse`.

Experiment II: In-Plane Problem

The second experiment demonstrates the ability of the method to determine multiple uncorrelated forces. It is not convenient to arrange multiple impacts of a structure; instead, we use multiple forces associated with the decoupling of a complex structure. This also allows a demonstration of the one-sided Hopkinson bar.

One-Sided Hopkinson Bar

In experimental verification studies, it often occurs that we need to apply a known force history to a structure. In some instances, where the frequency content of the excitation is low, we can use an instrumented hammer as is common in modal analysis,[30] and as done in the previous experiment. In other instances, however, we may need a high energy, high frequency input and here the common force transducers do not have the required specifications. The one-sided Hopkinson bar was designed for this purpose.

A widely used method for dynamic material analyses is by means of the split Hopkinson pressure bar (SHPB).[31] The standard configuration has evolved into one which has two long bars of identical material, instrumented with strain gages, and with a small specimen placed between them. The first rod, or input bar, is impacted by firing a projectile and a

stress wave is generated. This wave causes the specimen in between the two bars to be dynamically loaded. Typically, because the specimen is small, the wave propagation effects within the specimen are negligible, and it quickly reaches a uniform stress state. Under this circumstance, both the reflected and transmitted signals in the long bars have a simple relationship to the specimen stress (and strain) state. Using the strain gages to measure the waves propagated in both the input and output bars, the stress and strain histories of the specimen can be determined. Since the bars are such that the cross-sectional radius is not very large, dispersion effects are minimal and therefore the analysis can be done using 1-D rod theory.

In the one-sided Hopkinson bar configuration as shown in Fig. 8, the specimen is the complex structure for which we wish to apply the known force. A pulse of short duration is generated at one end of the bar, usually by impacting it. The important difference in comparison to the SHPB is that wave effects in the specimen are significant and consequently the reflected signal does not have a simple relation to the transmitted force. That is, since the duration of the event covers several passages of the original pulse up and down the incident bar, simple time shifting of the recorded signal is no longer adequate. Both the impacting force and the transmitted force must, in general, be determined simultaneously.

The one-sided Hopkinson bar was designed with the following parameters: It is 254 mm (100 in) long with BLH SR-4 semiconductor gages (model SNB1-16-35-S6) placed at the mid- and three-quarters points from the impacted end. The rod is made of steel with a cross-sectional diameter of 6.3 mm (0.25 in). The rather long length of the rod allows for pulses of long duration to be generated.

Inverse Problem

The finite element convergence study indicated a maximum element size of 6.3 mm (0.25 in) modules could be used in the analysis. The model and mesh are shown in Fig. 9, A symmetric model was used to allow for more elements of a smaller size; the number of elements for the resulting model was 2159 plate elements, which creates a problem with a system size of 3407 degrees of freedom.

The specimen was machined from a 6.3 mm (1/4 in) thick sheet of aluminum, cut to be a 305 mm (12 in) square with a 76 mm (4 in) diameter hole bored in its center. A small hole was drilled and tapped at the mid-point of one of the edges for the connection of the one-sided Hopkinson bar. Accelerometers were placed on the edge of the force input and on the opposite edge of the plate at 50.8 mm (2 in) off the force input line, as shown in Fig. 9. The accelerometer and strain gage data were collected using the data acquisition system of Fig. 3. The data were collected at a 4 μs rate over 4000 μs.

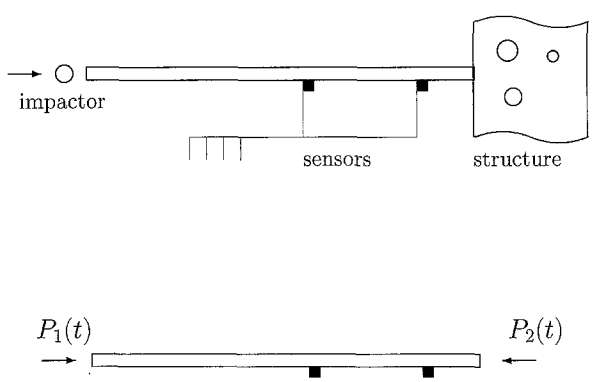


Fig. 8—The one-sided Hopkinson bar

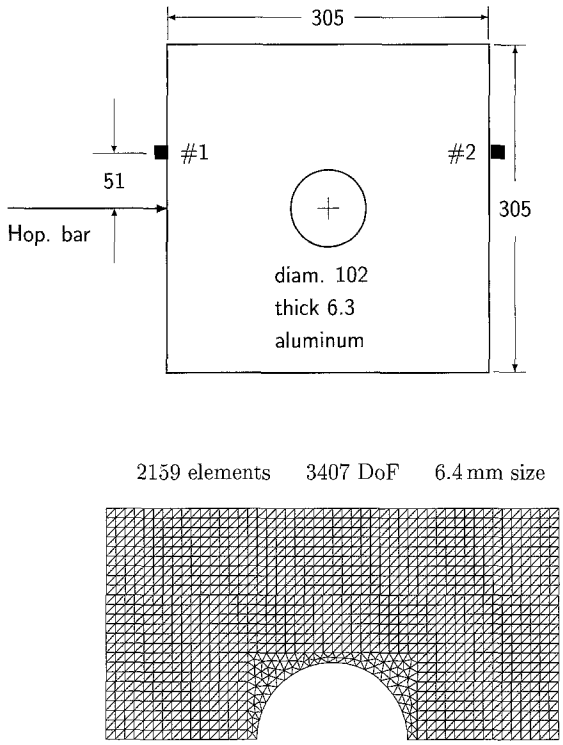
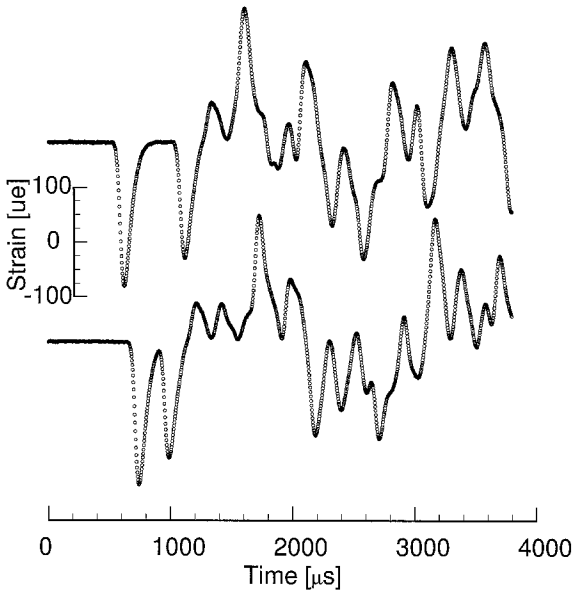


Fig. 9—Square plate with central hole (dimensions in mm). The Hopkinson bar is attached to the left side

The acceleration data were converted to velocities and detrended, and the strain gage data were modified to account for the slight nonlinearity of the Wheatstone bridge (since the resistance change of the semiconductor gages is relatively large). These data are shown in Fig. 10.

The inverse problem was divided into two separate problems corresponding to the bar and the plate. The data from the two strain gages on the Hopkinson bar were used simultaneously to reconstruct the impact force and the force input to the plate. These force reconstructions are shown labelled as P and R^s, respectively, in Fig. 11. The impact force is as expected, in that it is essentially a single pulse input. The reconstructions of the force at the connection point, on the other hand, are quite complicated persisting with significant amplitude for the full duration of the recording. Note that the connection force is larger than the impact force because of the high impedance of the plate.

The R^s force from the Hopkinson bar was used as input to the forward problem for the plate and the computed velocities are compared to the measured ones in Fig. 10. The comparison is quite good considering the extended time



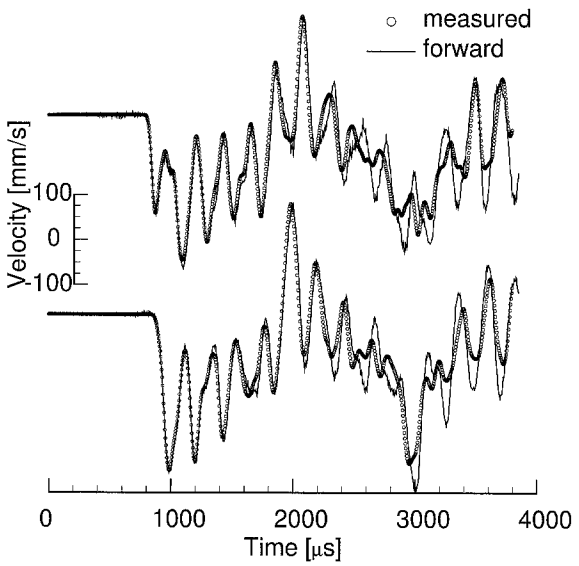


Fig. 10—Experimentally recorded data. Top, strains from Hopkinson bar. Bottom, velocities computed from accelerations

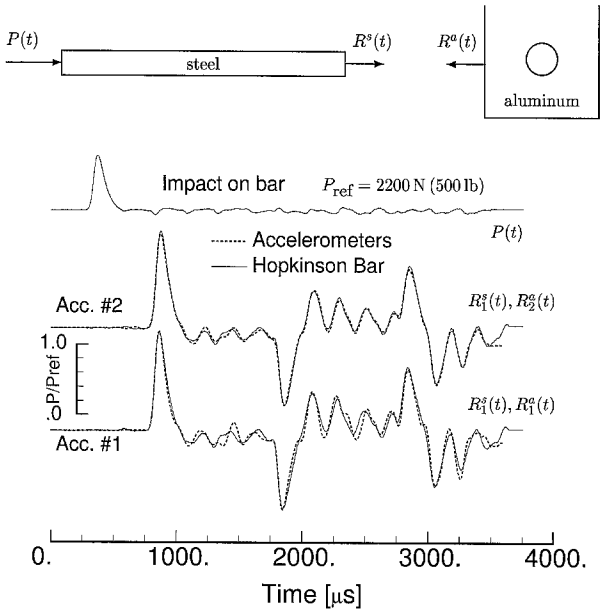


Fig. 11—Force reconstructions for the plate with a hole

period of the comparison. The data from the accelerometers were then used as separate single sensor inputs to reconstruct the force at the connection point. These force reconstructions are shown labelled as R_1^a and R_2^a in Fig. 11. The comparison of the reconstructed forces at the connection point agree quite closely both in character and magnitude. This is significant because the force transmitted across the boundary is quite complex and has been reconstructed from different sensor types placed on quite different structural types.

Discussion

This paper brings together a number of technologies to solve the force identification problem on complex structures. At the core is the embedding of the finite element modeling of complex structures to determine the system matrix. Consequently, structures as complex as is usually modeled with the finite element method can be handled conveniently. Special effort was made to insure that the computational costs scale in a manner similar to the forward problem using the finite element method. The problem of ill-conditioning is handled through regularization. This alleviates the burden of proper sensor placement and, as a result, the difficult problem of traction and pressure distributions can be solved. This opens up the exciting possibility of solving problems with unknown boundary conditions or using subdomains for the modeling. A preliminary investigation of these are given in Refs. 28 and 32.

Further efficiencies can still be achieved in the programming; for example, operations on the banded representation of the stiffness matrix can be replaced with skyline operations. This will decrease the storage requirements as well as increase the speed. An aspect of the programming that could have a profound effect in the future is that the backward solution stage (in particular, the matrices $[R_n]$ or $[K_n]$) actually solves for any force history at the stated force locations independent of the measured data. Thus inverse problems that require iteration (e.g., force location) or use multiple data sets (e.g., real-time health monitoring) need solve the backward stage only once (which is the expensive stage) and use the forward stage for manipulating the actual input data. The implications of this are significant.

The power of the presented method is that it embeds a general purpose FEM program in it. Ironically, this can also be perceived as a drawback, in that as the structural complexity increases (so that it requires a new element type, say) so must the complexity of the inverse coding also increase. Thus the power of commercial FEM programs cannot be utilized. In this regard, the methods presented in Refs. 11 and 33 offer a significant advantage because the FEM analysis is

run as an external process to the inverse program and therefore the power of commercial FEM codes can be leveraged. The methods do not scale as well as the one presented in this paper, so it will be interesting to see which of the attributes—scalability or generality—turns out to be more important.

Appendix I: Some Forms for [A] and [B]

We have implemented two schemes for the time discretization of the equations of motion. One is the conditionally stable explicit central difference method,[18,19] and the other is the unconditionally stable implicit Newark or constant acceleration method.[18,19]

Using the central difference finite difference scheme, the governing system of equations can be written as

$$\begin{Bmatrix} u \\ \dot{u} \\ g \end{Bmatrix}_{n+1} =$$

$$\begin{bmatrix} (I - M_1^{-1}K\Delta t^2) & (I - M_1^{-1}C\Delta t)\Delta t & M_1^{-1}B_g\Delta t^2 \\ -M_1^{-1}K\Delta t & (I - M_1^{-1}C\Delta t) & M_1^{-1}B_g\Delta t \\ 0 & 0 & I \end{bmatrix}$$

$$\begin{Bmatrix} u \\ \dot{u} \\ g \end{Bmatrix}_n + \begin{Bmatrix} 0 \\ 0 \\ \dot{g} \end{Bmatrix}_n \Delta t$$

with $[M_1] \equiv [M + \frac{1}{2}C\Delta t]$ and $[I]$ as the unit matrix. Note that the rate of force, $\{\dot{g}\}_n$, is taken as the actual applied load; this will give the equivalent of first order regularization in the time direction. This relation is abbreviated as

$$\{u\}_{n+1} = [A]\{u\}_n + [B]\{g\}_n \qquad \text{(A1)}$$

where n is the subscript over the discretized time. In this, the degrees of freedom vector, $\{u\}$, is of size $\{(m = 2m_u + m_g) \times 1\}$.

The system array $[A]$ is very large and therefore it is to our advantage to give special treatment to its manipulations. The structure of this matrix is shown in Fig. A1. Because of the nature of the structural systems, the stiffness matrix $[K]$ is banded and symmetric; consequently, the two upper left portions can be made banded (although not symmetric). As is usual with explicit discretization schemes, both the mass and damping matrices are taken as diagonal and hence the two upper middle portions are diagonal only. The first two bottom matrices are zero and the third is diagonal with unity. The remaining far left matrices are mostly zeros but because $m_g \ll m_u$ usually, we do not give it any special treatment.

We never do actually assemble the $[A]$ matrix but instead retain the individual component matrices. The storage used for the array is then

$$[(2m_u + m_g) \times (2m_u + m_g)] \quad \Longleftrightarrow \quad [m_u \times b + 2m_u].$$

This affords a substantial reduction for large systems.

The vector product $\{w\} = [A]\{v\}$ is accomplished by breaking $\{v\}$ into three parts $\{v_1, v_2, v_3\}^T$ and then adding

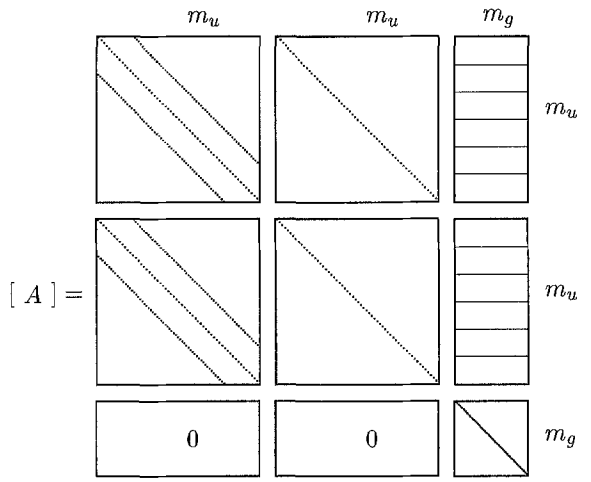


Fig. A1—Structure of the system matrix $[A]$

the separate products as

$$\{w_1\} = \{v_1\} - \lceil M_1^{-1}\Delta t^2 \rfloor [K]\{v_1\} + (\lceil v_2 - M_1^{-1}C\Delta t\, v_2 \rfloor + \lceil M_1^{-1}\Delta t \rfloor \{B_g\}\{v_3\})\Delta t$$

$$\{w_2\} = -\lceil M_1^{-1}\Delta t \rfloor [K]\{v_1\} + (\lceil v_2 - M_1^{-1}C\Delta t\, v_2 \rfloor + \lceil M_1^{-1}\Delta t \rfloor \{B_g\}\{v_3\})$$

$$\{w_3\} = \{v_3\}.$$

The only expensive computation is the product $[K]\{v_1\}$, and this is accomplished in banded form. Also, it is re-used for both $\{w_1\}$ and $\{w_2\}$. The middle and right parenthesis terms are also re-used. This is coded in its own subroutine. The vector product $\{w\} = [A]^T\{v\}$ is accomplished in a similar manner with its own subroutine. Note that the additional products $\{w\} = \{v\}^T[A] = \{[A]^T\{v\}\}^T$ and $\{w\} = \{v\}^T[A]^T = \{[A]\{v\}\}^T$ can be obtained using the above two subroutines.

In this way, both the storage and computational cost of manipulating the system array $[A]$ are reduced considerably. The complete set of arrays associated with describing the system are of size

$$\{u\} = \{(m = 2m_u + m_g) \times 1\}$$
$$\{g\} = \{m_g \times 1\}$$
$$[B_g] = [m_u \times m_g]$$
$$[B] = [m \times m_g]$$
$$\lceil K \rfloor = [m_u \times b] \qquad \text{banded, symmetric}$$
$$\lceil M \rfloor = \{m_u \times 1\} \qquad \text{diagonal}$$
$$\lceil C \rfloor = \{m_u \times 1\} \qquad \text{diagonal.}$$

As is the case with the forward finite element problem, the largest array is the stiffness matrix.

Using the Newmark implicit scheme, the governing system of equations can be written as

$$\begin{Bmatrix} u \\ \dot{u} \\ \ddot{u} \\ g \end{Bmatrix}_{n+1} =$$

$$\begin{bmatrix} Z_1 & \Delta t Z_2 & \Delta t^2 Z_3 & Z_4 \\ \frac{2}{\Delta t}(Z_1 - I) & (2Z_2 - I) & 2\Delta t Z_3 & \frac{2}{\Delta t} Z_4 \\ \frac{4}{\Delta t^2}(Z_1 - I) & \frac{4}{\Delta t}(Z_2 - I) & (4Z_3 - I) & \frac{4}{\Delta t^2} Z_4 \\ 0 & 0 & 0 & I \end{bmatrix}$$

$$\begin{Bmatrix} u \\ \dot{u} \\ \ddot{u} \\ g \end{Bmatrix}_n + \begin{Bmatrix} Z_4 g \\ \frac{2}{\Delta t} Z_4 g \\ \frac{4}{\Delta t^2} Z_4 g \\ \dot{g} \end{Bmatrix}_n \Delta t$$

with

$$Z_1 = [K^{-1}][2C\Delta t + 4M]/\Delta t^2,$$

$$Z_2 = [K^{-1}][C\Delta t + 4M]/\Delta t^2,$$

$$Z_3 = [K^{-1}][M]/\Delta t^2,$$

$$Z_4 = [K^{-1}][B_g]$$

and $[K_1] = [K + 2C/\Delta t + 4M/\Delta t^2]$. This can also be abbreviated as Eq (A1).

As in the explicit case, we never do actually assemble the $[A]$ matrix, but instead retain the individual component matrices, and the matrix products are performed in an efficient manner. The expensive computations are the products $[K_1^{-1}][M_i]\{v_j\}$. We do not form the inverse of the stiffness matrix (since this would be too expensive and require too much memory); rather, the product is accomplished in banded form as follows:

$$\{z\} = [K_1^{-1}][M_i]\{v_j\} \quad \text{or} \quad [K = UDU]\{z\} = \{M_i v_j\}.$$

The $[UDU]$ decomposition is performed only once, thus the computational cost is essentially that of the back-substitutions. This is the same cost as occurs in the forward implicit problem. The arrays associated with describing the implicit system are similar in size to that of the explicit scheme except that size $\{u\} = \{(m = 2m_u + m_g) \times 1\}$.

Appendix II: Minimizing Principle with Regularization

The usual least squares procedure is posed in the form of finding the set of discretized unknowns $\{u\}$ given the set of data $\{d\}$ so that the positive functional $\mathcal{A} = \chi^2$ is a minimum; that is,

$$\text{minimize:} \quad \mathcal{A} = [d - Au]^T[d - Au].$$

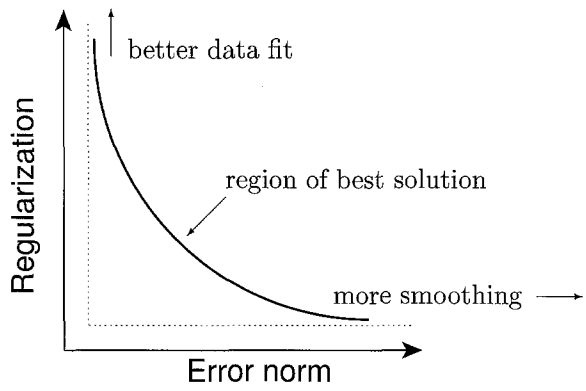


Fig. A2—Trade-off curve

The number of datapoints is typically larger than the number of unknowns and therefore minimizing with respect to the unknown $\{u\}$ gives a determinate system of equations. It is actually very difficult to get robust answers for these problems (even when there is an excess of data) because they are highly ill-conditioned. Regularization is used to make ill-conditioned problems better conditioned.

We now paraphrase some of the discussion of regularization as given in Ref. 3; a more mathematical treatment (plus an extensive bibliography) is given in Ref. 34. The essential idea in our inverse theory is the objective

$$\text{minimize:} \quad \mathcal{A} + \lambda\mathcal{B} = \{d - Au\}^T\{d - Au\} + \lambda\{u\}^T[H]\{u\}$$

for various values of $0 < \lambda < \infty$ along the so-called trade-off curve of Fig. A2. There are two positive functionals, $\mathcal{A}$ and $\mathcal{B}$. The first measures the agreement of the model to the data; when $\mathcal{A}$ by itself is minimized, the agreement with the data becomes very good but the solution becomes unstable. This reflects that $\mathcal{A}$ alone typically defines an ill-conditioned minimization problem. The second term $\mathcal{B}$ is introduced to overcome this problem; it measures the smoothness of the desired solution or sometimes a quantity reflecting *a priori* judgments about the solution. This function is called the regularizing operator. Minimizing $\mathcal{B}$ by itself gives a solution that is well-behaved but clearly does not reflect the data.

Increasing λ pulls the solution away from minimizing χ^2 in favor of minimizing $\{u\}^T[H]\{u\}$. We settle on a "best" value of λ by some criterion ranging from fairly objective to entirely subjective; this is discussed a little further in the main text.

The regularization method we use is generally called Tikhonov[35–37] regularization. Typically, the functional $\mathcal{B}$ involves some measures of smoothness that derive from first or higher derivatives and given by $[H] = [D]^T[D]$ where $[D]$ is the difference matrix. Detailed discussion of the different choices of regularization is given in Refs. 28 and 32 and therefore will not be elaborated on here.

The arrays associated with describing the regularization are of size

$$[D] = [(m_g - 1) \times m_g]$$
$$[H] = [m_g \times m_g]$$

Both arrays are relatively small since usually $m_g \ll m_u$.

Acknowledgments

This work was supported in part by a U.S. Army Multi-Disciplinary University Research Initiative (United States Army Grant No. DAAH04-96-10331) awarded to Purdue University.

References

1. Stevens, K.K., "Force Identification Problems: An Overview," Proceedings of SEM Spring Meeting, Houston, 838–844 (1987).

2. Martin, M.T. and Doyle, J.F., "Impact Force Identification from Wave Propagation Responses," International Journal of Impact Engineering, **18**, *65–77 (1996).*

3. Press, W.H., Flannery, B.P., Teukolsky, S.A., and Vetterling, W.T., Numerical Recipes, 2nd ed., Cambridge University Press, Cambridge (1992).

4. Doyle, J.F., "Further Developments in Determining the Dynamic Contact Law," EXPERIMENTAL MECHANICS, **24**, *265–270 (1984).*

5. Doyle, J.F., "Determining the Contact Force during the Transverse Impact of Plates," EXPERIMENTAL MECHANICS, **27**, *68–72 (1987).*

6. Doyle, J.F., "Force Identification from Dynamic Responses of a Bi-Material Beam," EXPERIMENTAL MECHANICS, **33**, *64–69 (1993).*

7. Michaels, J.E. and Pao, Y-H., "The Inverse Source Problem for an Oblique Force on an Elastic Plate," Journal of Acoustical Society of America, **77***(6), 2005–2011 (1985).*

8. Rizzi, S.A. and Doyle, J.F., "Spectral Analysis of Wave Motion in Plane Solids with Boundaries," Journal of Vibration and Acoustics, **114**, *133–140 (1992).*

9. Rizzi, S.A. and Doyle, J.F., "A Spectral Element Approach to Wave Motion in Layered Solids," Journal of Vibration and Acoustics, **114**, *569–577 (1992).*

10. Danial, A.N. and Doyle, J.F., "A Massively Parallel Implementation of the Spectral Element Method for Impact Problems in Plate Structures," Computing Systems in Engineering, **5**, *375–388 (1994).*

11. Doyle, J.F., "A Wavelet Deconcolution Method for Impact Force Identification," EXPERIMENTAL MECHANICS, **37**, *404–408 (1997).*

12. Trujillo, D. M., "Application of Dynamic Programming to the General Inverse Problem," International Journal for Numerical Methods in Engineering, **12**, *613–624 (1978).*

13. Busby, H.R. and Trujillo, D.M., "Numerical Solution to a Two-Dimensional Inverse Heat Conduction Problem," International Journal for Numerical Methods in Engineering, **21**, *349–359 (1985).*

14. Trujillo, D.M. and Busby, H.R., "Optimal Regularization of the Inverse Heat-Conduction Problem," Journal of Thermophysics, **3** *(4), 423–427 (1989).*

15. Busby, H.R. and Trujillo, D.M., "Solution of an Inverse Dynamics Problem using an Eigenvalue Reduction Technique," Computers & Structures, **25** *(1), 109–117 (1987).*

16. Hollandsworth, P.E. and Busby, H.R., "Impact Force Identification using the General Inverse Technique," International Journal of Impact Engineering, **8**, *315–322 (1989).*

17. Trujillo, D.M. and Busby, H.R., Practical Invere Analysis in Engineering, CRC Press, New York (1997).

18. Doyle, J.F., Static and Dynamic Analysis of Structures, Kluwer, Dordrecht, The Netherlands (1991).

19. Doyle, J.F., Nonlinear Analysis of Thin-walled Structures: Statics, Dynamics, and Stability, Springer-Verlag, New York (2001).

20. Bellman, R. and Kalaba, R., Dynamic Programming and Modern Control Theory, Academic Press, New York (1965).

21. Lapidus, L. and Luus, R., Optimal Control of Engineering Processes, Blaisdell Publishing Company, Waltham, Mass. (1967).

22. Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," ASME Journal of Basic Engineering, **82D**, *35–45 (1960).*

23. Kalman, R.E., New Methods in Wiener Filtering Theory, In Engineering Applications of Random Function Theory, J.L. Bogdanoff and F. Kozin, editors, 270–388, Wiley, New York (1963).

24. Jazwinski, A. H., Stochastic Processes and Filtering Theory, Academic Press, New York (1970).

25. Ince, E.L., Ordinary Differential Equations, Dover, New York (1956).

26. Kailath, T., "Some New Algorithms for Recursive Estimation in Constant, Linear, Discrete-Time Systems," IEEE Transactions on Information Theory, **AC–19** *(4), 83–92 (1973).*

27. Morf, M. and Sidhu, G. and Kailath, T., "Some New Algorithms for Recursive Estimation in Constant, Linear, Discrete-Time Systems," IEEE Transactions on Automatic Control, **AC-19** *(4), 315–323 (1974).*

28. Adams, R.A., Force Identification in Complex Structures, M.S. Thesis, Purdue University (1999).

29. Mehra, M. K., Practical Aspects of Designing Kalman Filters, In Applications of Kalman Filter Theory to Hydrology, Hydraulics and Water Resources, C. L. Chiu, editor, 89–115, University of Pittsburgh, Pittsburgh (1978).

30. Ewins, D.J., Modal Testing: Theory and Practice, Wiley & Sons, New York (1984).

31. Kolsky, H., Stress Waves in Solids, Dover, New York (1963).

32. Cho, S-M., A Sub-Domain Inverse Method for Dynamic Crack Propagation Problems, M.S. Thesis, Purdue University (2000).

33. Doyle, J.F., "Reconstructing Dynamic Events from Time-limited Spatially Distributed Data," International Journal for Numerical Methods in Engineering, to appear (2002).

34. Neumaier, A., "Solving Ill-conditioned and Singular Linear Systems: A Tutorial on Regularization," Society for Industrial and Applied Mathematics, **40** *(3), 636–666 (1998).*

35. Tikhonov, A.N. and Arsenin, V.Y., Solutions of Ill-Posed Problems, Wiley & Sons, New York (1977).

36. Alifanov, O. M., "Methods of Solving Ill-Posed Inverse Problems," Journal of Engineering Physics, **45** *(5), 1237–1245 (1983).*

37. Martinez, Y. and Dinth, A., "A Generalization of Tikhonov's Regularizations of Zero and First Order," Computers and Mathematics with Applications, **12B** *(5/6), 1203–1208 (1986).*