Chapter 2

Introduction to Inverse Dynamics

2.1 Introduction

In structural dynamics the forward problem is concerned with estimating the response of the structure with respect to time to a known forcing function. The inverse problem is concerned with estimating the forcing functions that cause the system to best match the known or measured response. The procedure utilised to solve this type of inverse problem is known as the method of general inverse theory, (Trujillo 1977, Trujillo & Busby 1997) and uses three areas of mathematics. These are least squares minimisation with regularisation known as Tikhonov regularisation, dynamic programming which provides an efficient solution to the least squares problem and finally the L-curve method to calculate the optimal regularisation parameter. This chapter deals with each of these concepts separately, and how all three methods act together to predict the unknown forcing function from given measurements on the structure.

2.2 State Space Equations

In order to implement the methods stated above it is necessary to convert the equilibrium equation of motion into a discrete time integration scheme. The second order matrix differential equation for structural dynamics is defined as,

$$[M_g]\left\{\frac{d^2y}{dt^2}\right\}+[C_g]\left\{\frac{dy}{dt}\right\}+[K_g]\{y\}=F(t) \tag{2.1}$$

where $[M_g]$ is the $[n_{dof} \times n_{dof}]$ consistent or lumped mass matrix, $[C_g]$ is the $[n_{dof} \times n_{dof}]$ damping matrix, $[K_g]$ is the $[n_{dof} \times n_{dof}]$ stiffness matrix, y is the $[n_{dof}\ x\ 1]$ degree of freedom vector and $F(t)$ is the $[n_{dof} \times 1]$ vector of forcing functions. This equation is non-singular when the boundary conditions are applied for an n_{dof} degree of freedom (dof) system. There are various numerical methods to convert the equilibrium equation of motion into a discrete time integration scheme (Newmark method, backward implicit method and the exponential matrix). It has been shown by Trujillo and Busby (1997) that the exponential matrix for the inverse dynamics problem is the most accurate numerical method. The equation of motion is rewritten in the form,

$$[M_g]\{a\}+[C_g]\{v\}+[K_g]\{u\}=[L]g(t) \tag{2.2}$$

where $[L]$ is the $[n_{dof} \times m_g]$ location matrix, which assigns the forces of the vector $g(t)$ $[m_g \times 1]$ to the degrees of freedom, and $\{a\},\{v\}$ and $\{u)$ are the acceleration's velocities and displacements of the degrees of freedom respectively. It is necessary to make two changes to the variables defined in equation (2.2). These are,

$$\{a\}=\left\{\frac{dv}{dt}\right\}=\{\dot{v}\} \tag{2.3}$$

$$\{v\}=\left\{\frac{du}{dt}\right\}=\{\dot{u}\} \tag{2.4}$$

Substituting $\{\dot{v}\}$ into equation (2.2) gives,

$$[M_g]\{\dot{v}\}+[C_g]\{v\}+[K_g]\{u\}=[L]\{g(t)\} \tag{2.5}$$

Solving equation (2.5) for $\{\dot{v}\}$ gives

$$\{\dot{v}\}=-[M_g]^{-1}[C_g]\{v\}-[M_g]^{-1}[K_g]\{u\}+[M_g]^{-1}[L][g(t)] \tag{2.6}$$

For an n_{dof} degree of freedom system the equation of motion can be written as a matrix differential equation of the form,

$$\frac{d\{X\}}{dt}=[A]\{X\}+f(t) \tag{2.7}$$

where $\{X\}$ is the $[\,2n_{dof} \times 1]$ vector of state variables containing both displacements and velocities of all the degrees of freedom and:

$$\{X\}=\begin{Bmatrix}u\\v\end{Bmatrix} \tag{2.8}$$

Rearranging equations (2.4) and (2.6) gives a system of equations of the form,

$$\frac{d}{dt}\begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{bmatrix} 0 & I \\ -[M_g]^{-1}[K_g] & -[M_g]^{-1}[C_g] \end{bmatrix} \begin{Bmatrix} u \\ v \end{Bmatrix} + \begin{bmatrix} 0 \\ [M_g]^{-1}[L] \end{bmatrix} \{g(t)\} \tag{2.9}$$

which is of the form of equation (2.7), with:

$$A = \begin{pmatrix} 0 & I \\ -[M_g]^{-1}[K_g] & -[M_g]^{-1}[C_g] \end{pmatrix} \tag{2.10}$$

and

$$f(t) = \begin{bmatrix} 0 \\ [M_g]^{-1}[L] \end{bmatrix} \{g(t)\} \tag{2.11}$$

This matrix vector form of the equation of motion is then converted into a discrete time integration scheme using the exponential matrix.

2.2.1 Solution to the Matrix Differential Equation

Defining the initial condition of equation (2.7) as,

$$X(0) = c \tag{2.12}$$

and taking the Laplace transform of both sides of equation (2.7) results in:

$$s\{X(s)\} - \{X(0)\} = [A]\{X(s)\} + \{f(s)\} \tag{2.13}$$

Solving for *X(s)* yields,

$$\{X(s)\} = [[I]s - [A]]^{-1}(\{X(0)\} + \{f(s)\}) \tag{2.14}$$

Taking the inverse Laplace transform of equation (2.14), the solution can be defined by,

$$\{X(t)\} = L^{-1}([[I]s - [A]]^{-1}\{X(0)\}) + L^{-1}([[I]s - [A]]^{-1}\{f(s)\}) \tag{2.15}$$

where

$$L^{-1}([[I]s - [A]]^{-1}\{X(0)\}) = e^{At}\{X(0)\} \tag{2.16}$$

and

$$e^{[A]t} = \sum_{k=0}^{\infty} \frac{1}{k!}([A]t)^k = [A]^0 t^0 + [A]t + \frac{1}{2!}[A]^2 t^2 + \frac{1}{3!}[A]^3 t^3 ..... \tag{2.17}$$

Equation (2.17) is what is commonly referred to as the exponential of a matrix, and arises in the solution to a system of differential equations; its calculation is numerically complex (Moler & Loan 1978, 2003, Fung 2004) and is not included here. Now from the convolution integral, the second inverse Laplace transform in equation (2.15) can be found by writing it as,

$$L^{-1}([[I]s - [A]]^{-1}\{f(s)\}) = [[I]s - [A]]^{-1} * f(s) = \int_0^t e^{A(t-\tau)} f(\tau) d\tau \tag{2.18}$$

Substituting equations (2.18) and (2.16) into equation (2.15) gives,

$$X(t) = e^{[A]t} X(0) + \int_0^t e^{[A](t-\tau)} f(\tau) d\tau \tag{2.19}$$

Alternatively equation (2.19) can be found by using e^{At} as an integration factor (Appendix A, Trujillo & Busby 1997, Bellman 1967a, 1967b, 1997, Boyce & DiPrima 1977). In order to solve equation (2.19) numerically it is necessary to replace the interval of *0* to *t* with an equivalent discrete form, essentially the integral from *0* to *t* will be replaced by an integral from *t* to *t+h*. Replacing the values at a time (*t*) with those at (*t+h*) gives:

$$X(t+h) = e^{[A](t+h)}X(0) + \int_0^{t+h} e^{[A](t+h-\tau)} f(\tau) d\tau \tag{2.20}$$

equation (2.19) can be shown to be equal to,

$$e^{[A](t+h)}X(0) = e^{[A]h}X(t) - \int_0^{t} e^{[A](t+h-\tau)} f(\tau) d\tau \tag{2.21}$$

Now substituting equation (2.21) into (2.20) gives

$$X(t+h) = e^{[A]h}X(t) + e^{[A]h}e^{[A]t} \int_t^{t+h} e^{-[A]\tau} f(\tau) d\tau \tag{2.22}$$

The solution given by equation (2.22) is a unique solution, where no approximations have been introduced. Introducing what are known as Pade Approximations (Trujillo 1975, 1997), it is assumed that $f(\tau)$ is constant over the interval (t, $t+h$). This leads to the further approximation that for h suitably small,

$$f(\tau) = f(t), t \le \tau \ge t+h \tag{2.23}$$

By approximating the forcing function in this fashion, the final part of the integrand can be completed by integrating with respect to the exponential matrix, as the forcing function is assumed constant over the time interval. Integrating the second function in equation (2.22), assuming $f(\tau)$ is now constant in this time interval, gives,

$$\int_t^{t+h} e^{-[A]\tau} d\tau = -[A]^{-1} e^{-[A]\tau} \Big]_t^{t+h} \tag{2.24}$$

$$\int_t^{t+h} e^{-[A]\tau} d\tau = -[A]^{-1} e^{-[A](t+h)} + [A]^{-1} e^{-[A](t)} \tag{2.25}$$

$$\int_t^{t+h} e^{-[A]\tau} d\tau = -e^{[A]t}[[A]^{-1} e^{-[A]h} + [A]^{-1}] \tag{2.26}$$

Substituting this into equation (2.22) and making use of the fact that the product of a matrix exponential and its inverse is the identity matrix gives:

$$X(t+h) = e^{[A]h} X(t) + ([A]^{-1} e^{[A]h} - [A]^{-1}) f(\tau) \tag{2.27}$$

Replacing *X(t)* with X_j , where X_j denotes the solution at time ($j \times h$), equation (2.28) becomes,

$$X_{j+1} = e^{[A]h} X_j + ([A]^{-1} e^{[A]h} - [A]^{-1}) f_j \tag{2.28}$$

Let the system matrix [*M*] defining the dynamics of the system, be defined by,

$$[M] = e^{[A]h} \tag{2.29}$$

Substituting [*M*] into equation (2.30) yields,

$$X_{j+1} = MX_j + [M - I][A]^{-1} f_j \tag{2.30}$$

The second term on the right hand side of equation (2.30) relates the forces to the degrees of freedom, by making use of the fact that,

$$f_j = \begin{bmatrix} 0 \\ [M_g]^{-1}[L] \end{bmatrix} \{g_j\} \tag{2.31}$$

the second term on the right hand side of equation (2.30) can be defined by,

$$[P] = [M - I][A]^{-1} \begin{pmatrix} 0 \\ [M_g]^{-1}[L] \end{pmatrix} \tag{2.32}$$

and the final system is represented by,

$$\{X\}_{j+1} = [M]\{X\}_j + [P]\{g\}_j \tag{2.33}$$

2.3 Regularisation

In many areas of mathematics and engineering, the need arises to solve a system of equations of the form,

$$[A]\{x\} = \{b\} \tag{2.34}$$

where A is an $[m \times n]$ matrix, b is an $[m \times 1]$ vector and x is an $[n \times 1]$ vector to be determined. The least squares approximation to this set of linear equations can be found by minimising the Euclidian norm of the residual, where the residual is defined by;

$$r = Ax - b \tag{2.35}$$

The Euclidian norm of the residual is then defined by Stoer & Burlisch (1992) as,

$$\left\| Ax - b \right\|_2 = \left\| r \right\|_2 = \left[\sum_{i=1}^{m} r_i^2 \right]^{1/2} \tag{2.36}$$

By minimising this residual norm with respect to x it can be shown that the solution to equation (2.36) is given by,

$$[A^T A]\{x\} = [A^T]\{b\} \tag{2.37}$$

In many cases $[A]$ represents a model of a system under investigation and $\{b\}$ represents a data vector of measured variables. The inverse problem then consists of determining the unknown vector $\{x\}$ that best fits the measured $\{b\}$. When equation (2.37) arises, it is often the result of a discretization of an inverse problem. The system could be the Bridge weigh in motion equations, (Moses 1970), where $\{b\}$ is a vector of strain measurements at some point on the bridge, and $[A]$ is a matrix of influence ordinates for the unit response of the measurements, and $\{x\}$ is the vector of Axle weights. If the inverse problem is formulated as in equation (2.37) it may be highly ill conditioned (Wheatly 1996, Golub & Van Loan 1989), meaning that very small perturbations in the

matrix [A] or the vector $\{b\}$, cause very large fluctuations in the solution vector $\{x\}$. Whilst this form of the inverse equation is ill conditioned it may not necessarily be what is coined as ill-posed, (Tikhonov & Anrsen 1968). An ill-posed inverse problem in the general linear algebraic results from a poor formulation of the inverse problem, where the matrix [A] is poorly formulated and thus even without small perturbations in the matrix [A] or the vector $\{b\}$ there could still be significant fluctuations in the solution vector $\{x\}$ resulting from the huge conditioning number of the matrix [A]. An ill-posed problem is ill conditioned and neither a simple nor a complicated reformulation of the problem will improve the conditioning of the problem significantly. However an ill-conditioned inverse problem may not necessarily be ill posed and thus a re-formulation of the problem may greatly improve the conditioning of the system. Fortunately using the mathematical tool of regularisation it is possible to obtain accurate solutions to ill-conditioned and ill-posed problems.

The method of regularisation often referred to as Tikhonov regularisation was developed by Tikhonov and Anrsenin (1977). However it was pointed out by Neumaier (1999), that much of the early literature on regularisation is discussed "in terms of functional analytic language geared towards infinite-dimensional problems". For a finite dimensional inverse problem, the method of regularisation is best explained in a basic linear algebraic setting, (Hansen 1992, 1994, 1998)

2.3.1 Regularisation of Discrete Problems

The method of Tikhonov regularisation minimises a linear combination of:

$$x_\lambda = \min_x \{ || Ax - b ||_2 + \lambda || Hx ||_2 \} \tag{2.38}$$

where λ is the regularisation parameter and [H] is some form of a penalty operator or higher order smoothing matrix (Hansen 1998, Cullum 1979). The simple choice is to let [H] be the identity matrix and thus the regularisation restricts the norm of $\{x\}$. The regularised solution is that which minimises the residual norm $|| Ax - b ||_2$ and that of the solution norm $\lambda || Hx ||_2$. If λ is sufficiently small, equation (2.38) is very close to the minimum of equation (2.37). However, the addition of the second term can significantly reduce the ill-conditioning of the system and consequently reduce the error

inherent in equation (2.37). With the addition of this penalty function, the solution vector is no longer the solution of the linear system given in equation (2.37) but that of a new system, which seeks a fair balance between the residual norm and that of the solution norm. If $\lambda=0$, then the problem is that of minimising the standard least squares; if λ is very large the solution norm is small at the cost of a large residual norm, meaning the least squares error will be high. Thus solving involves a trade-off between the residual norm and the solution norm, this being determined by the single regularisation parameterλ. The solution to the Tikhonov regularisation method is found by minimising the function,

$$\Pi = (Ax_\lambda - b)^T (Ax_\lambda - b) + \lambda (H^T x_\lambda^T)(Hx_\lambda) \tag{2.39}$$

The regularised solution with respect to x is found from,

$$\frac{d\Pi}{dx_\lambda} = 2A^T (Ax_\lambda - b) + 2\lambda H^T H x_\lambda = 0 \tag{2.40}$$

The Tikhonov problem is now formulated as

$$(A^T A + \lambda H^T H) x_\lambda = A^T b \tag{2.41}$$

This is the same as equation (2.37) except for the term $\lambda H^T H$ on the left hand side. The regularised solution in terms of the normal equations is given as,

$$x_\lambda = (A^T A + \lambda H^T H)^{-1} A^T b \tag{2.42}$$

2.3.2 Choice of Regularisation Parameter

The final and most difficult stage in the regularisation of ill-posed problems is the choice of regularisation parameter. There are many methods in the current literature for calculating these parameters but most of them require some form of prior knowledge of the noise itself (see Golub & Van Loan 1989, Golub et al 1979). What is required is an efficient method for calculating the optimal regularisation parameter using only the data or measurements given. For the inverse dynamics problem there are two methods in the current literature for estimating the optimum regularisation parameter. These are

Generalised Cross-Validation (GCV) and the L-curve method (see Lawson and Hanson 1960, Hansen 1990 and, Trujillo & Busby 1991). Trujillo and Busby showed that when solving inverse problems that utilise finite element models, whether they are inverse heat conduction problems or inverse dynamic problems, the optimal approach is to use the L-curve criterion. The method was first proposed by Lawson and Hanson (1960) and further developed by Hansen (1990, 1992, 1994, 1999). The basic idea behind the L-curve is to plot the discrete smoothing norm of the regularised solution versus the residual norm of the error on a log-log scale. Hansen has shown that in producing this, the plot will continuously depend on the smoothing parameter. It will always have a corner and that the corner is the optimal regularisation parameter [see figure 2.1].

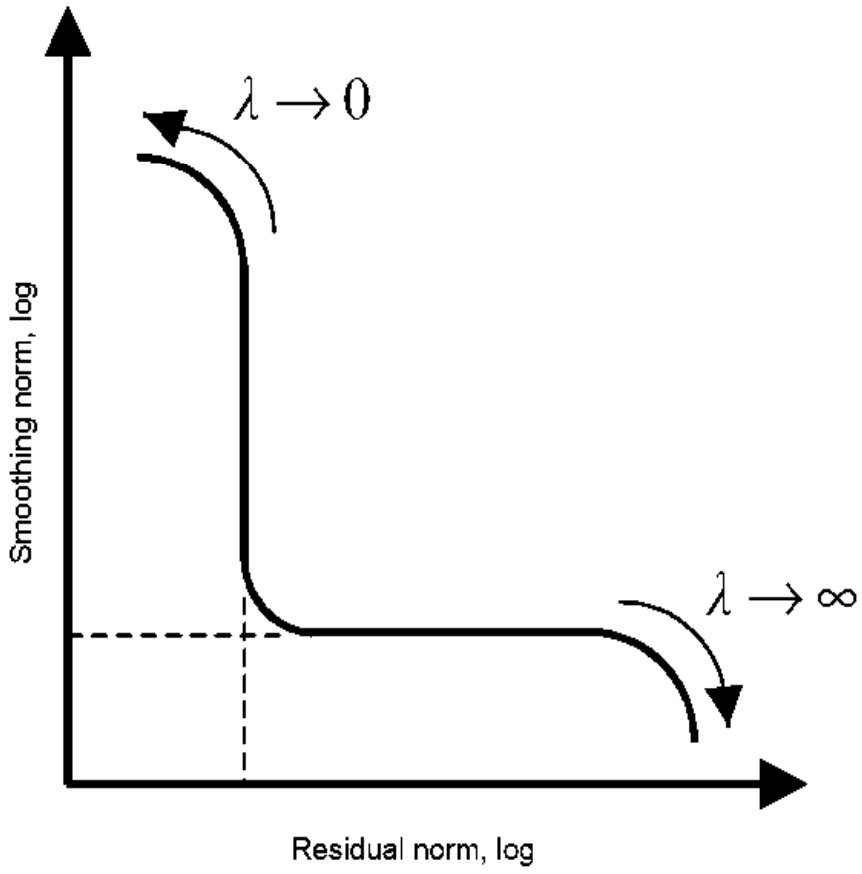

Figure 2.1 - Typical L-curve; a plot in log-log scales of the smoothing norm versus the residual norm of the solution.
(Hansen 1990)

2.4 Measurements and Problem Formulation

As defined previously the inverse dynamics problem is concerned with estimating the forces $\{g\}$ that cause the system to best match the given measurements. This section brings together both the recursive solution of the state space equations and the method of regularisation, coupled with the powerful method of dynamic programming (Bellman, 1965, 1967a, 1967b, 1997) to solve the inverse dynamics problem.

In the inverse dynamics problem it is normally not possible to take measurements on all of the state variables, as the state vector contains the displacements and velocities of all degrees of freedom (dof's) in the system. The common case is where the number of measurements is significantly less than the number of state variables. It is assumed that over the total time interval one or more measurements have been taken on the state variables. This is represented as a vector of measurements $\{d\}_j$. For each time interval, $\{d\}_j$ is an $[m_z \times 1]$ vector where m_z is the number of measurements. The measurements are then related to the state variables by,

$$d_j = QX_j \tag{2.43}$$

In this form $[Q]$ is a $[m_z \times 2n_{dof}]$ matrix, which selects the state variables from the state vector $\{X\}$ where the measurements have been taken. The $[Q]$ matrix is generally some form of a finite difference scheme related to the strain displacement matrix of the individual finite element used. The optimisation problem is to calculate the forcing functions $\{g_j\}$ that minimise the general least squares error given by,

$$E(X_j, g_j) = \sum_{j=1}^{N} ((QX_j - d_j), W(QX_j - d_j)) \tag{2.44}$$

subject to,

$$\{X\}_{j+1} = [M]\{X\}_j + [P]\{g\}_j \tag{2.45}$$

where $[W]$ is a general weighting matrix of size $[m_z \times m_z]$, usually the identity matrix, and (x,y) denotes the vector product. The (x,y) notation for vector products has been

introduced, as the derivation of the solution is better suited to notation of this type, the notation and vector product definitions are detailed in appendix B. Equation (2.44) is the least squares error $\| QX_j - d_j \|_2$ written as a vector product, the problem is to minimise the least squares error for each time step *j* subject to equation (2.45). It is possible to construct a global system of simultaneous equations to calculate the set of forces, which minimises the error of $E(X_j, g_j)$ (Doyle 2004). An optimisation scheme of this type is not suitable to tackle transient problems of this nature. For example, a transient problem with *N* time steps and *n* degrees of freedom, would involve the construction of a global matrix of size [$(N \times n) \times (N \times n)$]. Even for a system with only 40 dofs and 2000 time steps the global matrix would be of the order [$(8 \times 10^4) \times (8 \times 10^4)$]. The procedure is not really feasible. The procedure that has been adopted to solve transient problems of this nature is the method of dynamic programming.

From a mathematical viewpoint the least squares criterion is not in fact sufficient. This is due to the fact that the least squares error will produce an exact match to the data. Also inverse problems of this type are highly ill conditioned or ill-posed (Tikhonov & Arsenin 1977), meaning that the solution is very sensitive to small perturbations in measurements. In order to overcome this problem, a regularisation term is added to least squares error function. The regularisation term added to equation (2.44) can be defined in a similar manner to that of equation (2.39), given by,

$$E(X_j, g_j) = \sum_{j=1}^{N} ((QX_j - d_j), W(QX_j - d_j)) + \lambda(g_j, H^T H g_j) \qquad (2.46)$$

where the first term on the right hand side of equation (2.46) is the least squares error and the second term on the right hand side is the regularisation term. If it is assumed that the smoothing matrix [*H*] is the identity matrix, then equation (2.46) can be rewritten as,

$$E(X_j, g_j) = \sum_{j=1}^{N} ((QX_j - d_j), W(QX_j - d_j)) + (g_j, B g_j) \qquad (2.47)$$

where,

$$B = \lambda[H^T][H] = \lambda[I] \qquad (2.48)$$

The addition of the regularisation term allows control of the smoothness in the solution by varying the regularisation parameter [B], where B is a diagonal matrix of size [$m_g \times m_g$].

2.5 Dynamic Programming

The problem is now to minimise the function $E(X,g)$ over the sequence of forcing vectors $\{g_j\}$. This can be treated as a multi-dimensional control process via dynamic programming. In this case a control process as defined by Bellman (1967a) is considered to be a multistage decision process consisting of the following operations

(1) At time t (discretised as j), the state of the system is observed
(2) Based on this information and a control law, a decision is made
(3) This decision produces an effect upon the system

This leads to the idea that a minimisation problem can be treated as a control process. Instead of asking for a solution at each particular point in time, the most efficient control is sought, at each point in phase space. Using this idea, each decision can be regarded as the initial decision to a new control process. The function *E(X,g)* defined in equation (2.47) is what is called a policy function. A policy is a rule for prescribing the action to be taken at every possible position of the system in phase space. A policy that minimises the functional representing the overall cost of the control process is called an optimal policy

Principle of optimality; " An optimal policy has the property that whatever the initial state and initial decision are the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision" (Richard Bellman 1965, 1967 a & b)

The dynamic programming solution to the function E begins by defining the following function,

$$f_{n-1}(X) = \min_{g_{n-1}} E_{n-1}(X_{n-1}, g_{n-1}) \tag{2.49}$$

This function is the minimum value of E starting at any stage ($j = n$). Then the system is simulated to the end ($j = N$) using the optimal g_j's. Applying the principle of optimality to equation (2.49) yields a recurrence formula of the form,

$$f_{n-1}(X_{n-1}) = \min_{g_{n-1}}[(QX_{n-1} - d_{n-1}, W(QX_{n-1} - d_{n-1})) + (g_{n-1}, Bg_{n-1}) + f_n(X_n)] \quad (2.50)$$

This equation relates two functions, the optimal cost f_n and the optimal forcing term g_n. The minimum at any point is obtained by choosing the decision g_{n-1} to minimise the immediate cost, represented by the first and second terms and adding the remaining cumulative cost resulting from the previous decisions (the third term). In this way the minimisation is performed over a previously calculated optimal function f_n, so the minimisation is not only dependent on the current state of the system but the optimal cost of the previous state. In order to obtain a solution, this procedure is initiated at the end state $j = N$ and solved backwards to $j = 1$. The process starts at $j = N$, [it is assumed at this stage that $g_N = 0$] therefore,

$$f_N(X_N) = ((QX_N - d_N), W(QX_N - d_N)) \quad (2.51)$$

Expanding this and making note of the fact that (x,y) denotes the vector product, it can be shown that,

$$f_N(X_N) = (X_N, Q^T WQX_N) - (X_N, Q^T Wd_N) - (WQX_N, d_N) + (d_N, Wd_N) \quad (2.52)$$

Rearranging this as a quadratic in terms of X_N yields,

$$f_N(X_N) = (X_N, Q^T WQX_N) - 2(X_N, Q^T Wd_N) + (d_N, Wd_N) \quad (2.53)$$

The functional f for the last time step is a quadratic in terms of X_N. It is now necessary to show that all of the time steps are quadratic functions in terms of X_n where n is an arbitrary time step. The cost function for the $N\text{-}1^{th}$ time step can be defined using the principle of optimality as,

$$f_{N-1}(X_{N-1}) = X_{N-1}, Q^T W Q X_{N-1} - 2X_{N-1}, Q^T W d_{N-1} + d_{N-1}, W d_{N-1} + g_{n-1}, B g_{n-1} + f_N(X_N) \tag{2.54}$$

Now substituting the optimal cost from the previous time step N and making use of equation (2.33) and (2.51),

$$X_N = M X_{N-1} + P g_{N-1} \tag{2.55}$$

and

$$\begin{aligned} f_{N-1}(X_{N-1}) &= X_{N-1}, Q^T W Q X_{N-1} - 2X_{N-1}, Q^T W d_{N-1} + d_{N-1}, W d_{N-1} + g_{N-1}, B g_{N-1} \\ &+ (M X_{N-1} + P g_{N-1}), Q^T W Q (M X_{N-1} + P g_{N-1}) - \\ &2(M X_{N-1} + P g_{N-1}), Q^T W d_N + d_N, W d_N \end{aligned} \tag{2.56}$$

expanding equation (2.56) with respect to the inner products and rearranging the terms with respect to the powers of $\{X\}$ gives,

$$\begin{aligned} f_{N-1}(X_{N-1}) &= X_{N-1}, [Q^T W Q + M^T Q^T W Q M] X_{N-1} \\ &- 2X_{N-1}, [M^T Q^T W d_N + Q^T W d_{N-1}] \\ &+ d_N, W d_N + d_{N-1}, W d_{N-1} \\ &+ g_{N-1}, [P^T Q^T W Q M X_{N-1} + B] g_{N-1} \\ &+ 2g_{N-1}, [P^T Q^T W Q M X_{N-1} - P^T Q^T W d_N] \end{aligned} \tag{2.57}$$

The optimal cost of the function f_{N-1} is found by minimising the cost with respect to the forcing function for that time step.

$$\frac{\partial f_{N-1}(X)}{\partial g_{N-1}} = g_{N-1}, [P^T Q^T W Q P + B] + P^T Q^T W Q M X_{N-1} - P^T Q^T W d_N = 0 \tag{2.58}$$

The optimal forcing term for the $N\text{-}1^{th}$ time step is now denoted by,

$$g^*_{N-1} = [P^T Q^T W Q P + B]^{-1} [P^T Q^T W d_N - P^T Q^T W Q^T M X_{N-1}] \tag{2.59}$$

Equation (2.59) is then substituted into equation (2.57) and after much rearrangement (Appendix B), can be written in quadratic form in terms of X as,

$$f_{N-1}(X) = X_{N-1},(\beta - \varepsilon^T \phi^{-1} \varepsilon) X_{N-1} + 2X_{N-1}(-\chi + \delta^T \phi^{-1} \varepsilon) + \alpha - \delta^T \phi^{-1} \delta \quad (2.60)$$

where

$$\alpha = d_N, Wd_N + d_{N-1}, Wd_{N-1} \quad (2.61)$$

$$\beta = Q^T WQ + M^T Q^T WQM \quad (2.62)$$

$$\chi = M^T Q^T Wd_N + Q^T Wd_{N-1} \quad (2.63)$$

$$\delta = P^T Q^T Wd_N \quad (2.64)$$

$$\varepsilon = P^T Q^T WQM \quad (2.65)$$

$$\phi = [P^T Q^T WQP + B] \quad (2.66)$$

By induction it can now be assumed that all of the optimal cost functions are quadratics in terms of $\{X\}$ of the form,

$$f_n = (X_n, R_n X_n) + (X_n, S_n) + q_n \quad (2.67)$$

Comparing equations (2.67) and (2.53), and equating like terms of X, gives the initial conditions for the recursive equations as (Trujillo 1978, Trujillo & Busby 1990, 1997, Busby & Trujillo 1986, 1995),

$$R_N = Q^T WQ \quad (2.68)$$

$$S_N = -2Q^T Wd_N \quad (2.69)$$

$$q_N = d_N, Wd_N \quad (2.70)$$

It is now required to find recurrence relations for all of the R_n, S_n and q_n. By taking an arbitrary functional over the *n-1*th time step, the optimal cost can be defined by,

$$f_{n-1} = \min_{g_{n-1}}[(QX_{n-1} - d_{n-1}, W(QX_{n-1} - d_{n-1}) + (g_{n-1}, Bg_{n-1}) + f(X_n)] \quad (2.71)$$

Expanding equation (2.71) with respect to equations (2.65) and (2.33) yields,

$$\begin{aligned}&(X_{n-1}, R_{n-1}X_{n-1}) + (X_{n-1}, S_{n-1}) + q_{n-1} = \\ &\min_{g_{n-1}}[(QX_{n-1} - d_{n-1}, W(QX_{n-1} - d_{n-1}) + (g_{n-1}, Bg_{n-1}) + \\ &(MX_{n-1} + Pg_{n-1}), R_n(MX_{n-1} + Pg_{n-1}) + (MX_{n-1} + Pg_{n-1}), S_n + q_n\end{aligned} \tag{2.72}$$

Expanding equation 2.71 with respect to its inner products gives,

$$\begin{aligned}&(X_{n-1}, R_{n-1}X_{n-1}) + (X_{n-1}, S_{n-1}) + q_{n-1} = \\ &(X_{n-1}, Q^T WQX_{n-1}) - (2X_{n-1}, Q^T Wd_{n-1}) + d_{n-1}, Wd_{n-1} \\ &+g_{n-1}, Bg_{n-1} + (X_{n-1}, M^T R_n MX_{n-1}) + 2(X_{n-1}, M^T R_n Pg_{n-1}) \\ &+(g_{n-1}, P^T R_n g_{n-1}) + X_{n-1}, M^T S_n + g_{n-1}, P^T S_n + q_n\end{aligned} \tag{2.73}$$

In order to calculate the optimal cost, the optimal forcing function must first be calculated by minimising equation (2.73) with respect to *g*, which gives,

$$\frac{\partial f_{n-1}(X)}{\partial g_{n-1}} = 2Bg^*_{n-1} + P^T S_n + 2P^T R_n(MX_{n-1} + Pg^*_{n-1}) = 0 \tag{2.74}$$

Solving for the optimal force g^* for the $n\text{-}1^{th}$ time step gives,

$$g^*_{n-1} = [2B + 2P^T R_n P]^{-1}\{-P^T S_n - 2P^T R_n MX_{n-1}\} \tag{2.75}$$

It is necessary at this point to make two substitutions for ease of derivation let,

$$D_n = [2B + 2P^T R_n P]^{-1} \tag{2.76}$$

$$H_n = 2P^T R_n \tag{2.77}$$

Substituting equations (2.65) and (2.77) into (2.75) defines the optimal forces as,

$$g^*_{n-1} = -D_n P^T S_n - D_n H_n MX_{n-1} \tag{2.78}$$

Equation (2.78) is now substituted into equation (2.73), which gives,

$$
\begin{aligned}
&(X_{n-1}, R_{n-1}X_{n-1}) + (X_{n-1}, S_{n-1}) + q_{n-1} = \\
&(X_{n-1}, Q^T WQX_{n-1}) - (2X_{n-1}, Q^T Wd_{n-1}) + d_{n-1}, Wd_{n-1} \\
&+(-D_n P^T S_n - D_n H_n MX_{n-1}), B(-D_n P^T S_n - D_n H_n MX_{n-1}) + \\
&(X_{n-1}, M^T R_n MX_{n-1}) + 2(X_{n-1}, M^T R_n P(-D_n P^T S_n - D_n H_n MX_{n-1})) \\
&+((-D_n P^T S_n - D_n H_n MX_{n-1}), P^T R_n(-D_n P^T S_n - D_n H_n MX_{n-1})) \\
&+X_{n-1}, M^T S_n + (-D_n P^T S_n - D_n H_n MX_{n-1}), P^T S_n + q_n
\end{aligned}
\tag{2.79}
$$

The right hand side of equation (2.79) is fully expanded in Appendix (C), and like powers of X_{n-1} equated. This leads to what are commonly referred to as the discrete matrix Riccatti equations defined by (Trujillo 1975, Trujillo and Busby 1991,1997),

$$R_{n-1} = Q^T WQ + M^T (R_n - H_n^T D_n H_n / 2) \tag{2.80}$$

$$S_{n-1} = -2Q^T Wd_{n-1} + M^T (I - H_n^T D_n P^T) S_n \tag{2.81}$$

It should also be noted that as q_{n-1} is not required to calculate the optimal force at each time step, it is not required to calculate it over any of the time steps. There are now two steps involved in the inverse dynamic programming procedure, a backward and forward sweep.

The Backward sweep

(1) Initial conditions

(a) $R_N = Q^T WQ$

(b) $S_N = -2Q^T Wd_N$

(c) $H_N = 2(P^T R_N)$

(d) $D_N = (2B + 2P^T R_N P)^{-1}$

(2) Starting at $j = N-1$ and proceeding to $j = 1$, calculate in the following order,

$$
\begin{aligned}
R_{n-1} &= (Q^T WQ + M^T (R_n - H_n^T D_n H_n / 2) M) \\
H_{n-1} &= (2P^T R_{n-1}) \\
D_{n-1} &= (2B + 2P^T R_{n-1} P)^{-1} \\
S_{n-1} &= (-2Q^T Wd_{n-1} + M^T (I - H_n^T D_n P^T) S_n)
\end{aligned}
$$

(3) During the backward sweep save all of the (R_n, H_n, D_n and S_n) as they are required in the forward sweep.

The Forward sweep

(1) Initial conditions

$$X_0 = \{0\}$$
$$g_1 = -(2B + 2(P^T R_2 P))^{-1} P^T (S_2 + 2R_2 M X_1)$$

(2) Starting at $j = 2$ and proceeding to $j = N\text{-}1$ as it has been assumed that $g_N = 0$, calculate the g_j terms from equation (2.78),

$$g_j = -(D_{j+1} P^T S_{j+1}) - (D_{j+1} H_{j+1}) M X_j$$
$$X_{j+1} = M X_j + P g_j$$

This completes the full set of operation for the dynamic programming routine.

2.6 Inverse Dynamics of a Cantilever

To illustrate its application, the method of inverse dynamics is applied to a cantilevered beam as described by (Busby & Trujillo 1986, 1995, Trujillo & Busby 1997). A finite element model of the beam is built in MATLAB and this model is then subjected to various types of forcing functions, all of which are applied at the tip of the cantilever. The inverse dynamics procedure outlined in sections (2.3) and (2.4) is commonly referred to as the zeroth order regularisation, (Trujillo and Busby 1997). In this method, it is the force itself that is smoothed for over time.

2.6.1 Model

The cantilever beam is 1m long, with a cross section of .000832 m^2. The Young's modulus is 72 x 10^9 N/m^2, second moment of area (11717 x 10^{-12} m^4) and density 2800

kg/m^3. A finite element model of the cantilever was generated in MATLAB. The cantilever is divided into 10 equally spaced elements with eleven nodes and two degrees of freedom per node (see figure 2.2), to allow for a translation and a rotation at each node. The elemental stiffness and mass matrices for a beam element with two degrees of freedom per node can be approximated by (see Appendix D, Bathe 1996, Logan 2002, Reddy 1993, Smith & Griffiths 2004, Przemineiecki 1968):

$$K_e = EI/L^3 \begin{pmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{pmatrix} \tag{2.82}$$

$$M_e = \rho AL/420 \begin{pmatrix} 156 & 22L & 54 & -13L \\ 22L & 4L^2 & 13L & -3L^2 \\ 54 & 13L & 156 & -22L \\ -13L & -3L^2 & -22L & 4L^2 \end{pmatrix} \tag{2.83}$$

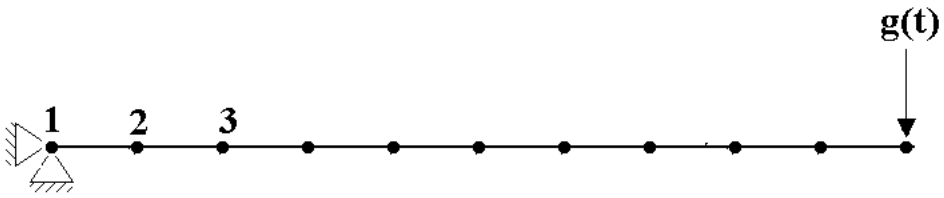

Figure 2.2- Finite element model of a cantilever subject to a dynamic force at the tip

The model was validated using two simple checks, a static deflection and validation of the first natural frequency with the theoretical natural frequency using programmes developed in Matlab. Dynamic simulations are carried out using a Newmark beta direct integration scheme (see Clough & Penzin 1993, Craig 1981, Bathe 1996 and Appendix

E) using another program developed in Matlab. The final dynamic model assumes zero damping so the equilibrium equation of motion reduces to,

$$\left[M_g\right]\{a\}+\left[K_g\right]\{u\}=\left[L\right]\{g(t)\} \tag{2.84}$$

For the cantilever model with the boundary conditions applied, $[M_g]$ and $[K_g]$ are square symmetric matrices of size [20 x 20] and the location matrix $[L]$ is a [20x1] vector with the 19th entry equal to one as the force is acting on the tip.

2.6.2 Zeroth-Order Regularisation for an Impulse Load

The procedure outlined in sections (2.3) and (2.4) is what is known as the zeroth order regularisation. In this method it is the force itself that is regularised. The dynamic model is first converted into a state space equation as defined in equation (2.33). For the cantilever model, the sampling rate (h) is 5 x 10^{-4} s and as zero damping is assumed; the state space equations are defined by,

$$A=\begin{pmatrix} 0 & I \\ -[M_g]^{-1}[K_g] & 0 \end{pmatrix} \tag{2.85}$$

$$M=\exp(Ah) \tag{2.86}$$

$$P=[M-I][A]^{-1}\begin{bmatrix} 0 \\ [M_g]^{-1}[L] \end{bmatrix} \tag{2.87}$$

The cantilever is subjected to an impulse load of –20 N for .05 seconds. Figure 2.3 shows the simulated response of the tip of the cantilever over a period of .15s, it can be seen that once the impulse force has terminated the cantilever is in free vibration. The simulated strain at .05 m from the left hand side of the cantilever is used as the point of measurement for the inverse analysis. The strain within an elementary beam can be

obtained from the displacements and rotations at their end nodes. For an elementary beam defined by end nodes of longitudinal coordinates x_i and x_{i+1}, and displacements y_i, θ_i, y_{i+1} and θ_{i+1}, the strain at any distance x from the left hand side of the beam is given by, (see Rocky et al 1975, Przemineiecki 1968 and Appendix D)

$$\varepsilon(x) = \frac{-y}{l_e^3}\left[(6l_e - 12(x_e))u_i + (4l_e^2 - 6l_e(x_e))\theta_i + (-6l_e + 12(x_e))u_{i+1} + (2l_e^2 - 6l_e(x_e))\theta_{i+1}\right] \tag{2.88}$$

The simulated strain is then contaminated with 5% Gaussian noise as a percentage of the maximum strain (see figure 2.4) given by,

$$\varepsilon = \varepsilon_{calc} + E_p \varepsilon_{\max} N_{oise} \tag{2.89}$$

where ε and ε_{calc} are the contaminated and calculated strains respectively, E_p is the noise level. N_{oise} is a standard normally disturbed vector of random numbers with zero mean and a standard deviation of one, $\varepsilon_{\max}$ is the maximum calculated strain. The strain at a location x of an elementary beam element can be defined in vector form by,

$$\varepsilon_x = \frac{-y}{l_e^3}\begin{Bmatrix} 6l_e - 12(x_e) \\ 4l_e^2 - 6l_e(x_e) \\ -6l_e + 12(x_e) \\ 2l_e^2 - 6l_e(x_e) \end{Bmatrix}\begin{Bmatrix} u_1 & \theta_1 & u_2 & \theta_2 \end{Bmatrix} \tag{2.90}$$

As the measurements are taken at .05 m from the left hand side of the cantilever, both the displacement and rotation at the first node are zero. Now substituting $l_e = .1$, $x_e = .05$ and $y = .032$m into equation (2.90) gives,

$$\varepsilon_x = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ -.32 \end{Bmatrix}\begin{Bmatrix} u_1 & \theta_1 & u_2 & \theta_2 \end{Bmatrix} \tag{2.91}$$

Figure 2.4 shows the simulated strain at .05 m from the left hand side, and the corrupted strain used as the input to the inverse dynamics problem. Equation (2.91) is the relationship for the Q matrix as defined in equation (2.43). In this example Q relates the strain to the state variables. Hence Q is an array [1 x 40] and the second entry is equal to -.32 with the rest being all zeros:

$$Q = [0, -.32, 0.....0] \tag{2.92}$$

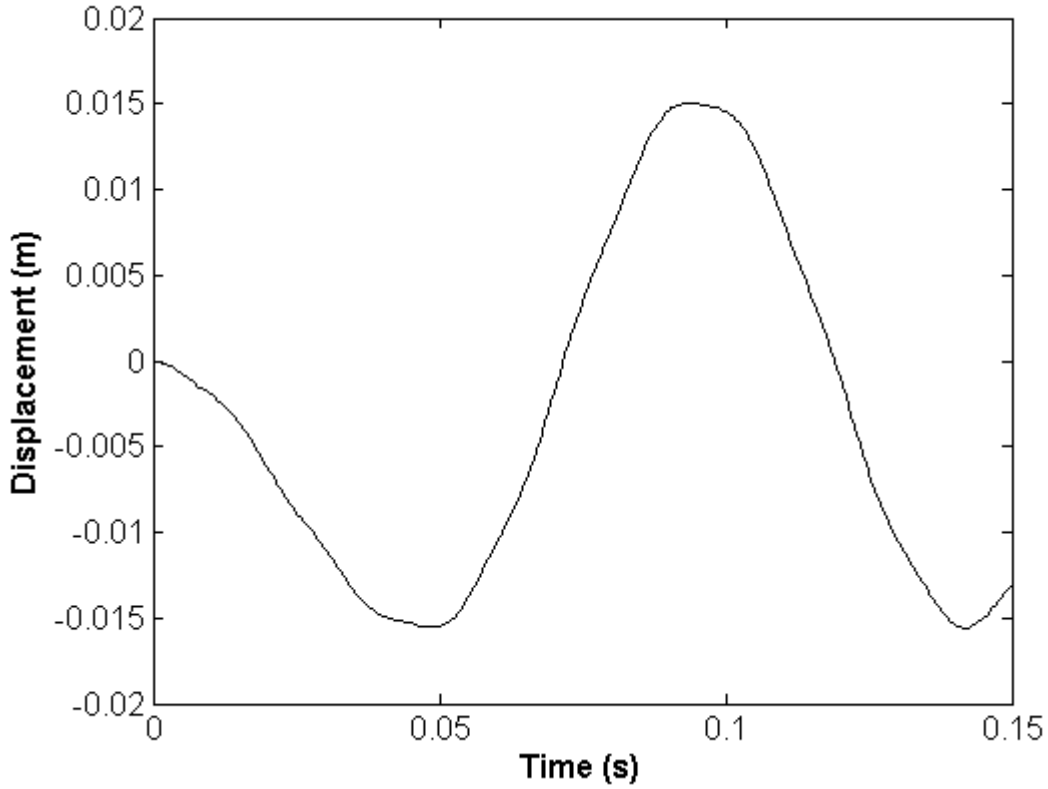

Figure 2.3 – Tip displacement of a cantilever due to an impulse load

In the formulation of the inverse dynamics problem as described in section 2.3 the residual norm is defined as,

$$E_{norm} = (\sum_{j=1}^{N}((Qx_j - d_j), W(Qx_j - d_j)))^{1/2} \tag{2.93}$$

and the solution norm is defined as,

$$F_{norm} = (\sum_{j=1}^{N}(g_j, g_j))^{1/2} \tag{2.94}$$

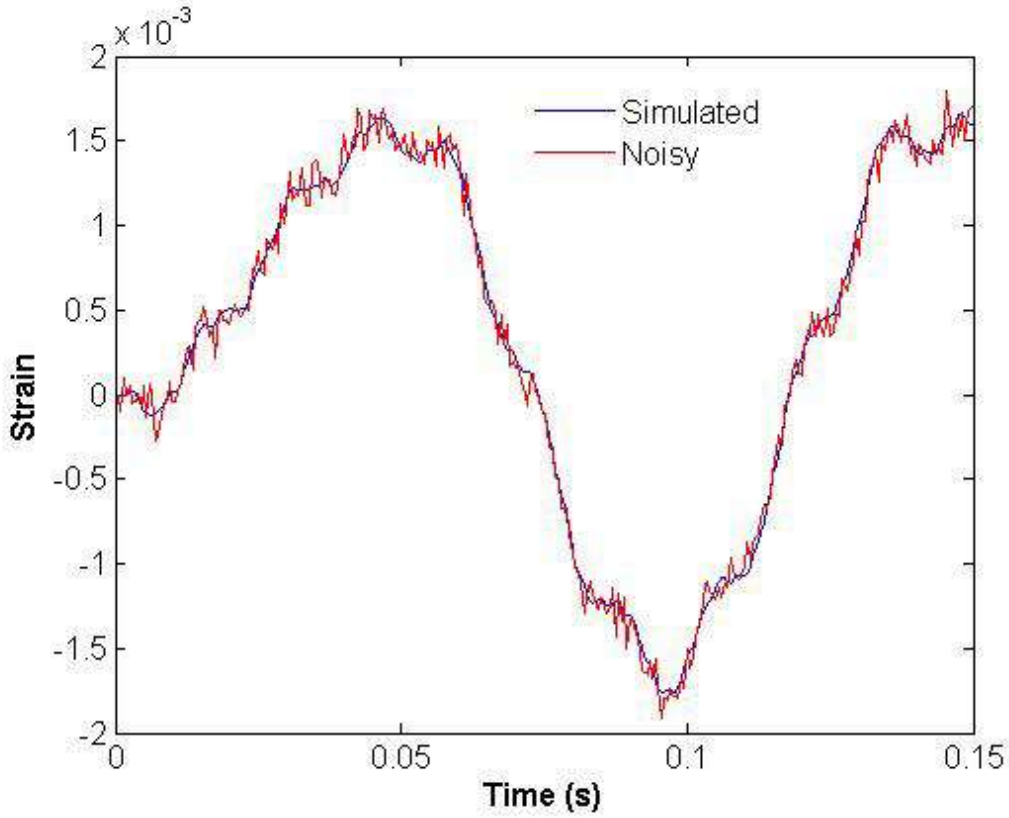

Figure 2.4 – Simulated and noisy strain at .05m from left hand side

The optimal regularisation parameter is calculated from the L-curve. The error and solution norms are calculated over a range of regularisation parameters. In this example λ is varied from 1×10^{-19} to 1×10^{-5} in increments of 1×10^{-1}. From the L-curve the optimal regularisation parameter is between 1×10^{-11} and 1×10^{-10} (see figure 2.5). The smoothed solutions using 1×10^{-11} and 1×10^{-10} as the optimal regularisation parameters are shown in figures 2.6 and 2.7. It can be seen from the regularised solutions, that the solution is very sensitive to the regularisation parameter. In figure 2.7 the regularisation parameter λ is slightly closer to the least squares solution, even though there is considerable variability in this particular solution, the trend is there and it follows the correct pattern. This shows the necessity of adding the regularisation parameter in order to obtain accurate solutions. Figure 2.6 shows the regularised solution for a slighter larger regularisation parameter and the smoother solution is clearly visible, the aforementioned tradeoff between the residual norm and the solution norm is evident from the comparison of these two regularised solutions.

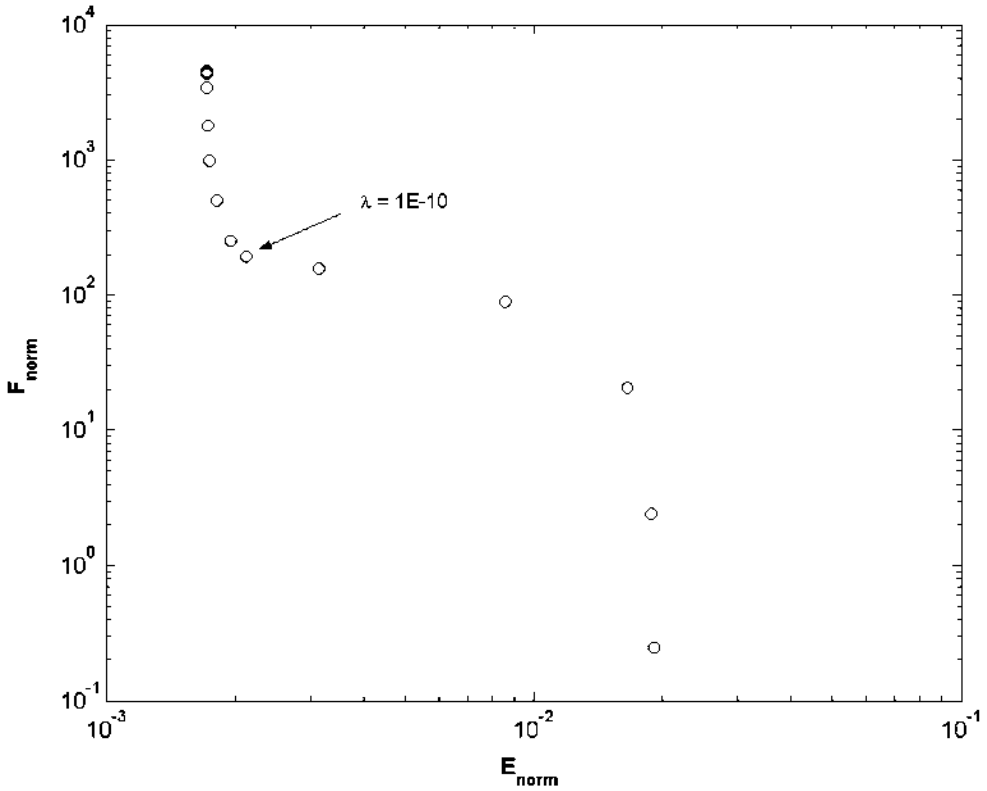

Figure 2.5 – L-curve for zeroth order regularisation of an impulse load

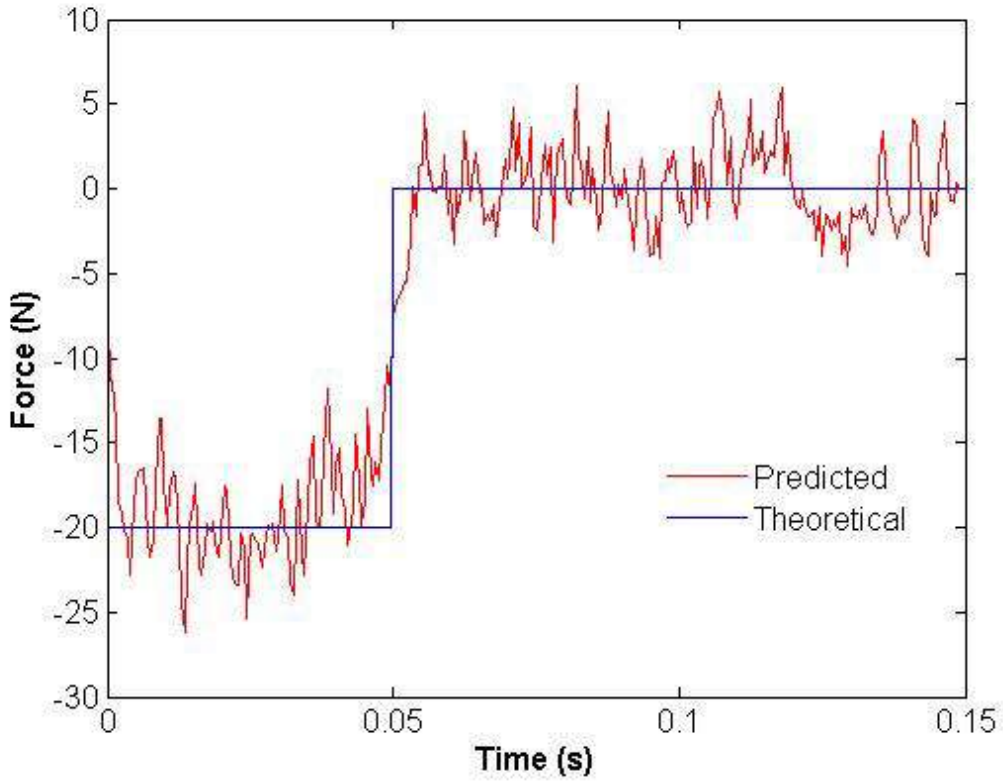

Figure 2.6 – Zeroth Order regularisation, $\lambda = 1 \times 10^{-10}$

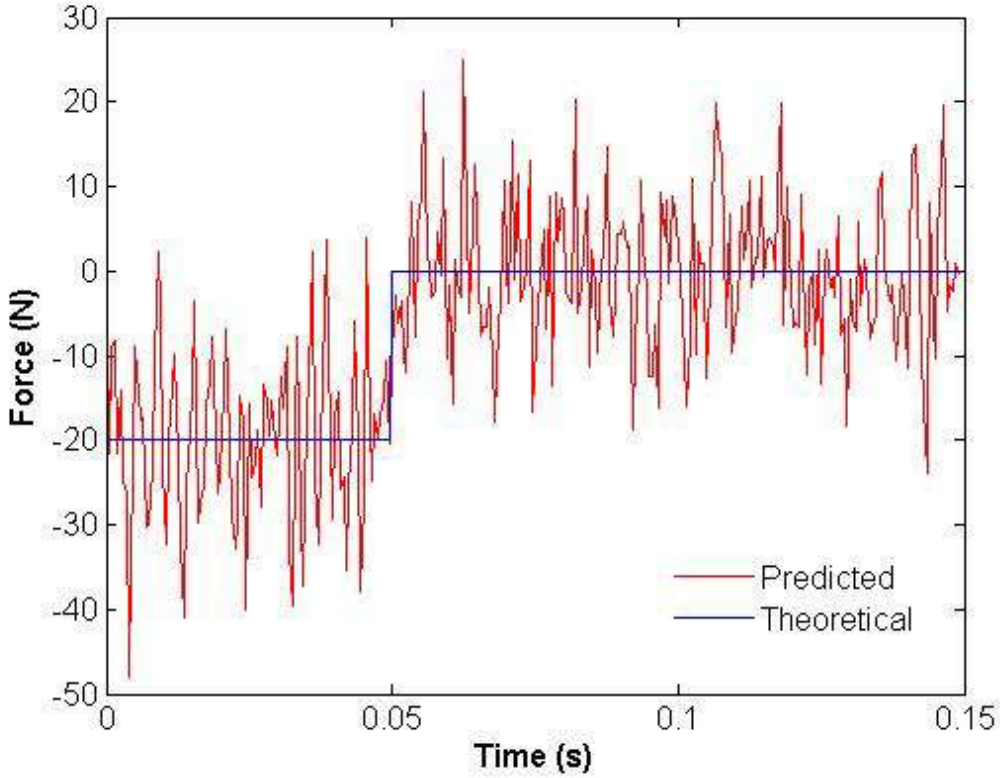

Figure 2.7 – Zeroth order regularisation, $\lambda = 1 \times 10^{-11}$

2.6.3 First Order Regularisation for a Triangular and Sinusoidal Load

Previous studies have shown that in the area of inverse dynamics the first order regularisation performs significantly better than that of the zeroth order (Trujillo & Busby 1997). In the first order regularisation it is the derivative of the force that is regularised for, if the force at each time step is represented as the sum of the force at the previous time step and its unknown derivative:

$$g_{j+1} = g_j + r_j \tag{2.95}$$

The force vector can then be included in the state vector and the state space equations can be represented by,

$$\begin{Bmatrix} X_{j+1} \\ g_{j+1} \end{Bmatrix} = \begin{bmatrix} M & P \\ 0 & I \end{bmatrix} \begin{Bmatrix} X_j \\ g_j \end{Bmatrix} + \begin{Bmatrix} 0 \\ r_j \end{Bmatrix} \tag{2.96}$$

With the inclusion of the forcing vector in the state variable the problem is reformulated as,

$$E = \sum_{j=1}^{N} (QX_j - d_j, W(QX_j - d_j)) + (r_j, Br_j) \tag{2.97}$$

and the dynamic programming algorithm can be applied in the same way as described in section 2.5. In this case as the force vector is also included in the state variable the selector matrix [Q] is now a [1 x 81] vector with the second entry equal to -.32 and the rest zeros. For the L-curve analysis, the solution norm will be that of the derivative of the force for each value of the regularisation parameter, so for the first order regularisation, the norms that are plotted for each value of λ on the L-curve are,

$$E_{norm} = (\sum_{j=1}^{N} ((Qx_j - d_j), W(Qx_j - d_j)))^{1/2} \tag{2.98}$$

$$F_{norm} = (\sum_{j=1}^{N} (r_j, r_j))^{1/2} \tag{2.99}$$

A triangular forcing function of the form,

$$g(t) = 5 - 100(t), 0 \le t \le .05 \tag{2.100}$$

is applied at the tip of the cantilever. The system is then simulated for .15 seconds and the measurements were taken in the first element, .05 m from the support. The theoretical strains are then contaminated with 5% Gaussian noise as defined in equation (2.89). In order to calculate the optimal regularisation parameter the L-curve as described in section 2.3 is used. If the regularisation parameter is varied by orders of magnitude initially this will give a rough idea of where the optimal regularisation parameter lies. In the case of the triangular forcing function applied at the tip, the initial estimate of where the regularisation parameter lies was between 1×10^{-16} and 1×10^{-1}. The dynamic programming routine was carried out on all values of λ between these two extremities in increments of 1×10^{-1} (see figure 2.8). The L-curve was then further iterated between 1×10^{-9} and 1×10^{-8} in increments of $.1 \times 10^{-1}$ as it was assumed that the optimal regularisation parameter lay in this region (see figure 2.8). The refinement

of the L-curve is illustrated in figure 2.9. From figure 2.9 it was deduced that the optimal value of λ is approximately 5×10^{-9}. The forward and backward sweeps were then redone using the optimal regularisation parameter (see figures 2.10 to 2.12)

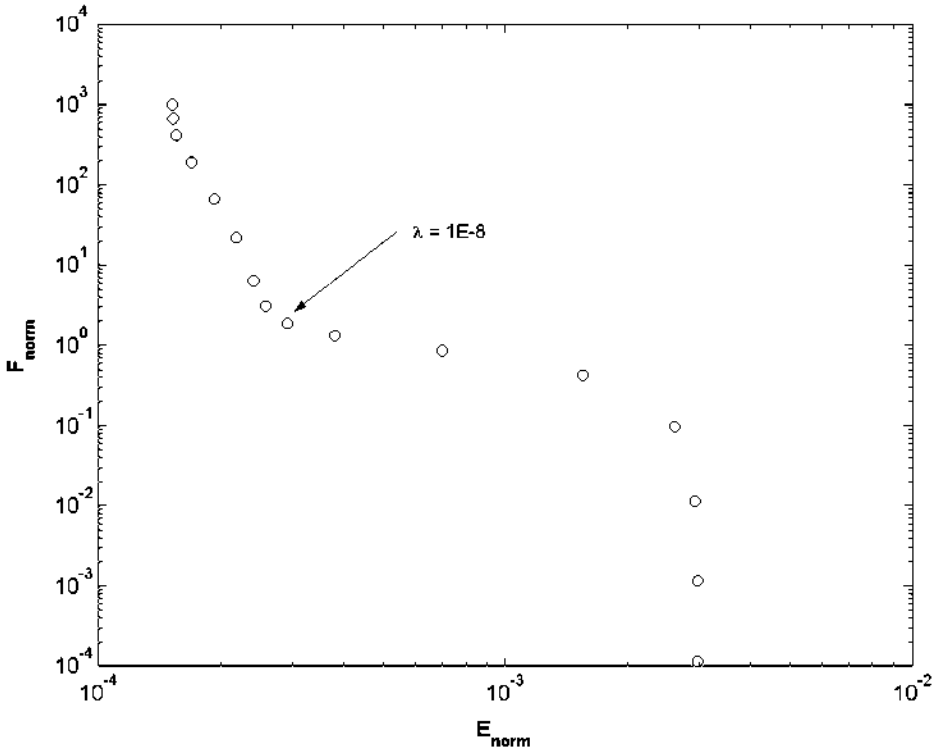

Figure 2.8 - Initial L-curve for first order regularisation of a triangular load

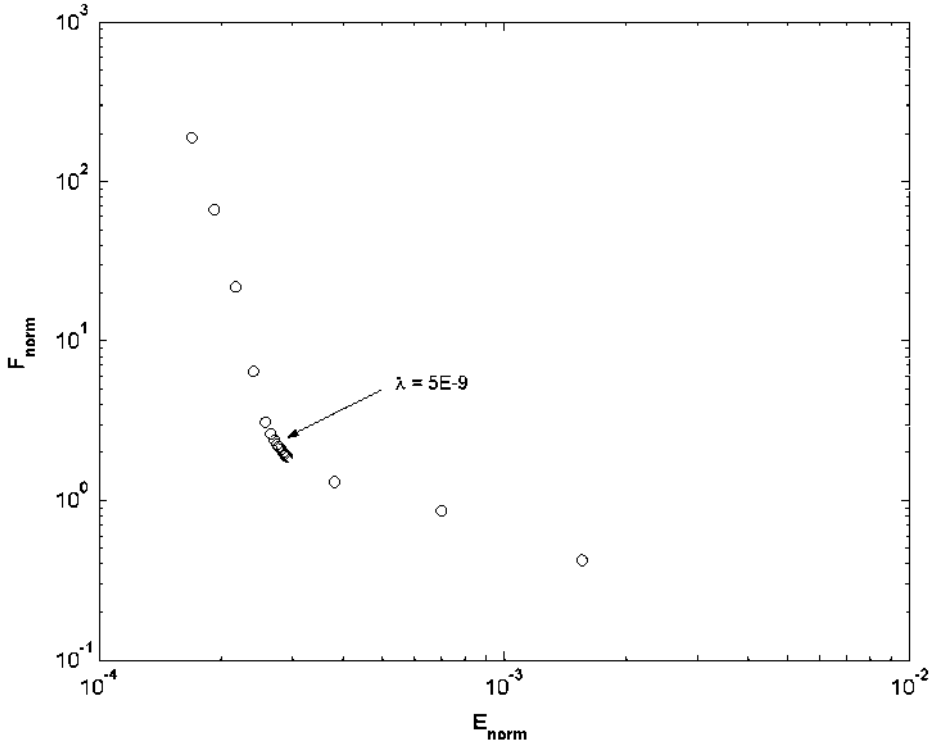

Figure 2.9 – Refined L-curve for first-order regularisation for a triangular load

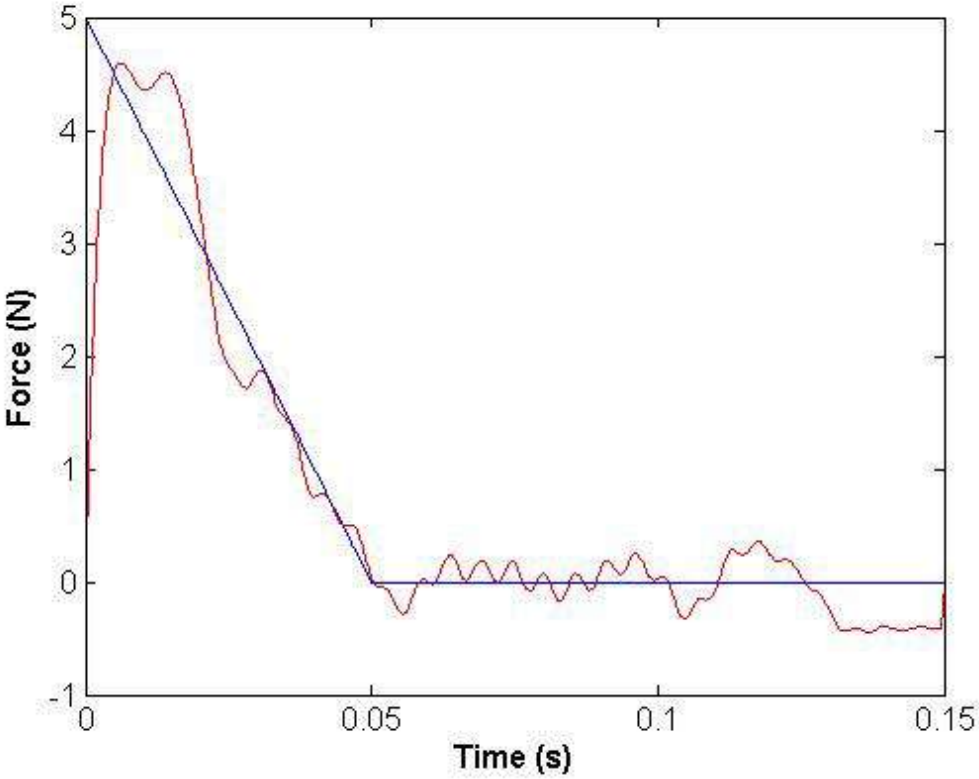

Figure 2.10 – First order regularisation with $\lambda = 1 \times 10^{-8}$

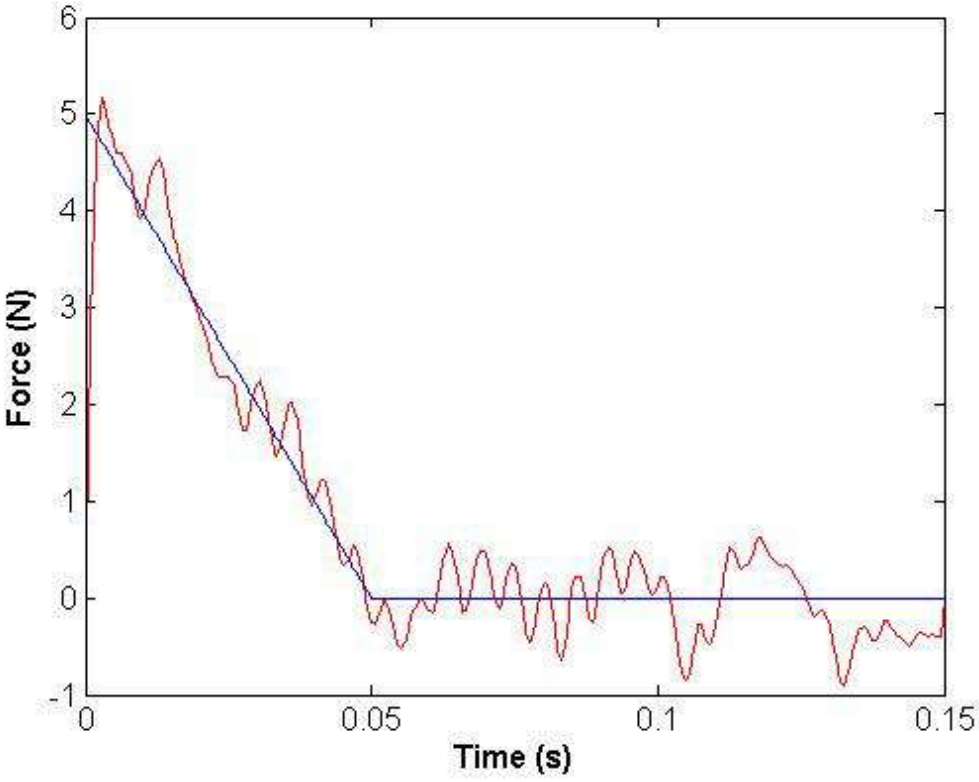

Figure 2.11 – First order regularisation with $\lambda = 1 \times 10^{-9}$

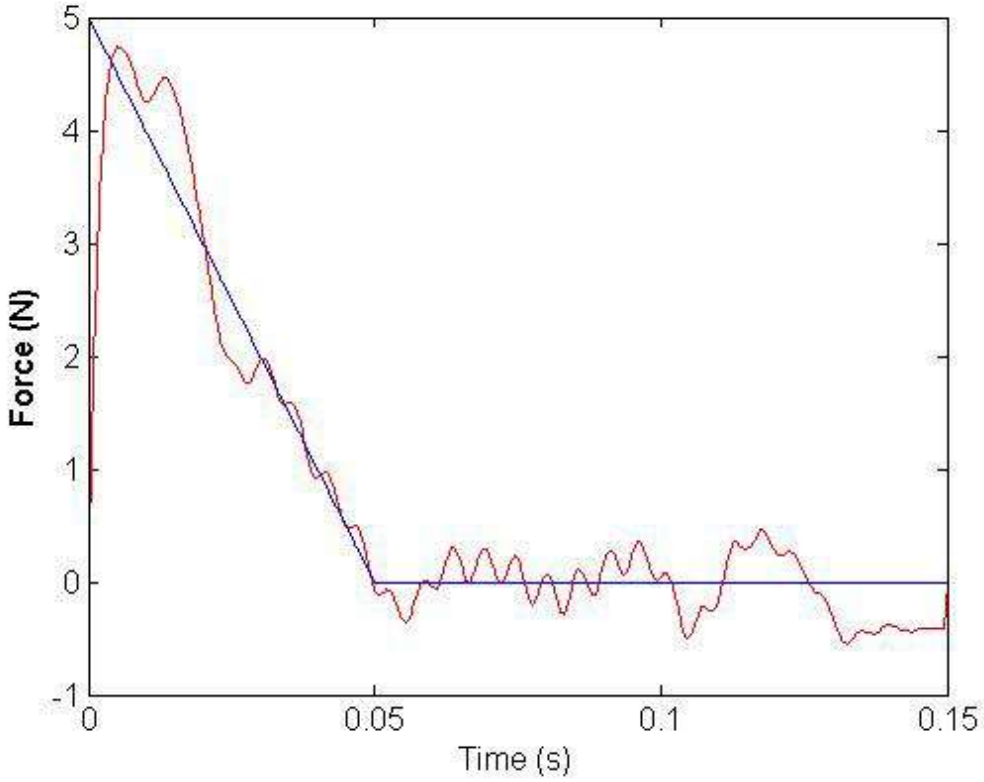

Figure 2.12 – First order regularisation for $\lambda = 5 \times 10^{-9}$

Figures 2.11 to 2.12 illustrate the effectiveness of the first order regularisation over that of the zeroth, the regularised solutions over the range of regularisation parameters at the corner of the L-curve all give acceptable results. These figures also show the stabilising nature of the first order regularisation, in all of the solutions shown the error in identified forces is visible smoother and closer to that of the actual applied force.

Finally the same procedure was applied for sinusoidal force acting at the tip of the cantilever. The sinusoidal load is applied for one full period of two seconds; the system was simulated for this period and one second of free vibration. The theoretical strains with respect to time are again contaminated with 5% Gaussian noise. As with the triangular load, the dynamic programming routine was run for various values of λ to calculate the optimal regularisation parameter, in this instance the routine was run for lambda between 1×10^{-13} and 1×10^{-4} in increments of $.1 \times 10^{-1}$. From the L-curve it was found that 1×10^{-7} is approximately the point of maximum curvature and thus the optimal regularisation parameter (see figure 2.13). Using this value of lambda the backward and forward sweeps were carried out and the optimal forcing terms were

calculated. Figure 2.15 shows the predicted forcing function in red, plotted with the theoretical force in blue, it should be noted that the regularised solution is almost indistinguishable form the theoretical applied force. The routine was rerun for a smaller value of lambda 1×10^{-8} (see figure 2.16), which lay just off the point of maximum curvature. Both values of lambda give excellent results particularly by comparison with the zeroth order regularisation.

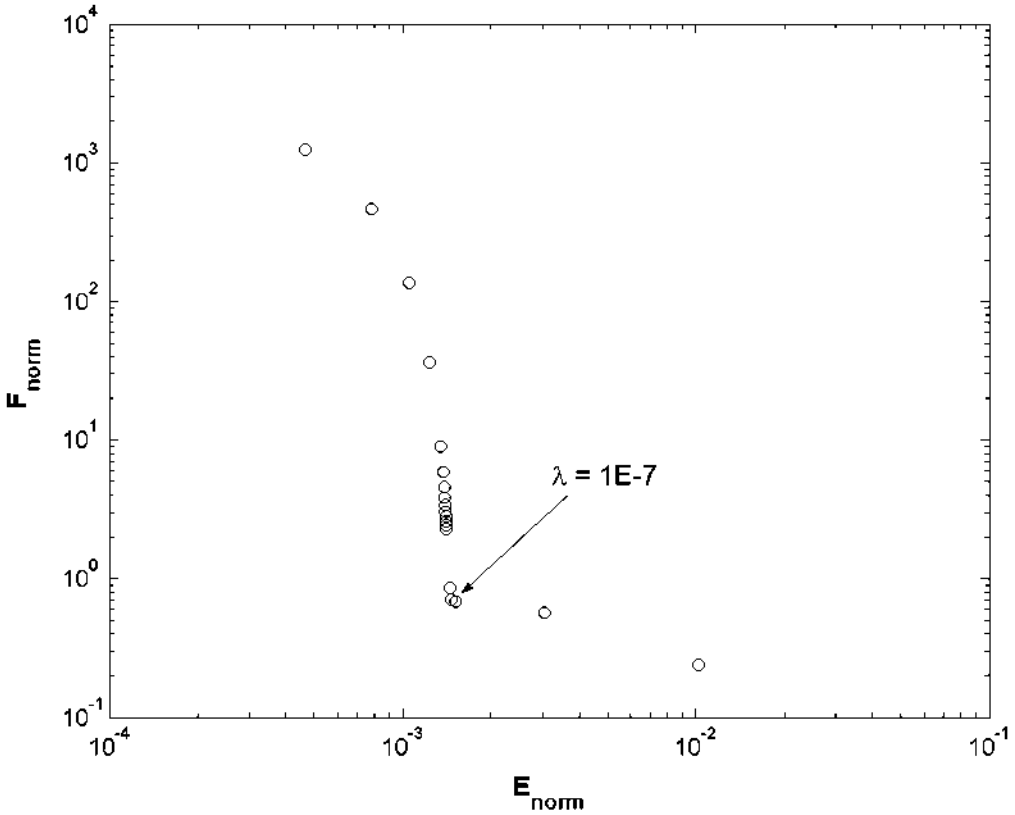

Figure 2.13 – L-curve for first order regularisation of a sinusoidal load

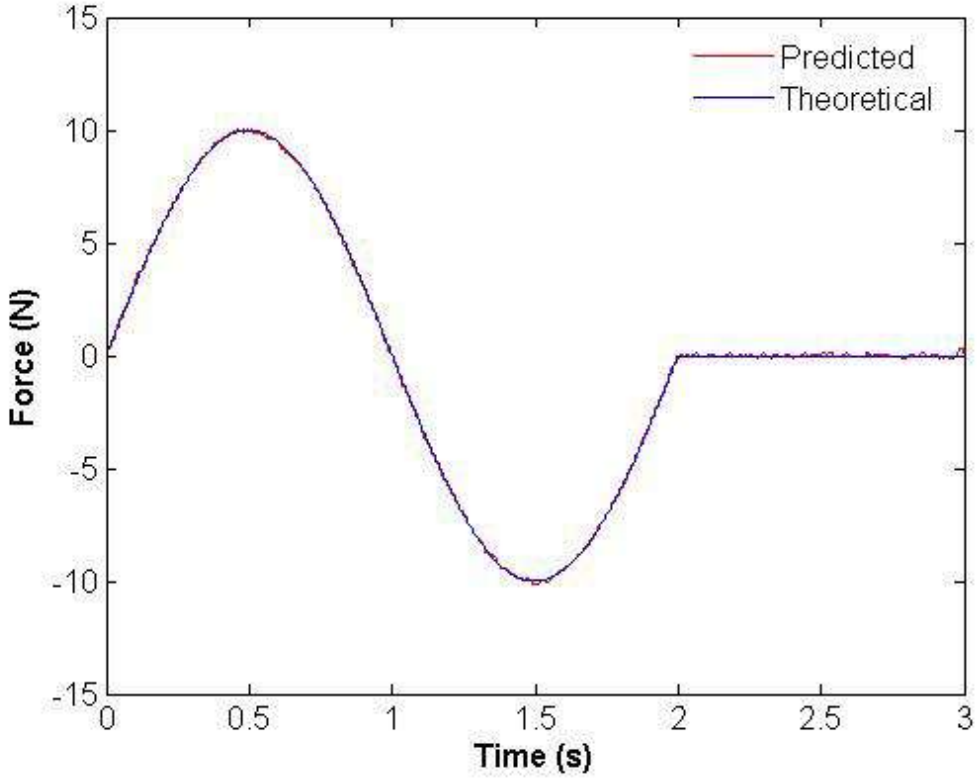

Figure 2.14 – Optimal first order regularisation of a sinusoidal load

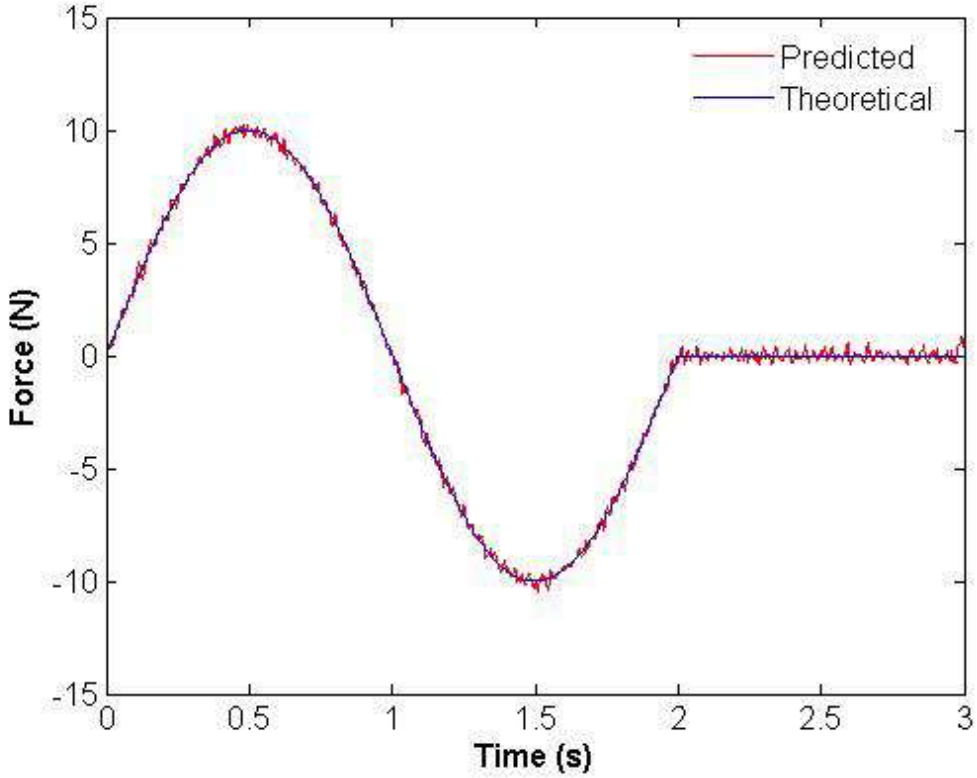

Figure 2.15 - First order regularisation of a sinusoidal load, $\lambda = 1 \times 10^{-8}$

2.7 Conclusions

The method of force identification using general inverse theory is discussed, and the mathematical theory behind the method has been discussed separately in each section. The conversion of the equilibrium equation of motion to a discrete time integration scheme has been discussed and the theory behind it explained. The mathematical tool of regularisation has been explained both quantively and qualitatively, the method has been explained in a general linear algebraic setting such that it can be implemented with ease in the following chapters. The powerful optimisation scheme of dynamic programming has been comprehensively explained, and its application to the inverse dynamics problem detailed. Finally the inverse dynamics approach using the method of general inverse theory has been validated using numerical experiments on a cantilevered beam. The example illustrates the ill-conditioning of the problem, and the need to have a regularisation term in the formulation of the problem. Further to this the examples show how the first order regularisation out performs the zeroth order regularisation. The theory outlined in this chapter forms a basis for the algorithms developed throughout this thesis.