

A Novel Method for the Link Mechanism Inverse Problem

by

David M. Trujillo
TRUCOMP
Fountain Valley, CA

March, 2006

Abstract

The inverse link mechanism problem is cast as a discrete optimal control problem and solved using dynamic programming and an L curve analysis. A natural coordinate model was chosen to represent the link mechanisms together with a stiff spring to approximate a rigid link. A numerical example of a two link model is included to demonstrate the performance of the methodology.

Keywords: Inverse problem; Link mechanisms; Optimal Control; Dynamic programming; L curve.

Please note: This document is © by David M. Trujillo. This manuscript may be freely duplicated and distributed so long as no consideration is received in return, and this copyright notice remains intact.

trucomp@earthlink.net

1.0 Introduction

The inverse dynamics problem consists of estimating externally applied forces and moments from measurements. It is presumed that the measurements will have some inaccuracies (noise). One successful approach to inverse problems is to cast it as an optimal control problem and solve it using dynamic programming. This approach has been able to solve inverse heat conduction and structural dynamics problems (Trujillo and Busby, 1997; Hollandsworth and Busby, 1989; Tanaka and Chen, 2000, Busby, 1997). These models were mostly linear or slightly nonlinear, such as temperature varying properties in the heat conduction problem. This paper explores the use of this approach to the highly nonlinear models that represent link mechanisms.

This approach to the inverse dynamic link mechanism problem also differs from other approaches (Risher, Schutte, and Runge, 1997) in that only displacement measurements are required. It does not require estimates of the velocities and accelerations. These are accounted for naturally in the solution. In addition, the dynamic model that is used for direct simulation is also used in the solution of the inverse problem. The approach is very similar to that of Runge et al. (1995) except that the nonlinear solution is obtained. This is accomplished with an iterative solution about the displacement time histories.

The paper will describe the model chosen to represent the link mechanisms and the trapezoidal method used to integrate the equations of motion which are then used to convert the model to a discrete state variable form. The discrete inverse problem is then

formulated and the solution is outlined. Finally, the efficaciousness of the overall approach is demonstrated with a numerical simulated two link mechanism driven with two time varying torques and noisy measurements.

2.0 Model Description

There are many ways to represent the equations of motion for link mechanisms. The method chosen in this paper is one using natural coordinates (Garcia de Janlon and Bayo, 1994). This method has several advantages when converting the equations to discrete form, mainly that the mass matrix is constant and the nonlinear terms are dependent only on the displacements. This type of model more easily fits into the discrete optimal control method used to solve the inverse problem. There is one disadvantage in that a stiff axial spring is used as a constraint to approximate a rigid link. This spring introduces spurious oscillations into the model and the inverse estimates. However, numerical experiments have shown that these oscillations were small and did not seriously affect the estimates.

The model for a link in a plane with a mass m , inertia I , length L_0 can be derived by first expressing the coordinates of the ends a and b in terms of the center of mass c_x and c_y and the angle θ . Figure 1 shows the properties of the link including lengths L_a and L_b and the coordinates x and y . These coordinate expressions for the ends of the link are

$$x_a = c_x - L_a \cos(\theta)$$

$$y_a = c_y - L_a \sin(\theta) \quad (1)$$

$$x_b = c_x + L_b \cos(\theta)$$

$$y_b = c_y + L_b \sin(\theta)$$

In addition to the usual rigid kinetic and potential energies for a link, a stiff axial spring k is connected between points a and b . Using Equations (1) to express the kinetic and potential energies in terms of the endpoint coordinates and velocities and Lagrange's equations, the equations of motion for a link become (see Appendix A)

$$(\mathbf{M} + \mathbf{I}) \frac{d^2 \mathbf{z}}{dt^2} + \mathbf{f} + \mathbf{g} = \mathbf{T} \quad (2)$$

where the displacement state vector \mathbf{z} is

$$\mathbf{z} = \begin{bmatrix} x_a \\ y_a \\ x_b \\ y_b \end{bmatrix} \quad (3)$$

The mass matrix \mathbf{M} is

$$\mathbf{M} = m \begin{bmatrix} b^2 & 0 & ab & 0 \\ 0 & b^2 & 0 & ab \\ ab & 0 & a^2 & 0 \\ 0 & ab & 0 & a^2 \end{bmatrix} \quad (4)$$

where $a = L_a/L_0$ and $b = L_b/L_0$.

The inertia matrix \mathbf{I} is

$$\mathbf{I} = (I/L_o^2) \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \quad (5)$$

The vector \mathbf{f} represents the nonlinear forces due to the spring.

$$\mathbf{f} = k(L-L_o)/L_o \begin{bmatrix} (x_a - x_b) \\ (y_a - y_b) \\ (x_b - x_a) \\ (y_b - y_a) \end{bmatrix} \quad (6)$$

where

$$L^2 = (x_b - x_a)^2 + (y_b - y_a)^2 \quad (7)$$

and L_o is the unstretched length of the spring.

The gravity force vector is

$$\mathbf{g} = \begin{bmatrix} 0 \\ mgL_b/L \\ 0 \\ mgL_a/L \end{bmatrix} \quad (8)$$

For an applied torque $T(t)$ to a link, the torque vector \mathbf{T} is

$$\mathbf{T} = (T(t)/L_o^2) \begin{bmatrix} (y_b - y_a) \\ -(x_b - x_a) \\ -(y_b - y_a) \\ (x_b - x_a) \end{bmatrix} \quad (9)$$

These equations for a link are very similar to a finite element representation. This is another advantage of this model in that the global equations for a system of links are

assembled just like a system of finite elements. This assembled model will be referred to as the FE link model. Isobe 2001 has also proposed a finite element method to solve the inverse dynamics problem for open and closed loop link mechanisms. That method differs from the one used here in that it uses a shifted integration technique with a beam element to model the link mechanism.

3.0 Discrete Model Representation

There are many methods that can be used to numerically integrate Equation (2). However for the inverse problem solution investigated here, single step methods are preferred. Of all the single step methods, the trapezoidal method offers the most stability, accuracy, and simplicity for nonlinear models (Klein and Trujillo 1983). Garcia de Jalon and Bayo 1994 have an excellent discussion on various integration methods including the trapezoidal method.

Defining the velocity state vector as

$$\mathbf{v} = d\mathbf{z}/dt \quad (10)$$

the trapezoidal formula for time step i to $i+1$ with a step size h is

$$(\mathbf{z}_{i+1} - \mathbf{z}_i)/h = \frac{1}{2} (\mathbf{v}_{i+1} + \mathbf{v}_i) \quad (11)$$

$$(\mathbf{M} + \mathbf{I}) (\mathbf{v}_{i+1} - \mathbf{v}_i)/h + \frac{1}{2}(\mathbf{f}_{i+1} + \mathbf{f}_i) = \mathbf{T}_i - \mathbf{g}_i \quad (12)$$

One advantage of the above dynamic link model is that the nonlinear forces \mathbf{f} are a function only of the displacements \mathbf{z} and not the velocities. With this in mind, the

trapezoidal formula can be rearranged for the unknowns \mathbf{v}_{i+1} and \mathbf{z}_{i+1}

$$(\mathbf{M} + \mathbf{I}) \mathbf{v}_{i+1} + h \mathbf{f}_{i+1}/2 = (\mathbf{M} + \mathbf{I}) \mathbf{v}_i - h \mathbf{f}_i/2 + h(\mathbf{T}_i - \mathbf{g}_i) \quad (13)$$

This formula requires solving a set of nonlinear equations for \mathbf{v}_{i+1} and \mathbf{z}_{i+1} at each time step. One method for solving a set of nonlinear equations is Newton's method which requires the Jacobian \mathbf{J} of \mathbf{f} , which is easily derived and assembled for the FE link model, another advantage.

The iterations at each timestep are for the unknowns \mathbf{v}_{i+1} and \mathbf{z}_{i+1} . In order to avoid confusion with the indices denoting time and those for the Newton iterations, let us express Equation (13) in terms of unknowns \mathbf{s} and \mathbf{q} representing \mathbf{v}_{i+1} and \mathbf{z}_{i+1} , respectively. Equations (11) and (13) become

$$\mathbf{q} = \mathbf{z}_i + h (\mathbf{s} + \mathbf{v}_i)/2 \quad (13a)$$

$$(\mathbf{M} + \mathbf{I}) \mathbf{s} + h \mathbf{f}(\mathbf{q})/2 - \mathbf{rhs} = 0 \quad (13b)$$

where \mathbf{rhs} is the right hand side of Equation (13) which involves only known quantities.

The task is to now solve the set of nonlinear equations (13a and b) for the unknowns \mathbf{s} and \mathbf{q} using Newton's iterative method.

The nonlinear term $\mathbf{f}(\mathbf{q})$ is approximated using a Taylor's expansion about the iterates \mathbf{q}_j and \mathbf{q}_{j+1} (note subscript j which denotes the Newton formula iterates).

$$\mathbf{f}(\mathbf{q}) \sim \mathbf{f}(\mathbf{q}_j) + \mathbf{J}(\mathbf{q}_j) (\mathbf{q}_{j+1} - \mathbf{q}_j) \quad (13c)$$

Using Equations (13a) to replace $(\mathbf{q}_{j+1} - \mathbf{q}_j)$, Equation (13b) can be written as

$$(\mathbf{M} + \mathbf{I}) \mathbf{s}_{j+1} + h/2 \mathbf{f}(\mathbf{q}_j) + h^2 \mathbf{J}(\mathbf{q}_j)/4 (\mathbf{s}_{j+1} - \mathbf{s}_j) - \mathbf{rhs} = 0 \quad (13d)$$

which can be rearranged as

$$(\mathbf{M} + \mathbf{I} + h^2\mathbf{J}(\mathbf{q}_j)/4)(\mathbf{s}_{j+1} - \mathbf{s}_j) + (\mathbf{M} + \mathbf{I})\mathbf{s}_j + h/2\mathbf{f}(\mathbf{q}_j) - \mathbf{rhs} = 0 \quad (13e)$$

Define the residual $\mathbf{r}(\mathbf{s}_j)$ as

$$\mathbf{r}(\mathbf{s}_j) = (\mathbf{M} + \mathbf{I})\mathbf{s}_j + h/2\mathbf{f}(\mathbf{q}_j) - \mathbf{rhs} \quad (13f)$$

The final iterative equations become

$$(\mathbf{M} + \mathbf{I} + h^2\mathbf{J}(\mathbf{q}_j)/4)(\mathbf{s}_{j+1} - \mathbf{s}_j) = -\mathbf{r}(\mathbf{s}_j) \quad (13g)$$

$$\mathbf{q}_{j+1} = \mathbf{z}_i + h(\mathbf{s}_{j+1} + \mathbf{v}_i)/2 \quad (13h)$$

The iterations continue until $\mathbf{r}(\mathbf{s}_j)$ is zero or very small, usually less than a convergence criterion.

The Jacobian is derived for the nonlinear force vector

$$\mathbf{f} = k(L-L_0)/L_0 \begin{bmatrix} (x_a - x_b) \\ (y_a - y_b) \\ (x_b - x_a) \\ (y_b - y_a) \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \quad (13i)$$

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f_1}{\partial x_a} & \frac{\partial f_1}{\partial y_a} & \frac{\partial f_1}{\partial x_b} & \frac{\partial f_1}{\partial y_b} \\ \frac{\partial f_2}{\partial x_a} & \frac{\partial f_2}{\partial y_a} & \frac{\partial f_2}{\partial x_b} & \frac{\partial f_2}{\partial y_b} \\ \frac{\partial f_3}{\partial x_a} & \frac{\partial f_3}{\partial y_a} & \frac{\partial f_3}{\partial x_b} & \frac{\partial f_3}{\partial y_b} \\ \frac{\partial f_4}{\partial x_a} & \frac{\partial f_4}{\partial y_a} & \frac{\partial f_4}{\partial x_b} & \frac{\partial f_4}{\partial y_b} \end{bmatrix} \quad (13j)$$

The upper terms of the symmetric matrix \mathbf{J} are (all multiplied by k/L)

$$J(1,1) = (L - L_0) + (x_b - x_a)^2 / L$$

$$J(1,2) = -(y_b - y_a) (x_a - x_b) / L$$

$$J(1,3) = -J(1,1)$$

$$J(1,4) = -J(1,2)$$

$$J(2,2) = (L - L_0) + (y_b - y_a)^2 / L$$

$$J(2,3) = (x_b - x_a) (y_a - y_b) / L$$

$$J(2,4) = -J(2,2)$$

$$J(3,3) = J(1,1)$$

$$J(3,4) = (y_b - y_a) (x_b - x_a) / L$$

$$J(4,4) = J(2,2)$$

where

$$L^2 = (x_b - x_a)^2 + (y_b - y_a)^2 \quad (7)$$

4.0 Validation of the FE Link Model

In order to validate the performance of the FE link model a comparison will be made to a two link problem that is also modeled with generalized coordinates (Ghorbel et al., 1994). The ordinary differential equations presented there will be numerically integrated with a fourth order Runge-Kutta formula. This model will be referred to as the Gen model. The FE link model is numerically integrated using the trapezoidal method (Equations 11 and 13).

The two link model is pinned at one end and starts at rest in the vertical position. It will be excited with two time varying torques. The physical properties for the two links

are

| | | | | |
|--------|---|-----------------------------------|-----------------------|-----------------------|
| Link 1 | mass = 1.0Kg | inertia = 4.E-2Kg-m ² | L _o = 0.7m | L _a =0.35m |
| Link 2 | mass = 0.2Kg | inertia = 1.7E-2Kg-m ² | L _o = 1.0m | L _a =0.50m |
| | gravitational acceleration = 9.806 m/s ² | | | |

The time varying applied torques are

Torque on link 1 = 1.0N-m for 0. < time < 0.5 seconds, and zero elsewhere

Torque on link 2 = -1.0N-m for 1.0 < time < 1.5 seconds, and zero elsewhere

For this comparison the FE link model used a stiff spring equal to 1.0E4N/m to approximate a rigid link. The integration time step for both models was 0.01 seconds.

The angular velocities of each link provide a good basis for comparing the two models. Figures 2 and 3 show the angular velocities of links 1 and 2, respectively for both methods. The agreement is excellent even with the spurious oscillations in the FE link model. Both models were also integrated with a smaller time step of 0.005 seconds and both models showed differences from the 0.01 second solution after about 2 seconds. Figure 4 shows the x and y displacement time histories of the free end of link 1. These large displacements exhibit the highly nonlinear nature of the model

The type of inverse problem that is the subject of this paper is the following:
Suppose one had measured the x and y displacement time histories at the ends of each link during an experiment, would it be possible to obtain good estimates of the applied torque time histories from only these measurements? In the next sections, the general

inverse problem will be stated and applied to the above two link model.

5.0 The General Inverse Problem

The formation of the general discrete inverse problem can be stated in terms of the state variable vector \mathbf{q}_i which represents the displacements and velocities of the model at time i . The discrete model of the system is represented as

$$\mathbf{q}_{i+1} = \mathbf{A}_i \mathbf{q}_i + \mathbf{n}_i + \mathbf{P}_i \mathbf{u}_i \quad (14)$$

where

\mathbf{q}_i represents the displacement and velocities $[\mathbf{z}_i \ \mathbf{v}_i]$

\mathbf{A}_i represents the model dynamics

\mathbf{n}_i represents the known forces and nonlinear terms

\mathbf{u}_i represents the unknown forces (torques)

\mathbf{P}_i represents the system matrix relating the unknown forces to the state variable

The measurements are represented with the vector \mathbf{d}_i and are related to the state variables \mathbf{q}_i with a matrix \mathbf{Q} . This will allow the situation where not all the state variables are measured. The general least squares error is now expressed in terms of the vector inner product (\cdot, \cdot) for the N steps as

$$E(\mathbf{u}_i) = \sum_{i=1}^N (\mathbf{Q}\mathbf{q}_i - \mathbf{d}_i, \mathbf{Q}\mathbf{q}_i - \mathbf{d}_i) + (\mathbf{u}_i, \mathbf{B}\mathbf{u}_i) \quad (15)$$

The first term represents a least squares error between the measurements and the

estimates from the model. The second term is known as the regularization term and is crucial to obtaining smooth and reasonable estimates for the unknown forces. The matrix smoothing matrix \mathbf{B} provides a way to control the amount of regularization. The L-curve method can be used to determine the optimal amount of smoothing. This will be demonstrated in the example.

The inverse problem is to find the unknown forces \mathbf{u}_i that minimize the error expressed in Equation 15 while using the model Equations 14. This can be also be thought of as an optimal control problem with \mathbf{u}_i as the control forces. The method used to solve this optimal control problem is dynamic programming. More details and examples can be found in Trujillo and Busby 1997.

The above formulation of the inverse problem is basically linear and in order to apply it to a nonlinear model requires an iterative solution. This is accomplished by replacing the nonlinear terms in the model Equations 13 with a Taylor-series expansion about previously determined displacements \mathbf{z}^*_i or

$$\mathbf{f}_i \sim \mathbf{f}^*_i + \mathbf{J}^*_i (\mathbf{z}_i - \mathbf{z}^*_i) \quad (16)$$

where \mathbf{J}^*_i is the Jacobian matrix. This approximation differs from the one used in the single step integration formulas Equations 11 and 12 in that the entire displacement history is required at each iteration. This is because the dynamic programming method used in the inverse solution requires a backward pass which in turn requires that certain terms be previously computed. Substituting Equation (16) and a similar one for \mathbf{f}_{i+1} into Equations 11 and 12, these equations are to be placed into the state space form of

Equation 14. First

$$(\mathbf{M} + \mathbf{I}) \mathbf{v}_{i+1} + h\mathbf{f}_{i+1}^*/2 + h\mathbf{J}_{i+1}^* (\mathbf{z}_{i+1} - \mathbf{z}_{i+1}^*)/2 = (\mathbf{M} + \mathbf{I}) \mathbf{v}_i - h\mathbf{f}_i^*/2 - h\mathbf{J}_i^* (\mathbf{z}_i - \mathbf{z}_i^*)/2 + h(\mathbf{T}_i - \mathbf{g}_i) \quad (16a)$$

The unknown torques \mathbf{u}_i are introduced using equation (9) and introducing \mathbf{D}_i^* which will be assumed to depend on \mathbf{z}_i^* .

$$\mathbf{T}_i = \mathbf{D}_i^* \mathbf{u}_i \quad (16b)$$

Rearranging gives

$$(\mathbf{M} + \mathbf{I}) \mathbf{v}_{i+1} + h\mathbf{J}_{i+1}^* \mathbf{z}_{i+1}/2 = (\mathbf{M} + \mathbf{I}) \mathbf{v}_i - h\mathbf{J}_i^* \mathbf{z}_i/2 + h(\mathbf{J}_{i+1}^* \mathbf{z}_{i+1}^* + h\mathbf{J}_i^* \mathbf{z}_i^*)/2 - h(\mathbf{f}_{i+1}^* + \mathbf{f}_i^*)/2 - h\mathbf{g}_i + h\mathbf{D}_i^* \mathbf{u}_i \quad (16c)$$

Replacing \mathbf{z}_{i+1} with

$$\mathbf{z}_{i+1} = \mathbf{z}_i + h(\mathbf{v}_{i+1} + \mathbf{v}_i)/2 \quad (11)$$

$$(\mathbf{M} + \mathbf{I} + h^2\mathbf{J}_{i+1}^*/4) \mathbf{v}_{i+1} = (\mathbf{M} + \mathbf{I} - h^2\mathbf{J}_{i+1}^*/4) \mathbf{v}_i - h(\mathbf{J}_{i+1}^* + \mathbf{J}_i^*) \mathbf{z}_i/2 + h(\mathbf{J}_{i+1}^* \mathbf{z}_{i+1}^* + h\mathbf{J}_i^* \mathbf{z}_i^*)/2 - h(\mathbf{f}_{i+1}^* + \mathbf{f}_i^*)/2 - h\mathbf{g}_i + h\mathbf{D}_i^* \mathbf{u}_i \quad (16d)$$

The above can be simplified by first defining some submatrices

$$\mathbf{K}_i = (\mathbf{M} + \mathbf{I} + h^2\mathbf{J}_{i+1}^*/4)^{-1} (\mathbf{M} + \mathbf{I} - h^2\mathbf{J}_{i+1}^*/4)$$

$$\mathbf{C}_i = -(\mathbf{M} + \mathbf{I} + h^2\mathbf{J}_{i+1}^*/4)^{-1} (\mathbf{J}_{i+1}^* + \mathbf{J}_i^*) \quad (16e)$$

$$\mathbf{D}_i = (\mathbf{M} + \mathbf{I} + h^2\mathbf{J}_{i+1}^*/4)^{-1} \mathbf{D}_i^*$$

$$\mathbf{b}_i = (\mathbf{M} + \mathbf{I} + h^2\mathbf{J}_{i+1}^*/4)^{-1} [(\mathbf{J}_{i+1}^* \mathbf{z}_{i+1}^* + h\mathbf{J}_i^* \mathbf{z}_i^*)/2 - (\mathbf{f}_{i+1}^* + \mathbf{f}_i^*)/2 - \mathbf{g}_i]$$

Equation (16d) becomes

$$\mathbf{v}_{i+1} = \mathbf{K}_i \mathbf{v}_i + h\mathbf{C}_i \mathbf{z}_i/2 + h\mathbf{b}_i + h\mathbf{D}_i \mathbf{u}_i \quad (16f)$$

The displacement equation becomes

$$\mathbf{z}_{i+1} = (\mathbf{I} + h^2\mathbf{C}_i/4)\mathbf{z}_i + h(\mathbf{I} + \mathbf{K}_i)/2\mathbf{v}_i + h^2\mathbf{b}_i/2 + h^2\mathbf{D}_i/2\mathbf{u}_i \quad (16g)$$

$$\mathbf{v}_{i+1} = h\mathbf{C}_i/2\mathbf{z}_i + \mathbf{K}_i\mathbf{v}_i + h\mathbf{b}_i + h\mathbf{D}_i\mathbf{u}_i \quad (16h)$$

These are the final state space formulas that correspond to Equation (14).

$$\begin{bmatrix} \mathbf{z}_{i+1} \\ \mathbf{v}_{i+1} \end{bmatrix} = \begin{bmatrix} \mathbf{I} + h^2\mathbf{C}_i/4 & h(\mathbf{I} + \mathbf{K}_i)/2 \\ h\mathbf{C}_i/2 & \mathbf{K}_i \end{bmatrix} \begin{bmatrix} \mathbf{z}_i \\ \mathbf{v}_i \end{bmatrix} + \begin{bmatrix} h^2\mathbf{b}_i/2 \\ h\mathbf{b}_i \end{bmatrix} + \begin{bmatrix} h^2\mathbf{D}_i/2 \\ h\mathbf{D}_i \end{bmatrix} \mathbf{u}_i \quad (16i)$$

which is in the state space form

$$\mathbf{q}_{i+1} = \mathbf{A}_i \mathbf{q}_i + \mathbf{n}_i + \mathbf{P}_i \mathbf{u}_i \quad (14)$$

So that

$$\mathbf{A}_i = \begin{bmatrix} \mathbf{I} + h^2\mathbf{C}_i/4 & h(\mathbf{I} + \mathbf{K}_i)/2 \\ h\mathbf{C}_i/2 & \mathbf{K}_i \end{bmatrix} \quad (16j)$$

$$\mathbf{n}_i = \begin{bmatrix} h^2\mathbf{b}_i/2 \\ h\mathbf{b}_i \end{bmatrix} \quad (16k)$$

$$\mathbf{P}_i = \begin{bmatrix} h^2\mathbf{D}_i/2 \\ h\mathbf{D}_i \end{bmatrix} \quad (16l)$$

The dynamic programming solution requires a backwards pass followed by a forward pass integrating the discrete equations 14 with the estimated optimal forces. The initial backwards pass will use the measurements for the \mathbf{z}_i^* terms. Subsequent iterations will use the current estimates obtained from the forward pass. These iterations continue until the differences are less than a specified tolerance.

The dynamic programming formulas are included here for sake of completeness.

First the backwards pass starting from the end of the process $n=N$ computes the vectors \mathbf{s}_n

and \mathbf{R}_n . Some intermediate variables are defined to simplify the formulas

$$\mathbf{D}_n^{-1} = -2(\mathbf{B} + \mathbf{P}_{n-1}^T \mathbf{R}_n \mathbf{P}_{n-1}) \quad (17a)$$

$$\mathbf{H}_n^T = 2\mathbf{R}_n \mathbf{P}_{n-1} \quad (17b)$$

The recursive formulas are then

$$\mathbf{R}_{n-1} = \mathbf{Q}^T \mathbf{Q} + \mathbf{A}_{n-1}^T (\mathbf{R}_n + \mathbf{H}_n^T \mathbf{D}_n \mathbf{H}_n / 2) \mathbf{A}_{n-1} \quad (17c)$$

$$\mathbf{s}_{n-1} = -2\mathbf{Q}^T \mathbf{d}_{n-1} + \mathbf{A}_{n-1}^T (2\mathbf{R}_n + \mathbf{H}_n^T \mathbf{D}_n \mathbf{H}_n) \mathbf{n}_{n-1} + \mathbf{A}_{n-1}^T (\mathbf{I} + \mathbf{H}_n^T \mathbf{D}_n \mathbf{P}_{n-1}^T) \mathbf{s}_n \quad (17d)$$

The initial conditions for the backwards pass are at $n = N$

$$\mathbf{R}_N = \mathbf{Q}^T \mathbf{Q} \quad (17d)$$

$$\mathbf{s}_N = -2\mathbf{Q}^T \mathbf{d}_N \quad (17e)$$

The forward pass uses Equation 11 with the optimal forces computed as

$$\mathbf{u}_n = \mathbf{D}_{n+1} \mathbf{P}_n^T \mathbf{s}_{n+1} + \mathbf{D}_{n+1} \mathbf{H}_{n+1} \mathbf{A}_n \mathbf{q}_n + \mathbf{D}_{n+1} \mathbf{H}_{n+1} \mathbf{n}_n \quad (17f)$$

6.0 Numerical Example of an Inverse Problem

In order to test the performance of the proposed inverse problem methodology, the Gen model in Section 4 was used to simulate measurements. The x and y displacement time histories calculated with the generalized coordinate model (time step of 0.01 seconds) , Gen model, were used and in order to simulate experimental inaccuracies, a random noise with a sigma of 0.001m was also added to the displacements. The inverse problem is to estimates the time histories of the applied torques using these simulated displacement measurements.

The estimated torques for a smoothing parameter of $1.0\text{E-}4$ are presented in

Figures 5 and 6. Recall that the applied torques used in the validation models were

Torque on link 1 = 1.0N-m for $0. < \text{time} < 0.5$ seconds, and zero elsewhere

Torque on link 2 = -1.0N-m for $1.0 < \text{time} < 1.5$ seconds, and zero elsewhere

As the figures show the estimates are fairly good. There is some smoothing of the torque estimates at the points of discontinuities. The spurious oscillations are undoubtedly caused by stiff spring present in the FE link model. An experiment with noise free data also showed slight oscillations in the torque estimates.

The optimal choice for the smoothing parameter was selected using an L-curve (Hansen 1992) analysis. This curve is created by solving the inverse problem for various values of the smoothing parameter and then plotting the total error, the first term in Equation 15 against the total force ($\mathbf{u}_i, \mathbf{u}_i$). The optimal choice for the smoothing parameter is then near the corner of the curve. Figure 7 was created using different smoothing parameters and the one chosen for this problem ($B= 1.0E-4$) is indicated on the figure.

7.0 Discussion

Casting the highly nonlinear inverse problem for link mechanisms as a discrete optimal control problem and solving it with dynamic programming shows considerable promise. The use of natural coordinates for the link model and a stiff spring to approximate a rigid link makes the formulation of the solution easier but did introduce

spurious oscillations into the optimal estimates of the torque time histories. The discontinuous nature of the applied torque time histories was purposely chosen to excite these oscillations. A possible technique for reducing these oscillations would be to add a small axial damper in parallel with the spring. The disadvantage is that the damper would take energy out of the system whereas the current model preserves energy.

The robust performance of the dynamic programming method was demonstrated and the fact that the L-curve gave an excellent estimate of the smoothing parameter is encouraging. This first simple example brings up several issues that will be investigated in future work, mainly the effects of integration step size and noise levels on the optimal estimates. Also, the addition of a damper would undoubtedly help smooth the estimates but whether they would also degrade them is to be determined. In linear inverse structural dynamics problems, the use of first order regularization proved superior to just regulating the unknown forces. First order regularization uses the first derivative of the unknown forces \mathbf{u}_i in the error Equation 15. This was not investigated in this paper but might also provide better estimates (Busby, 1997). Lastly, the application of the method to larger models, including closed ones, is another area of future work. Fortunately, the finite element methods and the dynamic programming formulas presented here are directly applicable to models of any size and there are no foreseen reasons why three dimensional models could not also be solved.

References:

- Busby, H.R., and Trujillo, D.M., 1997, Optimal regularization of an inverse dynamics problem, *Computers and Structures*, 63(2), 243-248, April..
- Garcia de Jalon, J., and Bayo, E., 1994. *Kinematic and Dynamic Simulation of Multi-Body Systems – the Real-Time Challenge*. Springer-Verlag, New York.
- Ghorbel, F., Chetelat, O., and Longchamp R., 1994. A reduced model for constrained rigid bodies with application to parallel robots. *Proc. of the IFAC Symposium of Robot Control SYROCO '94*, 57-62, Capri, Italy, September 19-21.
- Hansen, P.C., 1992. Analysis of discrete ill-posed problems by means of the L-curve. *SIAM Review*, 34(4), 561-580, December.
- Hollandsworth, P.E., and Busby, H. R., 1989. Impact force identification using the general inverse technique. *Int. J. Impact Eng*, 8, 315-322.
- Isobe, D., 2001. A unified numerical scheme for calculating inverse dynamics of open/closed link mechanisms. *IECON'01: The 27th Annual Conference of the IEEE Industrial Electronics Society*, 341-344, Denver, USA.
- Klein, S., and Trujillo, D.M., 1983. An unconditionally stable finite element analysis for nonlinear structures. *Computers and Structures*, 16,187-197.
- Risher, D.W., Schutte, L.M., and Runge, C.F., 1997. The use of inverse dynamics solutions in direct dynamics simulations. *Journal of Biomechanical Engineering*, 119, 417-422.
- Runge, C.F., Zajac, F.E., Allum, J.H.J., Risher, D. W., Bryson, A. E. , and Honegger, F.,

1995. Estimating net joint torques from kinesiological data using optimal linear system theory. IEEE Transactions on Biomedical Engineering. Vol. 42, No. 12, 1158-1164.

Tanaka, M., and Chen, W., 2000. Identification of elastodynamics loads using DRBEM and dynamic programming filter. Inverse Problems in Engineering Mechanics II, 333-342, Elsevier Science Ltd.

Trujillo, D.M., and Busby, H.R., 1997. Practical Inverse Analysis in Engineering, CRC Press LLC, Boca Raton, Fl.

Figure 1 Link Properties

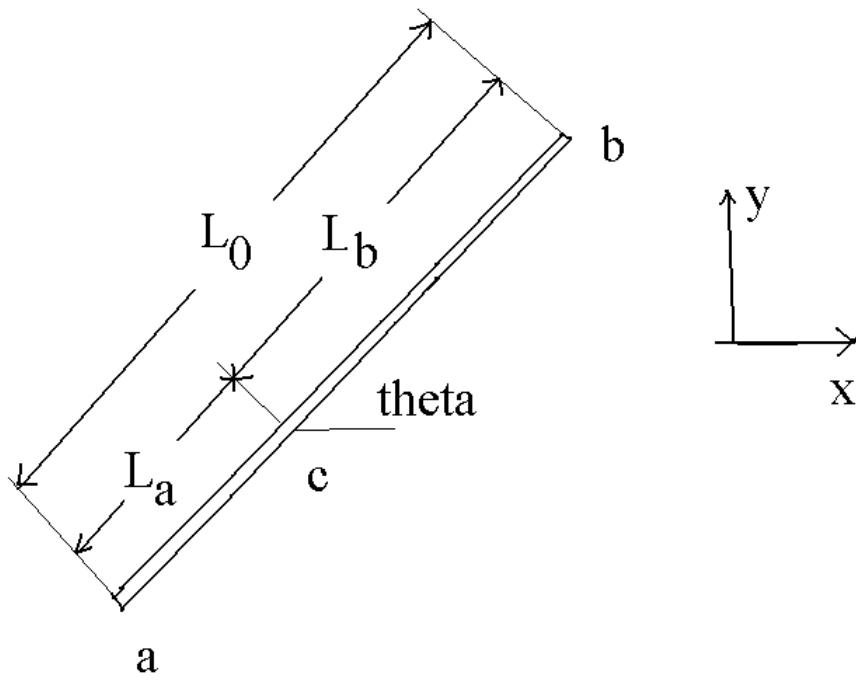


Figure 2 Comparison of Theta 1 Velocities

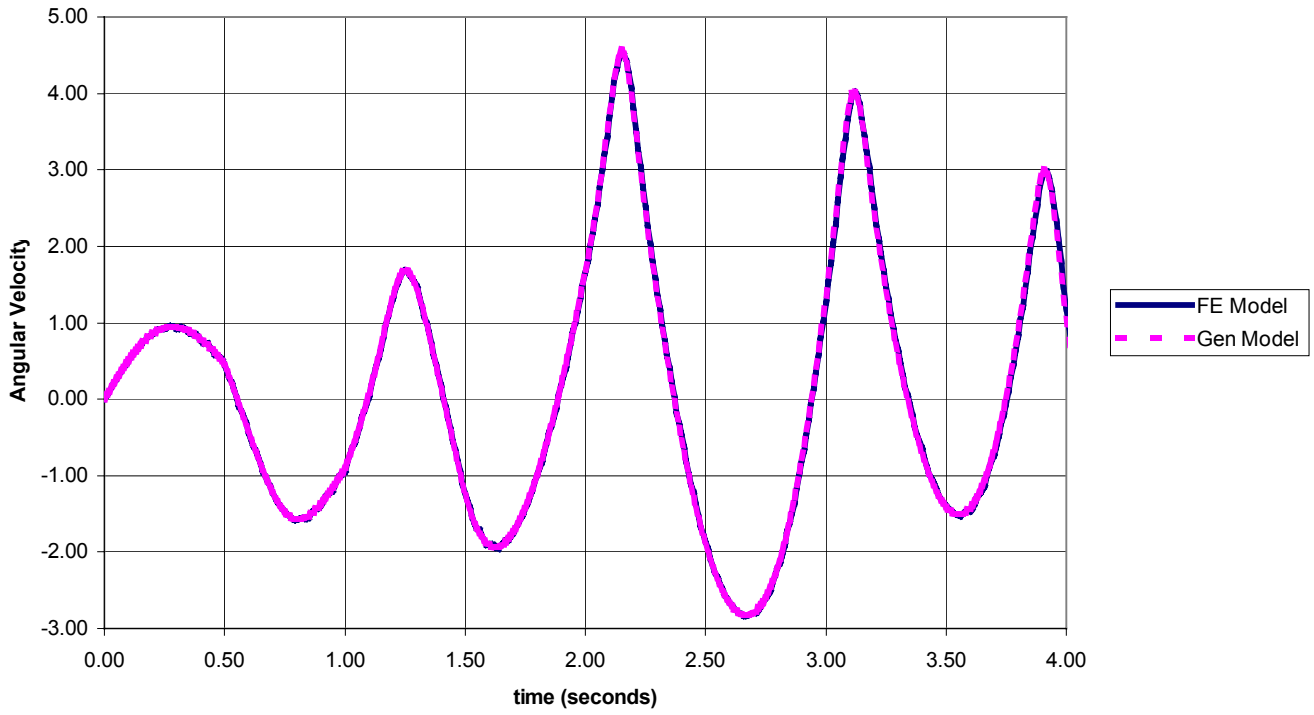


Figure 3 Comparison of Theta 2 Velocities

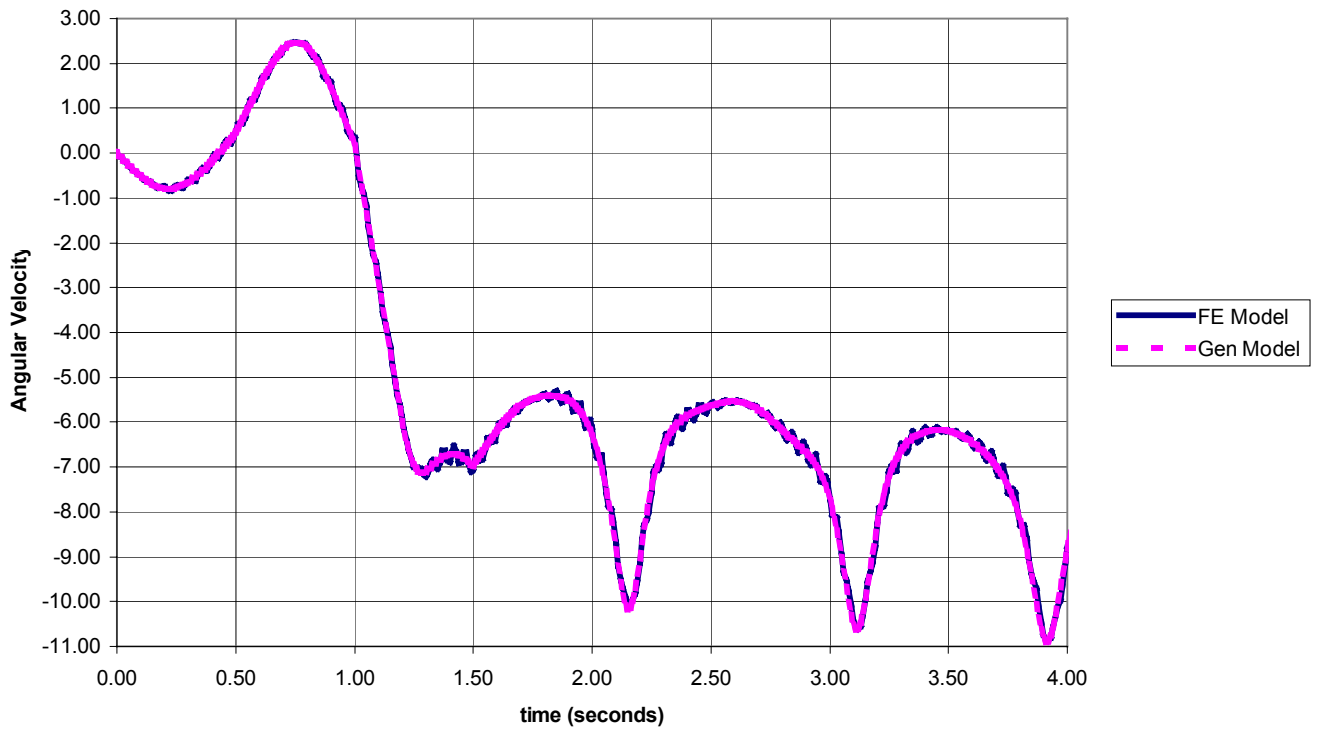


Figure 4 Displacement Histories of Tip

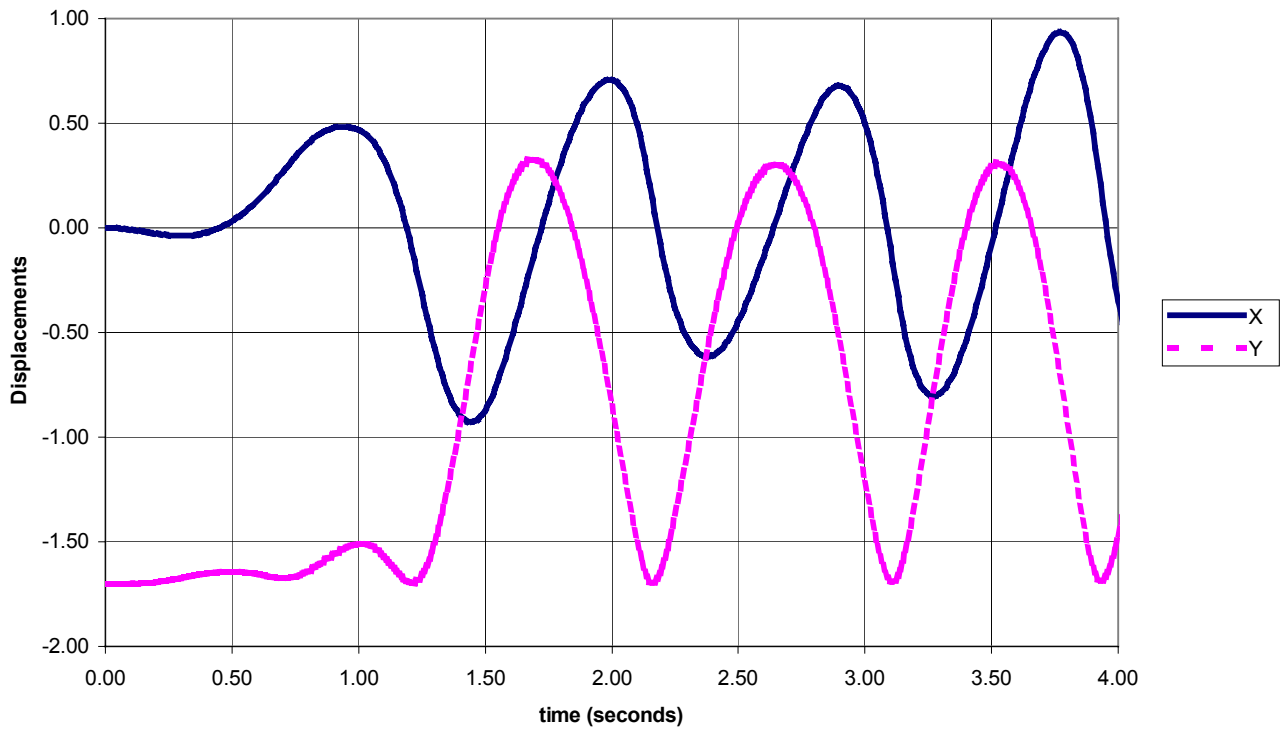


Figure 5 Optimal Estimate of Torque 1

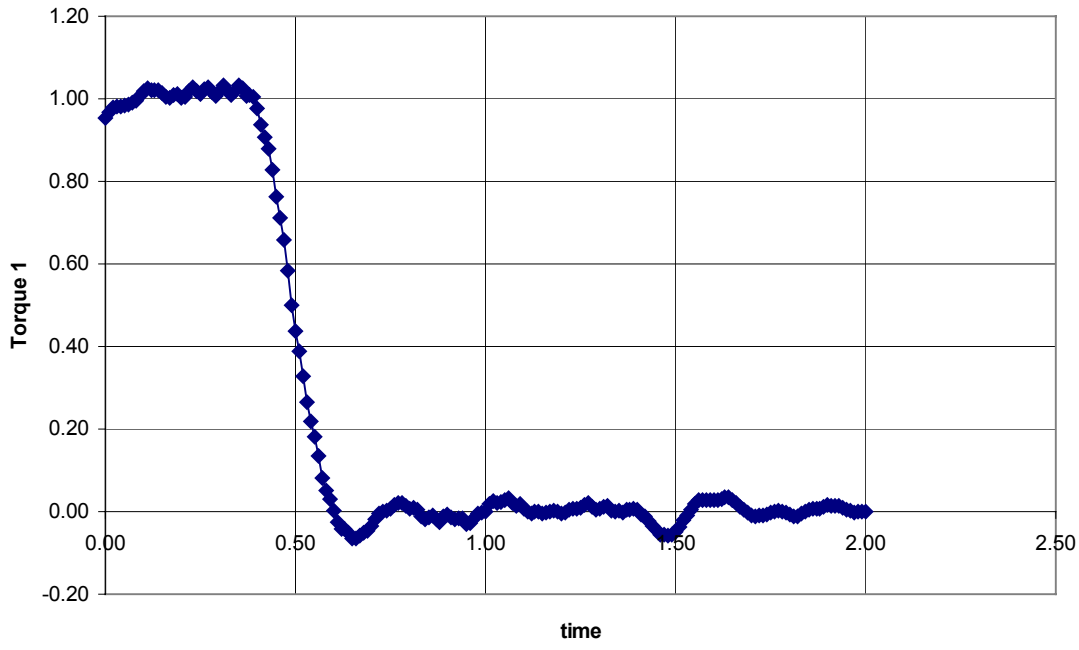


Figure 6 Optimal Estimate of Torque 2

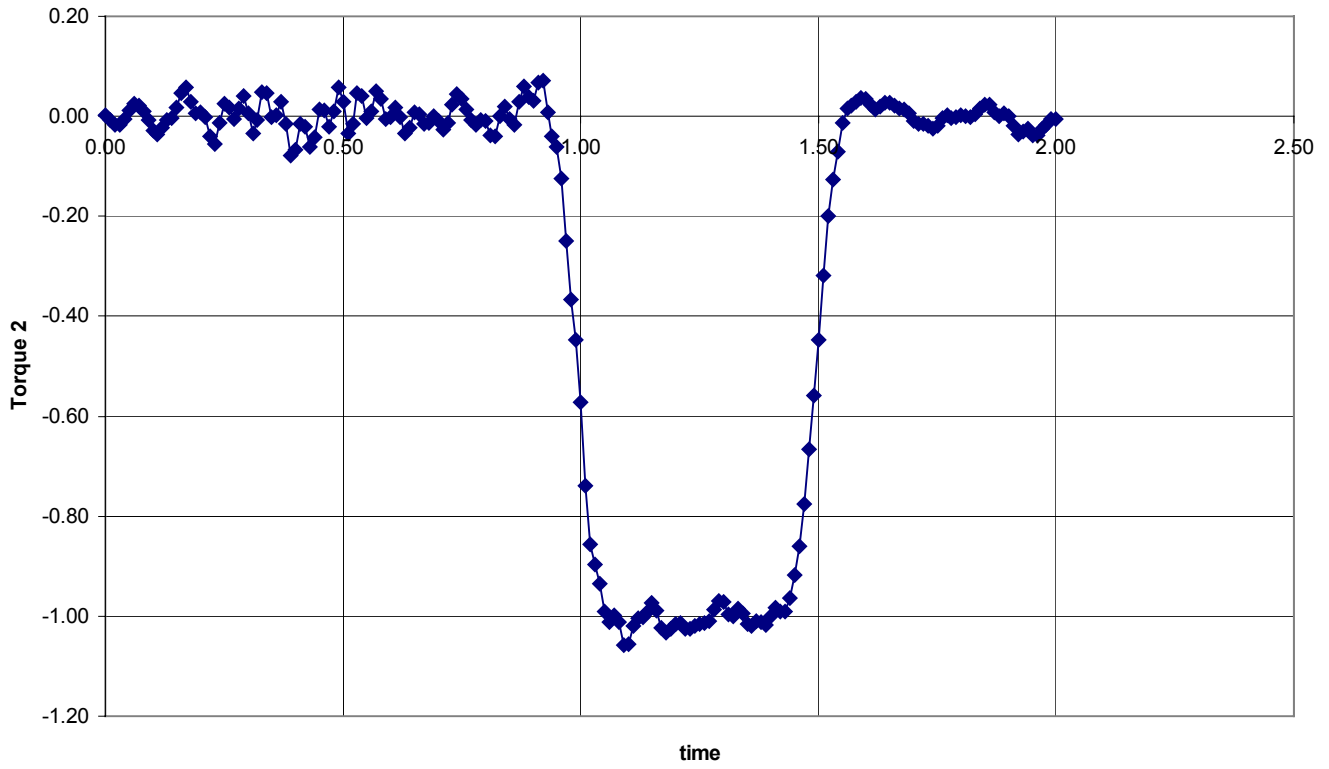
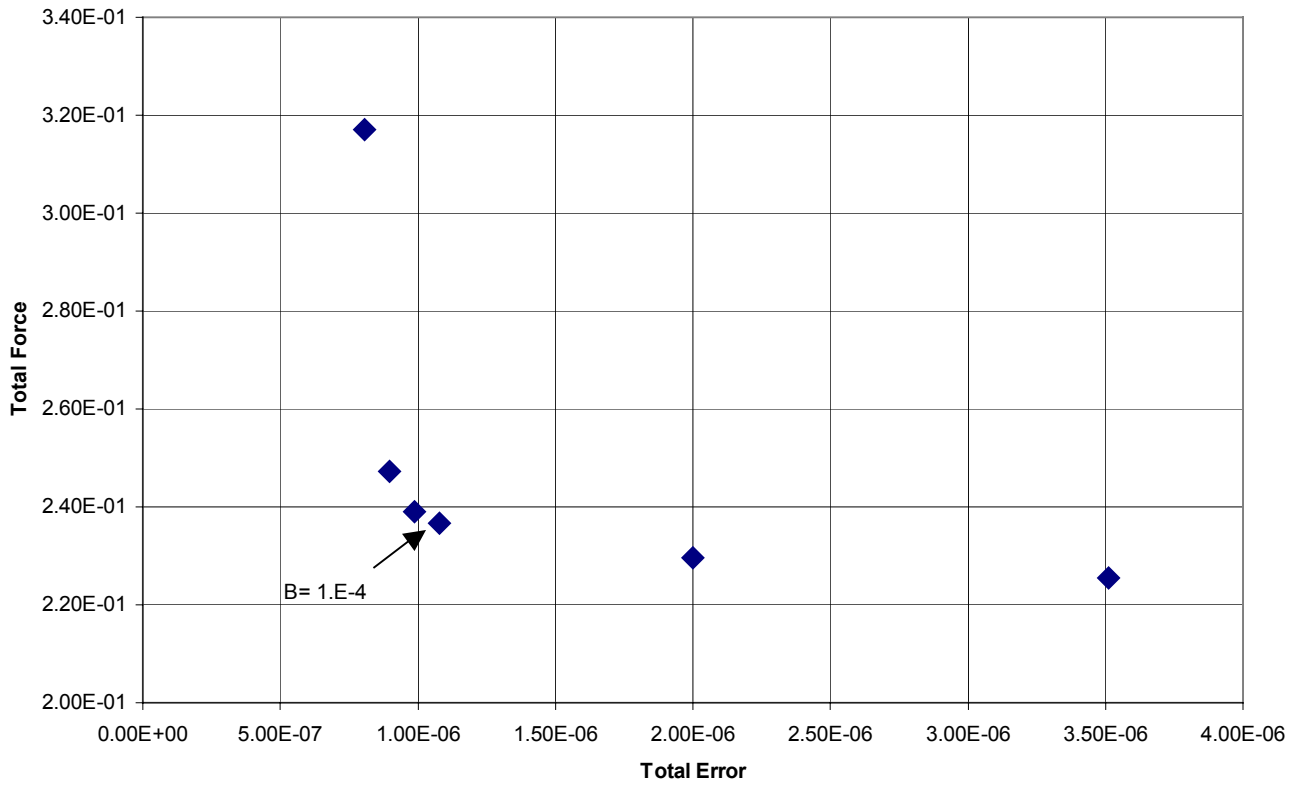


Figure 7 L-curve for Smoothing Parameters



Appendix A Lagrange's Equations

Lagrange's equations[see Reference] form the basis for deriving the finite element link model. This equation is in terms of the generalized coordinates q_i

$$\frac{\partial}{\partial t} \left(\frac{\partial T}{\partial \dot{q}_i} \right) - \frac{\partial T}{\partial q_i} + \frac{\partial V}{\partial q_i} = Q_i$$

Where T is the total kinetic energy, V is the total potential energy, and Q_i is the general force related to the work done by the q_i variable ($\delta W_i = Q_i \delta q_i$).

For the link in Figure 1 the total kinetic energy is

$$T = \frac{1}{2} M \dot{c}_x^2 + \frac{1}{2} M \dot{c}_y^2 + \frac{1}{2} I \dot{\theta}^2$$

and the total potential energy is due to gravity and the spring

$$V = M g c_y + \frac{1}{2} k (L - L_0)^2$$

where L_0 is the unstretched length and

$$L^2 = (x_b - x_a)^2 + (y_b - y_a)^2$$

Expressing center of gravity variables in terms of the end variables

$$c_x = b x_a + a x_b \quad c_y = b y_a + a y_b$$

where $a = L_a / (L_a + L_b)$ $b = L_b / (L_a + L_b)$

The derivatives are

$$\dot{c}_x = b \dot{x}_a + a \dot{x}_b \quad \dot{c}_y = b \dot{y}_a + a \dot{y}_b$$

The portion of the kinetic energy due to translation is

$$\frac{1}{2} M (b \dot{x}_a + a \dot{x}_b)^2 + \frac{1}{2} M (b \dot{y}_a + a \dot{y}_b)^2$$

Taking the partial derivative of this expression with respect to \dot{x}_a

$$\frac{\partial}{\partial t} \left(\frac{\partial T}{\partial \dot{x}_a} \right) = Mb^2 \ddot{x}_a + Mab \ddot{x}_b \quad (\text{translation portion})$$

The angular velocity expression begins with

$$x_a = c_x - L_a \cos(\theta) \quad y_a = c_y - L_a \sin(\theta)$$

$$x_b = c_x + L_b \cos(\theta) \quad y_b = c_y + L_b \sin(\theta)$$

Taking derivatives of these expressions and after some manipulations

$$\dot{\theta}^2 = [(\dot{x}_b - \dot{x}_a)^2 + (\dot{y}_b - \dot{y}_a)^2] / L_0^2$$

The rotational kinetic energy is then

$$\frac{1}{2} I \dot{\theta}^2 = \frac{1}{2} I [(\dot{x}_b - \dot{x}_a)^2 + (\dot{y}_b - \dot{y}_a)^2] / L_0^2$$

Taking derivatives with respect to \dot{x}_a gives

$$\frac{\partial}{\partial t} \left(\frac{\partial T}{\partial \dot{x}_a} \right) = I / L_0^2 \ddot{x}_a - I / L_0^2 \ddot{x}_b \quad (\text{rotational portion})$$

The potential energy due to gravity is

$$V = Mg c_y = Mg(b y_a + a y_b)$$

and

$$\frac{\partial V}{\partial y_a} = Mgb \quad \frac{\partial V}{\partial y_b} = Mga$$

which contributes to the gravity vector Equation (8).

The axial spring is represented with the potential energy

$$\frac{1}{2} k (L - L_0)^2$$

where L_0 is the unstretched length and

$$L^2 = (x_b - x_a)^2 + (y_b - y_a)^2$$

The partial derivative with respect to x_a is

$$\frac{\partial V}{\partial x_a} = k(L - L_0) \frac{\partial L}{\partial x_a}$$

where
$$\frac{\partial L}{\partial x_a} = (x_a - x_b)/L$$

which gives the contribution

$$\frac{\partial V}{\partial x_a} = k(L - L_0)(x_a - x_b)/L$$

The work done by the applied torque $T(t)$ can be expressed as

$$\delta W_i = T \delta \theta$$

Using

$$x_a = c_x - L_a \cos(\theta) \quad y_a = c_y - L_a \sin(\theta)$$

and the derivatives

$$\dot{x}_a = \dot{c}_x + L_a \sin(\theta) \dot{\theta} \quad \dot{y}_a = \dot{c}_y - L_a \cos(\theta) \dot{\theta}$$

Forming $L_a(\sin^2 \theta + \cos^2 \theta) \dot{\theta}$ gives an expression angular velocity

$$\dot{\theta} = [(y_b - y_a)(\dot{x}_a - \dot{x}_b) - (x_b - x_a)(\dot{y}_a - \dot{y}_b)] / L_0^2$$

Using the incremental form

$$\delta \theta = [(y_b - y_a)(\delta x_a - \delta x_b) - (x_b - x_a)(\delta y_a - \delta y_b)] / L_0^2$$

Equating each variable in the Lagrange equation yields the final torque vector Equation (9).

$$\mathbf{T} = (T(t)/L_o^2) \begin{bmatrix} (y_b - y_a) \\ -(x_b - x_a) \\ -(y_b - y_a) \\ (x_b - x_a) \end{bmatrix} \quad (9)$$

Reference : R.W. Clough and J. Penzien, Dynamics of Structures, McGraw-Hill , Inc, 1975.