FIFTH INTERNATIONAL CONGRESS ON SOUND AND VIBRATION
DECEMBER 15-18, 1997
ADELAIDE, SOUTH AUSTRALIA

PERFORMANCE EVALUATION OF FINITE DIFFERENCE AND FINITE ELEMENT METHODS IN THE REAL-TIME SIMULATION OF FLEXIBLE ROBOT MANIPULATORS

M. O. Tokhi §, Z. Mohamed ‡ and A. K. M. Azad §

§ Department of Automatic Control and Systems Engineering,
The University of Sheffield, UK.
‡ Faculty of Electrical Engineering, University of Technology Malaysia, Malaysia.

ABSTRACT

This paper presents an investigation into the performance evaluation of finite difference and finite element methods in the real-time simulation of flexible robot manipulator systems. A constrained planar single-link flexible manipulator is considered. Finite-dimensional simulation environments characterising the dynamic behaviour of the manipulator are developed using finite difference and finite element methods. The simulation algorithms thus developed are implemented on general-purpose digital processors. Experimental results verifying the performance of the algorithms in characterising dynamic behaviour of the system and comparative performance evaluation of the algorithms on the basis of accuracy and computational efficiency are presented and discussed.

Keywords: Dynamic simulation, finite difference method, finite element method, flexible robot manipulators, real-time simulation.

1. INTRODUCTION

Flexible manipulator systems offer several advantages over their traditional counterparts. These include light weight, faster system response, less power consumption, requiring smaller actuators, more manoeuvrable, more transportable, safer operation due to reduced inertia and in general less overall cost (Azad, 1995; Meng and Chen, 1988).

In order to control flexible manipulators efficiently, they must be modelled accurately. An accurate model will result in a satisfactory and good control. A further requirement is the efficiency in obtaining the model. Various approaches have previously been developed for modelling of flexible manipulators (Azad, 1995). Among these the finite element (FE) and finite difference (FD) techniques are commonly used. These methods allow the development of suitable simulation environments that can be utilised for real-time dynamic characterisation of the system and for test and verification of controller designs.

The FE method has been successfully used to solve many material and structural problems (Meng and Chen, 1988; Usoro *et al.*, 1986). The method involves discretising the actual system into a number of elements whose elastic and inertia properties are obtained from the system. This provides approximate static and dynamic properties of the actual system. the FE method is found to be more suitable for structures of irregular nature with mixed boundary conditions.

The FD method has previously been utilised in the dynamic characterisation of flexible beam and flexible manipulator systems (Azad, 1995; Kourmoulis, 1990; Tokhi and Azad, 1995). The method involves discretising the system into several sections (segments) and developing a linear relation for the deflection of end of each segment using FD approximations. This method is simple in mathematical terms and is found to be more suitable for uniform structures.

The performance of the FD and FE methods have previously been assessed in the dynamic characterisation of systems. However, not much has been reported on a comparative performance evaluation of these methods in the real-time simulation of dynamic systems. The aim of this work is to investigate the performance of the FD and FE methods within such a framework on the basis of accuracy, computational efficiency and computational requirements. The rest of the paper is structured as follows

Section 2 introduces the flexible manipulator considered in this study. Section 3 gives an outline of the FD and FE simulation algorithms characterising the flexible manipulator. Results of implementation of the algorithms are presented and discussed in Section 4. The paper is finally concluded in Section 5.

2. THE FLEXIBLE MANIPULATOR SYSTEM

The single-link flexible manipulator considered in this paper is described in Figure 1, where, I_h represents the hub inertia of the manipulator. A payload mass M_p with its associated inertia I_p is attached to the end-point. A control torque $\tau(t)$ is applied at the hub by an actuator motor. The angular displacement of the manipulator, in moving in the POQ – plane, is denoted by $\theta(t)$. The manipulator is assumed to be stiff in vertical bending and torsion, thus, allowing it to vibrate (be flexible) dominantly in the horizontal direction. The shear deformation and rotary inertia effects are also ignored.

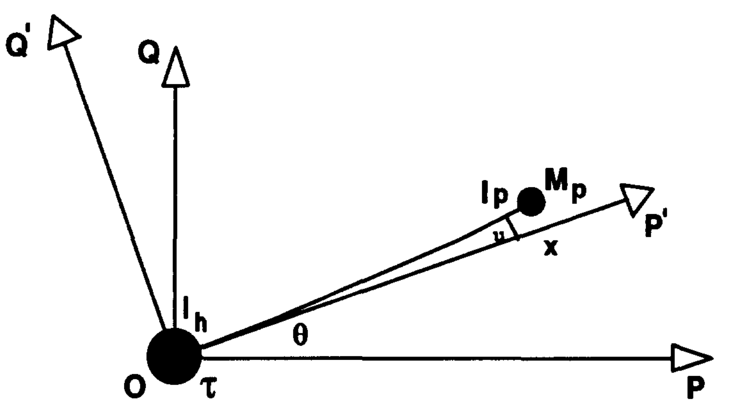

Figure 1: Description of the flexible manipulator system.

For an angular displacement θ and an elastic deflection u the total (net) displacement $y(x,t)$ of a point along the manipulator at a distance x from the hub can be described as a function of both the rigid body motion $\theta(t)$ and elastic deflection $u(x,t)$ measured from the line OX;

$$y(x,t) = x\theta(t) + u(x,t) \tag{1}$$

The dynamic equations of motion of the manipulator can be obtained using the Hamilton's extended principle (Meirovitch, 1967) with the associated kinetic, potential and dissipated energies of the system. The governing equation of motion of the manipulator can thus be obtained as (Tokhi and Azad, 1995)

$$EI\frac{\partial^4 y(x,t)}{\partial x^4} + \rho\frac{\partial^2 y(x,t)}{\partial t^2} = \tau(t) \tag{2}$$

with the corresponding boundary and initial conditions as

$$\begin{aligned}
&y(0,t)=0\ ,\ I_h\frac{\partial^3 y(0,t)}{\partial x\partial t^2} - EI\frac{\partial^2 y(0,t)}{\partial x^2} = \tau(t)\\
&M_p\frac{\partial^2 y(l,t)}{\partial t^2} - EI\frac{\partial^3 y(l,t)}{\partial x^3} = 0,\ \ EI\frac{\partial^2 y(l,t)}{\partial x^2} = 0\\
&y(x,0)=0\ ,\ \ \frac{\partial y(x,0)}{\partial x} = 0
\end{aligned}$$

where E, ρ and I represent the Young modulus, mass density and area moment of inertia of the manipulator. Equation (2) gives the fourth-order partial differential equation (PDE) which represents the dynamic equation describing the motion of the flexible manipulator with no structural damping.

3. SIMULATION ALGORITHMS

In this section the development of FD and FE based simulation algorithms of the manipulator are presented.

3.1 Finite difference algorithm

A finite dimensional simulation of the flexible manipulator system can be developed through discretisation both in time and space coordinates using an FD approximation to the PDE in equation (2). To solve the PDE, it is replaced by a set of difference equations defined by the central difference quotients of the FD method (Lapidus, 1982; Tokhi and Azad, 1995). The manipulator length and movement time are each divided into suitable number of sections of equal length represented by Δx ($x = i\Delta x$) and Δt ($t = j\Delta t$), where i and j are non-negative integer numbers, respectively. A difference equation for the end of each section (grid-point) is developed. The displacement $y_{i,j+1}$ at each time instant can thus be written as

$$y_{i,j+1} = -c\left[y_{i-2,j} + y_{i+2,j}\right] + b\left[y_{i-1,j} + y_{i+1,j}\right] + ay_{i,j} - y_{i,j-1} + \frac{\Delta t^2}{\rho}\tau(i,j) \tag{3}$$

where, $c = \Delta t^2 EI/\rho\Delta x^4$, $a = 2 - 6c$ and $b = 4c$.

Equation (3) gives the displacement of section i of the manipulator at time step $j+1$. Using matrix notation, equation (3) can be written as (Tokhi and Azad, 1995)

$$\mathbf{Y}_{i,j+1} = \mathbf{A}\mathbf{Y}_{i,j} - \mathbf{Y}_{i,j-1} + \mathbf{B}\mathbf{F} \tag{4}$$

where, $\mathbf{Y}_{i,k}$ $(k = j+1, j, j-1)$ represents the deflection of sections $i = 1,\ldots,n$ of the manipulator at time step k, $\mathbf{A}$ is a matrix with entries depending on the physical characteristics of the manipulator and on the boundary and initial conditions related to the dynamic equation of the system, $\mathbf{B} = \Delta t^2/\rho$ and $\mathbf{F} = [\tau(i,j) \quad 0 \quad \cdots \quad 0]^T$. Equation (4) is the general solution of the PDE, giving the displacement of section i of the manipulator at time step $j+1$, which can easily be implemented on a digital processor.

It follows from equation (3) that, to obtain the displacements $y_{i,j+1}$, $y_{n-1,j+1}$ and $y_{n,j+1}$ the displacements of the fictitious points $y_{-1,j}$, $y_{n+1,j}$ and $y_{n+2,j}$ are required. These are obtained using the boundary and initial conditions related to the dynamic equation of the flexible manipulator system. The stability of the algorithm can be examined by ensuring that the iterative scheme described in equation (4) converges to a solution. The necessary and sufficient condition for stability satisfying this convergence requirement is given by $0 \le c \le 0.25$ (Kourmoulis, 1990).

3.2 Finite element algorithm

Since its introduction in the 1950s, the FE method has been continually developed and improved (Fagan, 1992). The FE method involves decomposing the mechanical structure into several simple pieces or elements. The elements are assumed to be interconnected together at certain points known as nodes. For each element, an equation describing the behaviour of the element is obtained through an approximation technique and then assembled together to form a system equation. It is found that by reducing the element size of the structure, that is, increasing the number of elements, the overall solution of the system equation can be made to converge to the exact solution.

The main steps in an FE analysis include (1) discretisation of the structure into elements, (2) selection of an approximating function to interpolate the result, (3) derivation of the basic element equation, (4) calculation of the system equation, (5) incorporation of the boundary conditions and (6) solving the system equation with the inclusion of the boundary conditions. In this manner, the flexible manipulator is treated as an assemblage of n elements and the development of the algorithm can be divided into three main parts: the FE analysis, state-space representation and obtaining the system outputs.

The residual motion of the manipulator can be represented as

$$u(x,t) = N(x)Q(t) \tag{5}$$

where, $Q(t)$ and $N(x)$ represent the nodal displacement and shape function respectively. Substituting for $u(x,t)$ from equation (5) into equation (1) and simplifying yields

$$y(x,t) = N(x)^* Q(t)^* \tag{6}$$

where,

$$Q(t)^* = [\theta(t) \quad Q(t)]^T, \quad N(x)^* = [x \quad N(x)]$$

Using the above the element mass matrix M^e and stiffness matrix K^e can be obtained as (Mohamed, 1995)

$$M^e = \rho A \int_0^L (N^*)^T (N^*) dx, \quad K^e = EI \int_0^L (B^*)^T B^* dx$$

where, A and L are the cross-sectional area and length of the manipulator respectively and $B^* = d^2 N^* / dx^2$.

The shape function N^* and nodal displacement vector Q^* in equation (6) incorporate local and global variables. Among these, the angle $\theta(t)$ and the distance x are global variables while $N(x)$ and $Q(t)$ are local variables when the link is divided into n elements. Defining $s = x - \sum_{i=1}^{n-1} l_i$, where l_i is the length of the ith element, as a local variable of the nth element, the new element mass matrix and stiffness matrix can be obtained for the n elements (Mohamed, 1995). It is noted in this process that, the element mass matrix depends on the element number, whereas the element stiffness matrix has the same value regardless of the element number. The element mass and stiffness matrices thus obtained are assembled to obtain system mass and stiffness matrices, M and K, and used in the Lagrange equation to obtain the dynamic equation of the flexible manipulator as

$$M\ddot{Q}(t) + KQ(t) = F(t) \tag{7}$$

where $F(t)$ is the vector of applied forces and torques and

$$Q(t) = [\theta \quad U_1 \quad \theta_1 \quad . \quad . \quad . \quad U_{n+1} \quad \theta_{n+1}]^T$$

The M and K matrices in equation (7) are of size $m \times m$ and $F(t)$ is of size $m \times 1$, $m = 2n + 1$. For the manipulator, considered as a pinned-free arm, with the applied torque τ at the hub, the flexural and rotational displacement, velocity and acceleration at the hub are zero and the external force is $F = [\tau \quad 0 \quad \cdots \quad 0]^T$. Moreover, in this work it is assumed that $Q(0) = 0$.

The matrix differential equation in equation (7) can be represented in a state-space form as

$$\dot{x} = Ax + Bu \quad , \quad y = Cx + Du$$

where,

$$A = \left[\begin{array}{c|c} 0_m & I_m \\ \hline -M^{-1}K & 0_m \end{array}\right], \quad B = \left[\begin{array}{c} 0_{m\times 1} \\ \hline M^{-1} \end{array}\right], \quad C = [0_m \mid I_m], \quad D = [0_{2m\times 1}],$$

0_m is an $m \times m$ null matrix, I_m is an $m \times m$ identity matrix, $0_{m\times 1}$ is an $m\times 1$ null vector,

$$u = [\tau \quad 0 \quad \cdots \quad 0]^T, \quad x = [\theta \quad U_2 \quad \theta_2 \quad \cdots \quad U_{n+1} \quad \theta_{n+1} \quad \dot{\theta} \quad \dot{U}_2 \quad \dot{\theta}_2 \quad \cdots \quad \dot{U}_{n+1} \quad \dot{\theta}_{n+1}]^T$$

Solving the state-space representation gives the vector of states x, that is, the angular, nodal flexural and rotational displacements and velocities.

4. IMPLEMENTATIONS AND RESULTS

To implement the FD and FE algorithms an aluminium type flexible manipulator of dimensions $960 \times 19.23 \times 3.2\ \mathrm{mm}^3$, mass density $2710\ \mathrm{kg/m}^3$, inertia $0.0495\ \mathrm{kgm}^2$ and $I = 5.1924 \times 10^{-11}\ \mathrm{m}^2$ is considered. For simplicity purposes, the effects of hub inertia and payload are ignored. The simulation algorithms thus developed are coded within MATLAB and implemented on two general purpose computing domains, namely a 486DX (33 MHz) PC and a Sun 4-ELC (33 MHz) SPARC station. A bang-bang input of amplitude $0.1\ \mathrm{Nm}$ and duration $0.6\ \mathrm{sec}$ is used as input torque and the system response is obtained and analysed.

Note that the 486DX and the SPARC processors are not normally used in real-time applications. Moreover, such applications will favour programming languages other than MATLAB. For purposes of this investigation, these resources are adequate with which a consistent set of results would be obtained as with any other digital processor.

To investigate the accuracy of the FD simulation in characterising the behaviour of the system, the algorithm was implemented on the basis of varying number of sections along the link from 5 to 20. It was noted with the response of the system at the end-point, due to the bang-bang torque input using 5 and 20 sections, that, although the responses for the two cases were similar in character, with 20 sections a steady-state level was reached within $0.6\ \mathrm{sec}$, whereas, with 5 sections the response did not fully reach a steady-state level over the $1.2\ \mathrm{sec}$ measurement period. The first three resonance modes of the system with the algorithm using various number of sections are shown in Table 1. These were obtained through spectral analysis of the response of the system at the end-point. It is noted that the resonance frequency corresponding to the first three modes of vibration of the system converge to reasonably stable value with the algorithm using 10 sections or more and for the second and third modes with more than 15 sections. The corresponding execution times achieved with the two processing domains in implementing the FD algorithm with various number of sections are also shown in Table 1. As expected, the execution time increases with increasing number of sections. Moreover, it is noted that the two computing platforms appear to perform at a similar speed with lower number of sections. However, as the number of sections increase the SPARC processor outperforms the 486DX processor significantly. This is mainly due to the run-time memory management and relatively limited cache in the 486DX processor.

To investigate the accuracy of the FE simulation in characterising the behaviour of the system, the algorithm was implemented on the basis of varying number of elements from 1 to 20. It was noted that the response of the system at the end-point due to the bang-bang torque input reached a steady-state level within 0.6 sec with the algorithm using one or more elements. Moreover, the residual motion was found to be predominantly characterised by the first mode of vibration with one element, whereas, with more elements higher modes of vibration were also apparent. This is evidenced in Table 2 with the resonance modes of the system in relation to the number of elements used. It is noted that the number of resonance modes identified increases with increasing number of elements. Moreover, reasonable accuracy in the first mode is achieved with two elements, in the second mode with three elements, in the third mode with more than five elements and in the fourth mode with more than 10 elements. The corresponding execution times achieved with the two processing domains in implementing the FE algorithm with various number of elements are also shown in Table 2. As expected, the execution time increases with increasing number of sections. Moreover, it is noted that the SPARC processor outperforms the 486DX processor significantly. This is mainly due to the run-time memory management and relatively limited cache in the 486DX processor.

Table 1: Modes of vibration of the system and execution times of computing domains with number of FD sections.

Number of sections	Modes of vibration (Hz)			Execution times (sec)	
	Mode 1	Mode 2	Mode 3	486DX	SPARC
5	11.1917	33.5751	60.3109	1.4100	1.2600
10	12.4352	37.9275	77.4093	6.9767	3.6400
15	12.4352	39.5337	81.6062	29.9800	21.0433
20	12.4352	41.3472	81.7617	60.7750	42.9000

Table 2: Modes of vibration of the system and execution times of computing domains with number of FE elements.

Number of elements	Modes of vibration (Hz)				Execution times (sec)	
	Mode 1	Mode 2	Mode 3	Mode 4	486DX	SPARC
1	14.509	57.306	-	-	1.9067	0.8278
2	11.9689	44.974	109.896	-	2.1933	0.9389
3	11.969	40.622	93.575	-	2.2733	1.0833
5	11.969	40.259	86.684	112.798	3.6667	1.4833
10	11.969	40.259	85.596	132.383	11.7333	3.9944
15	11.969	40.259	85.321	135.448	20.3800	9.3834
20	11.969	40.259	85.232	135.285	37.9000	18.7722

Comparing the results in Tables 1 and 2 reveals that reasonable accuracy in characterising the behaviour of the manipulator up to the first two resonance modes is achieved with the FD and FE algorithms using at least 15 sections and 3 elements respectively. The corresponding execution times in implementing the algorithms are 29.98

and 2.2733 sec on the 486DX and 21.0433 and 1.0833 sec on the SPARC processor respectively. With the inclusion of the third resonance mode, however, similar level of accuracy is achieved with the FD and FE algorithms using at least 20 sections and 5 elements respectively. The corresponding execution times in implementing the algorithms are 60.775 and 3.6667 sec on the 486DX and 42.9 and 1.4833 sec on the SPARC processor. In each case, it is seen that the FE algorithm performs more accurately and efficiently than the FD algorithm.

5. CONCLUSION

A comparative performance evaluation of the FD and FE methods on the basis of accuracy and computational efficiency in the simulation of a flexible manipulator system has been presented. Simulation environments characterising the dynamic behaviour of a single-link flexible manipulator have been developed using FD and FE methods. The algorithms thus developed have been implemented on general-purpose computing domains and their performances on the basis of accuracy and computational speed have been investigated. It has been demonstrated that although the FE method is mathematically more complex than the FD method, better accuracy and efficient performance is achieved with the FE method in comparison to the FD method.

6. REFERENCES

Azad, A. K. .M. (1995). *Analysis and design of control mechanisms for flexible manipulator systems*, PhD thesis, Department of Automatic Control and Systems Engineering, The University of Sheffield, UK.

Fagan, M. J. (1992). *Finite element analysis, theory and practice*, Longman, Harlow.

Kourmoulis, P .K. (1990). *Parallel processing in the simulation and control of Flexible Beam Structure Systems*. PhD thesis, Department of Automatic Control and Systems Engineering, The University of Sheffield, UK.

Lapidus, L. (1982). *Numerical solution of partial differential equations in science and engineering*, John Wiley and Sons, New York.

Meirovitch, L. (1967). *Analytical methods in vibrations*, Macmillan, New York.

Meng, C. -H. and J. -S. Chen (1988). Dynamic model and payload - adaptive control of a flexible manipulator, *Proceedings of IEEE Conference on Robotics and Automation*, Philadelphia, April 1988, pp. 448-453.

Mohamed, Z. (1995). *A finite element approach to modelling a single-link flexible manipulator system*, MSc thesis, Department of Automatic Control and Systems Engineering, The University of Sheffield, UK.

Tokhi, M. O. and A. K. .M. Azad (1995). Real-time finite difference simulation of a single-link flexible manipulator system incorporating hub inertia and payload, *Proceedings of IMechE - I: Journal of Systems and Control Engineering*, **209**, (I1), pp. 21-33.

Usoro, P. B., R. Nadira and S. S. Mahil (1986). A finite element/Lagrange approach to modelling lightweight flexible manipulators, *Transactions of ASME Journal of Dynamic Systems, Measurement and Control*, **108**, (3), pp. 198-205.