

SCEFMAS: An Environment for Dynamic Characterisation and Control of Flexible Robot Manipulators*

M. O. TOKHI, A. K. M. AZAD and H. POERWANTO

Department of Automatic Control and Systems Engineering, The University of Sheffield, UK.

E-mail: O.Tokhi@sheffield.ac.uk

This paper presents the development of an interactive and user-friendly environment for simulation and control of flexible manipulator systems. A constrained planar single-link flexible manipulator is considered. A simulation algorithm characterising the dynamic behaviour of the manipulator is developed using finite difference methods. Several open-loop and closed-loop control strategies are developed and incorporated into the environment. The environment is implemented using MATLAB and SIMULINK. Several case studies, demonstrating the utilisation and potential of the environment are presented and discussed. The environment provides a valuable computer-aided educational and research facility for understanding the behaviour of flexible manipulator systems and development of various controller designs.

INTRODUCTION

FLEXIBLE MANIPULATOR systems are receiving increasing attention due to their advantages over conventional robot manipulators. The advantages of flexible manipulators are faster response, lower energy consumption, relatively smaller actuators, higher payload to weight ratio and, in general, less overall cost [1]. Some of the current applications of such manipulators include spacecraft, remote manipulation and radioactive material handling in nuclear power plants. Further industrial applications include light-weight assembling and manipulation tasks such as component assembling on a PCB, and car painting. Due to the flexible nature of such manipulators, induced vibrations appear in the system during and after a positioning motion. This restricts their wide spread use in industry. A considerable amount of research work has already been carried out on the vibration control of flexible manipulators. However, a generic solution to the problem is yet to be obtained [2–6].

To formulate and implement an effective control strategy for efficient vibration suppression of the system, it is important to recognise the flexible nature of the manipulator and construct a mathematical model for the system that accounts for the interactions with actuators and payload [7]. Such a model can be constructed using partial differential equations (PDEs). The finite element (FE) method has also been utilised to describe the flexible behaviour of manipulators. The computational complexity and consequent software coding

involved in the FE method is a major disadvantage of this technique [8]. An alternative solution is to utilise a finite difference (FD) method for simulating the system. The FD method has previously been utilised to simulate flexible manipulator systems using C++, Pascal and MATLAB software environments [9]. These environments have proved to be effective in the simulation of the system for test and verification of controller designs. However, these do not incorporate the features of an interactive and user-friendly nature that are desired especially in computer-aided teaching and research. Such features can be incorporated by developing a simulation environment for the flexible manipulator using SIMULINK and MATLAB.

This paper presents the development of an interactive and user-friendly environment referred to as SCEFMAS (Simulation and Control Environment of Flexible Manipulator Systems). A constrained planar single-link flexible manipulator is considered in this study. A finite dimensional simulation of the manipulator is developed by discretising the governing dynamic equations of motion of the system using FD methods [10]. Various approaches of implementation of the algorithm within SIMULINK are investigated and assessed in terms of performance. Two classes of control, namely open-loop and closed-loop strategies are developed and incorporated within the environment. The open-loop control strategies are based on bang-bang, Gaussian-shaped and filtered command inputs while the closed-loop control strategies include joint-based PD controller and hybrid collocated and non-collocated controller methods. Simulated results verifying the performance of the simulation environment

* Accepted 9 January 1999.

and the control strategies are presented and discussed through case studies.

SIMULATION OF FLEXIBLE MANIPULATOR SYSTEMS

The flexible manipulator system

The flexible manipulator system under consideration is modelled as a pinned-free flexible beam, with a mass at the hub, which can bend freely in the horizontal plane but is stiff in vertical bending and torsion. The model development utilises the Lagrange equation and modal expansion method [11, 12]. To avoid the difficulties arising due to time varying length, the length of the manipulator is assumed to be constant.

A schematic representation of the manipulator with a moment of inertia I_h , hub inertia I_h , a linear mass density ρ and a length of l is shown in Fig. 1. The payload mass is M_p and I_p is the inertia associated with the payload. A control torque $\tau(t)$ is applied at the hub of the manipulator by an actuator motor. The angular displacement of the manipulator, in moving in the POQ plane, is denoted by $\theta(t)$. The width of the arm is assumed to be much greater than its thickness, thus, allowing the manipulator to vibrate (be flexible) dominantly in the horizontal direction. The shear deformation and rotary inertia effects are also ignored [2, 9].

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 [2, 9]:

$$y(x, t) = x\theta(t) + u(x, t) \quad (1)$$

To obtain equations of motion of the manipulator, the associated energies have to be obtained. These include the kinetic, potential and dissipated energies. Thus, using the Hamiltonian's extended method, the dynamic equation of the flexible manipulator with the associated boundary and initial conditions can be expressed as [2]:

$$EI \frac{\partial^4 y(x, t)}{\partial x^4} + \rho \frac{\partial^2 y(x, t)}{\partial t^2} - D_s \frac{\partial^3 y(x, t)}{\partial x^2 \partial t} = 0 \quad (2)$$

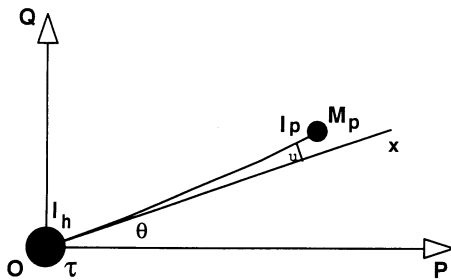


Fig. 1. Schematic representation of the flexible manipulator system.

$$y(0, t) = 0$$

$$I_h \frac{\partial^3 y(0, t)}{\partial t^2 \partial x} - EI \frac{\partial^2 y(0, t)}{\partial x^2} = \tau(t) \quad (3)$$

$$M_p \frac{\partial^2 y(l, t)}{\partial x^2} - EI \frac{\partial^3 y(l, t)}{\partial x^3} = 0$$

$$EI \frac{\partial^2 y(l, t)}{\partial x^2} = 0$$

$$y(x, 0) = 0, \quad \frac{\partial y(x, 0)}{\partial x} = 0 \quad (4)$$

where, E is the Young modulus, I is the second moment of inertia of the manipulator and $\tau(x, t)$ is the applied torque. For the system under consideration, the torque $\tau(x, t)$ is applied at the hub of the manipulator, therefore, it can be represented as $\tau(0, t)$ or simply $\tau(t)$.

Algorithm development

The PDE in equation (2) describing the flexible manipulator system is of a hyperbolic type and can be classified as a boundary value problem. This can be solved using an FD method [9]. This involves dividing the arm into a finite number of equal-length sections and developing a linear difference equation describing the deflection of end of each section (grid-point). Thus, using the FD method, a solution of the PDE in equation (2) can be obtained as [2]:

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

$$+ b[y_{i+1,j} + y_{i-1,j}] + a y_{i,j} - y_{i,j-1}$$

$$+ d[y_{i+1,j} - 2y_{i,j} + y_{i-1,j} - y_{i+1,j-1}$$

$$+ 2y_{i,j-1} - y_{i-1,j-1}] \quad (5)$$

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

Equation (5) gives the displacement of section i of the manipulator at time step $j+1$. It follows from this equation that, to obtain the displacements $y_{n-1,j+1}$ and $y_{n,j+1}$, the displacements of the fictitious points $y_{n+2,j}$, $y_{n+1,j}$ and $y_{n+1,j-1}$ are required. These fictitious displacements can be obtained using the boundary conditions related to the dynamic equation of the flexible manipulator. The discrete form of the corresponding boundary conditions in equations (3) and (4), obtained in a similar manner as above, are:

$$y_{0,j} = 0 \quad (6)$$

$$y_{-1,j} = y_{1,j} + \frac{\Delta x I_h}{EI \Delta t^2} [y_{1,j+1} - 2y_{1,j} + y_{1,j-1}]$$

$$+ \frac{\Delta t^2}{EI} \tau(j) \quad (7)$$

$$y_{n+2,j} = 2y_{n+1,j} - 2y_{n-1,j} + y_{n-2,j} + \frac{2\Delta x^3 M_p}{\Delta t^2 EI} [y_{n,j+1} - 2y_{n,j} + y_{n,j-1}] \quad (8)$$

$$y_{n+1,j} = 2y_{n,j} - y_{n-1,j} \quad (9)$$

Manipulating equation (5) using equations (6) to (9) yields a matrix formulation of the above as:

$$\mathbf{Y}_{i,j+1} = \mathbf{A}\mathbf{Y}_{i,j} + \mathbf{B}\mathbf{Y}_{i,j-1} + \mathbf{C}\mathbf{F} \quad (10)$$

where $\mathbf{Y}_{i,k}$ is the displacement of grid points $i = 1, 2, \dots, n$ of the manipulator at time step k ($k = j + 1, j, j - 1$). \mathbf{A} and \mathbf{B} are constant $n \times n$ matrices whose entries depend on the flexible manipulator specification and the number of sections the manipulator is divided into, \mathbf{C} is a constant matrix related to the time step Δt and mass per unit length of the flexible manipulator and \mathbf{F} is an $n \times 1$ matrix related to the given input torque [2]. Equation (10) thus represents the dynamic equation of the manipulator in the presence of hub-inertia and payload which can easily be implemented on a digital processor.

CONTROL STRATEGIES

The control strategies considered in this work can be divided into open-loop and closed-loop methods. Open-loop control involves altering the shape of actuator commands by considering the physical and vibrational properties of the system. Closed-loop control uses measurements of the system's states and alters the actuator input in order to reduce the system oscillations.

Open-loop control

The main source of the flexible manipulator's vibration is the motion itself. Open-loop control input torque profiles are generated by minimising input energy at system natural frequencies, so that the vibration in the flexible manipulator system is reduced during and after the move. Three types of open-loop shaped input control strategies are developed on the basis of extracting the energies around the natural frequencies. These are, Gaussian shaped input, lowpass filtered torque input and band-stop filtered torque input.

The **Gaussian-shaped input** torque profile developed corresponds to the first derivative of the Gaussian distribution function. The behaviour of the function as an input profile has previously been investigated by adopting a simple method of developing an input torque profile for a flexible manipulator system [9]. This study has looked into the variation of frequency distribution, duty cycle and amplitude of the Gaussian-shaped input torque with various parameters. This enables one to generate an appropriate input trajectory to move the flexible manipulator for a given position

with negligible vibration. The Gaussian-shaped torque input can be written as:

$$\tau(t) = \frac{-(t - \mu)}{\sqrt{2\pi\sigma^3}} \exp\left[-\frac{(t - \mu)^2}{2\sigma^2}\right]$$

where the parameters σ and μ are treated as constants. These can be set for desired cut-off frequency and duty cycle of the torque signal. Figure 2 shows the Gaussian torque input corresponding to a typical set of σ and μ . It is noted that the Gaussian-shaped input torque has a smooth start and stop behaviour. This is important for a vibrationless movement of the system.

The **filtered torque input** approach involves the use of a bang-bang input and filtering out of spectral energy near the natural frequencies of the system. One method to achieve this is to pass the bang-bang input through a lowpass filter. This will attenuate all frequencies above the filter cut-off frequency. An important design point is that the input filtered signal must have a steep roll-off at the cut-off frequency so that energy can be passed at frequencies nearly up to the lowest natural frequency of the flexible manipulator. There are various types of lowpass filter. In this work Butterworth and elliptic filter types are utilised.

The second approach adopted is to use band-stop filters to minimise spectral energy at the natural frequencies of the system. This can be achieved by using a set of bandstop filters, one for each mode, in cascade. The design procedure for band-stop filters is the same as for the lowpass filters; the transfer function of the bandstop filter is calculated using an appropriate low-pass to band-stop transformation. The bandstop strategies are also implemented in this work with Butterworth and elliptic filter types.

Closed-loop control

Two control strategies are developed using closed loop methods. The first method is a collocated proportional and derivative (PD) controller incorporating hub angle and hub velocity feedback. The second method is a hybrid (collocated and non-collocated) controller, comprising two feedback loops, one using the hub angle and hub

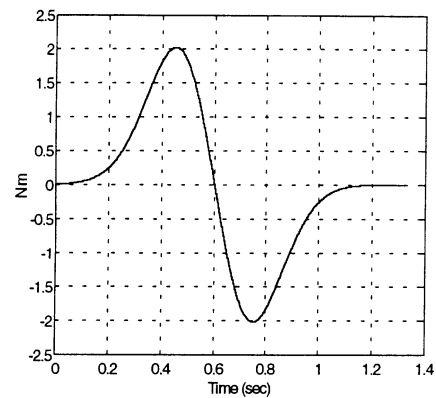


Fig. 2. Gaussian shaped input torque.

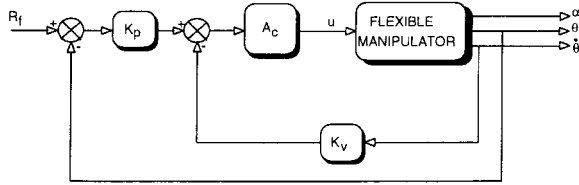


Fig. 3. Flexible manipulator system with joint-based collocated controller.

velocity as inputs to a PD control scheme and the other using the end-point acceleration as input to a PID control scheme.

A block diagram of the **joint-based collocated control PD controller** is shown in Fig. 3, where K_p and K_v are the proportional and derivative gains, θ represents the hub angle, $\dot{\theta}$ represents the hub velocity, $\ddot{\theta}$ represents the end-point acceleration, R_f is the reference hub angle and A_c is the gain of the motor amplifier. Here the motor/amplifier set is considered as a linear gain A_c , as, in practice, the set is found to function linearly in the frequency range of interest. The control signal $u(s)$ in Fig. 3 can be written as:

$$u(s) = A_c [K_p(R_f(s) - \theta(s)) - K_v s\theta(s)]$$

where s is the Laplace variable. The closed-loop transfer function is, therefore, obtained as:

$$\frac{\theta(s)}{R_f(s)} = \frac{K_p H(s) A_c}{1 + A_c K_v (s + K_p/K_v) H(s)}$$

where $H(s)$ is the open-loop transfer function from the input torque to hub angle, given by:

$$H(s) = C(sI - A)^{-1} B$$

where A , B , and C are the characteristic matrix, input matrix and output matrix of the system respectively. The closed-loop poles of the system are, thus, given by the closed-loop characteristic equation as:

$$1 + K_v (s + Z) H(s) A_c = 0$$

where $Z = K_p/K_v$ represents the compensator zero which determines the control performance and characterises the shape of the root-locus of the closed-loop system. The design of the joint based collocated control system involves obtaining suitable values for K_p and K_v so that the system achieves the demanded angular position R_f with reduced flexible motion [3–5].

A block diagram of the **hybrid control** structure, incorporating a combined collocated and non-collocated controller, is shown in Fig. 4. The controller design utilises end-point acceleration feedback through a PID control scheme. Moreover, the hub angle and hub velocity feedback are also used in a PD configuration for control of the rigid body motion of the manipulator. The control structure utilised thus comprises two feedback loops; one using the filtered end-point acceleration

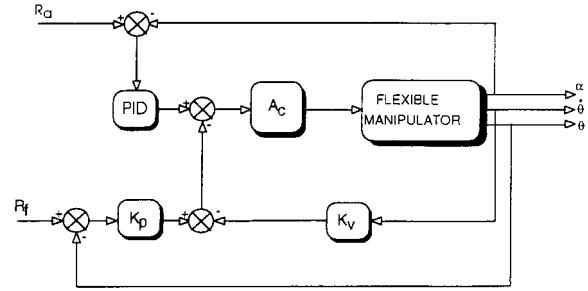


Fig. 4. Block diagram of the hybrid collocated and non-collocated controller.

as input to one control law, and the other using the filtered hub angle and hub velocity as input to a separate control law. These two loops are then summed to give a command motor input voltage, which produces a torque.

Consider first the rigid body control loop, in which the hub angle θ and hub velocity $\dot{\theta}$ are the output variables. To design the controller in this loop, a low-pass filter is required for both θ and $\dot{\theta}$ so that the flexible modes are attenuated before reaching the controller input. The flexible motion of the manipulator is controlled using the end-point acceleration feedback through a PID controller. The end-point acceleration is fed back through a low-pass filter. The values of proportional, derivative and integral gains are adjusted using the Ziegler-Nichols procedure [13, 14].

SIMULINK IMPLEMENTATION

SIMULINK is a program for simulating dynamic systems, as an extension to MATLAB. The major advantage of the SIMULINK environment is the integrated code generator. This code generator can translate SIMULINK diagrams to standard C code. This code can be utilised in special-purpose devices, such as transputers, for real time control applications.

Simulation algorithm

SIMULINK adds many features specific to dynamic systems while retaining all of MATLAB's general-purpose functionality. SIMULINK has two phases of use: model definition and model analysis. In implementing the simulation algorithm the first phase proved to be the most difficult to realise. SIMULINK can easily solve continuous-time (s-domain) and discrete time (z-domain) systems or mixed versions by using a standard library of fundamental items such as integrators, transfer functions that can be implemented either by declaring poles and zeros or coefficients for the numerator and the denominator polynomials. The FD algorithm is a recursive method for calculating the angular displacement of a flexible manipulator system. Three different approaches were investigated to implement the algorithm within SIMULINK.

The initial MATLAB implementation of the

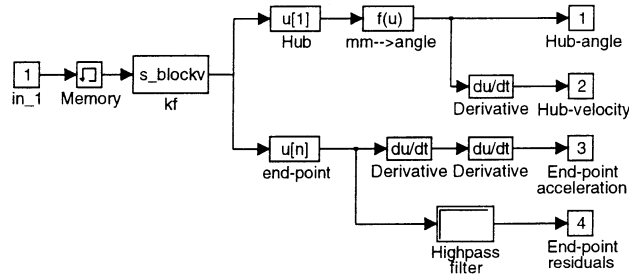


Fig. 5. S-function representation of the flexible manipulator system.

simulation algorithm showed that it was possible to run the algorithm under SIMULINK. The first approach was to utilise the existing MATLAB script files and interface them with the SIMULINK environment. This was achieved by using a block from SIMULINK labelled ‘S-Function’. This makes S-function available as blocks. The named S-function can be any MATLAB M-file. This block allows additional parameters to be passed directly from MATLAB workspace to the named S-function. In the S-function a MATLAB function is called up which implements equation (5). Figure 5 shows the FD model implementation in a block diagram form within SIMULINK. The block labelled “Memory” provides one time-step delay of the torque input signal at the port labelled ‘in-1’. The output of the block labelled ‘kf’ provides the hub-point and end-point displacements from which the corresponding velocity and acceleration are obtained through the derivative blocks. The end-point residuals are calculated from the differences between the displacements of the end of the rigid body and the end-point of the flexible arm displacements. This realisation was found extremely slow when coupled to other SIMULINK elements.

It was attempted to improve the S-function realisation using a simpler approach. This was achieved by using a SIMULINK block as a ‘MATLAB Function’. The MATLAB Function applies the specified function to the input. This realisation is shown in Fig. 6.

The major difference of this realisation with the S-function approach is that, a function is called up directly in the block labelled ‘MATLAB Fcn’. All the system parameters are fed into the same block. Another difference is the need for the use of a clock signal. This improved the performance by 40% in speed. Although this approach is faster, it is still too complicated and difficult to be utilised by any user. Moreover, the implementation of the algorithm in nested loop structure proved to be extremely difficult to debug.

Both MATLAB and SIMULINK can provide reduced processing time if the algorithm can be implemented in **matrix form**. Equation (10) provides the matrix form of the FD algorithm and therefore, can be implemented in SIMULINK by simple blocks provided by the library of the program. A schematic representation of equation (10) using SIMULINK is shown in Fig. 7.

The ‘Memory’ and the feedback loop from ‘out_1’ in Fig. 7 provide the displacements Y_{i-1} and Y_{i-2} of the n grid points. The buffer zero delay is used to break the algebraic loop which SIMULINK has to solve in the first iteration. Matrices A , B and C are calculated using MATLAB script files. This was further developed by replacing the blocks labelled ‘Matrix A’, ‘Matrix B’, and ‘Matrix C’ with a custom MATLAB function performing multiplication of two matrices. This approach proved to be the fastest of all the three approaches and the easiest to modify for any foreseen extensions of the FD algorithm. It requires less data

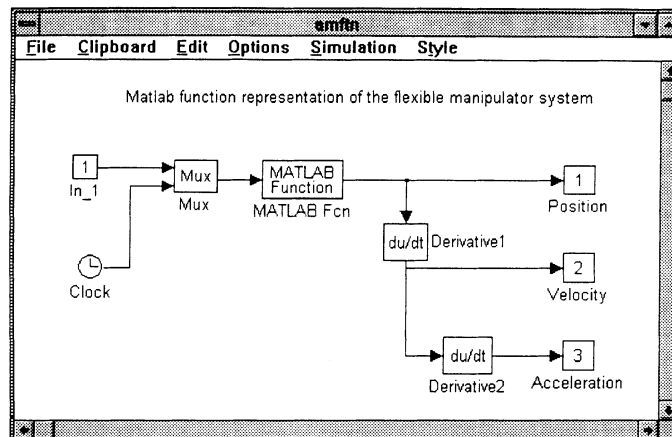


Fig. 6. MATLAB function representation of the flexible manipulator system.

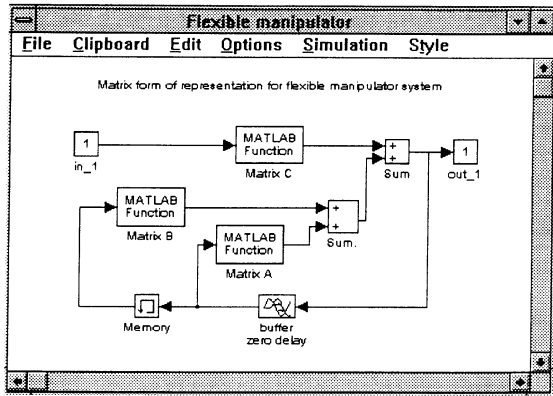


Fig. 7. Matrix form of representation of the flexible manipulator system.

storage, unless the user chooses to store data within MATLAB workspace.

SIMULINK library

For purposes of studying the behaviour of the flexible manipulator and testing the performance of controller designs, a SIMULINK library (ALIBRARY.M) was constructed to help the user to develop his/her own system with various input torque and controllers. In addition to the flexible manipulator system several conventional open-loop and closed-loop controllers, described earlier, are provided. A number of simulation case studies are also provided to demonstrate the potential of the environment.

Figure 8 shows the SIMULINK library for the flexible manipulator system with various related functions. This library is capable of analysing models with variable time-steps within the allowable limits of the parameter c in equation (5). Note that, for the iterative scheme to be stable the value of c must be between 0 and 0.25 [9]. A brief description of the various icons in Fig. 8 is given below.

Flexible Manipulator: Simulates the flexible manipulator with given physical dimensions and characteristics. These can either be specified by the user or the predefined default values

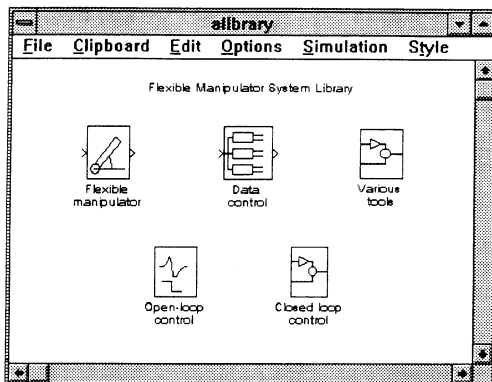


Fig. 8. SIMULINK library for simulation and control of the flexible manipulator system.

Table 1. Default parameters of the flexible manipulator

Parameter	Value
Length	960 mm
Thickness	3.2004 mm
Width	19.008 mm
Area of cross section	$6.08332 \times 10^{-2} \text{ m}^2$
Mass density per volume	2710 Kg/m^3
Young's Modulus	$7.11 \times 10^{10} \text{ N/m}^2$
Area moment of inertia	$5.1923 \times 10^{-11} \text{ m}^4$
Hub inertia	$1.6445 \times 10^{-4} \text{ Kg m}^2$
Manipulator Inertia	0.0486 Kg m^2

given in Table 1 can be used. This uses the matrix form of the FD implementation method.

Data control: Subgroup used to generate displacement, velocity, and acceleration signals for the hub and end-point of the flexible manipulator. As the hub point is pinned the nearest point of the hub represents the hub point information. The input signal for the data control block must be the output signal taken from the block labelled 'Flexible Manipulator'. This will give displacements for all stations along the arm.

Open-loop Control: Subgroup of various shaped torque signals, such as Gaussian-shaped, low-pass and band-stop filter, both analogue and digital type, see Fig. 9. By using these shaped inputs, input torque profiles are developed that minimise input energy at system natural frequencies and thereby reduces flexible manipulator's vibration during and after a movement. Low-pass filters could be used to reduce input energy to the system at all the frequencies including the first natural frequency of the system [2], while band-stop filters can be used to attenuate input energy at frequencies around the natural frequencies of the system. All the filter parameters are user specified according to the response behaviour of the system. The 'Digital filter' block as well as the 'Analog filter' block have four sub-groups.

Closed-loop control: Subgroup of two closed-loop control strategies, namely, a joint-based collocated proportional derivative (PD) controller and a hybrid collocated and non-collocated type controller [5]. This is shown in Fig. 10. The collocated type control incorporates hub position and hub velocity information for the controller design (see Fig. 11). The block

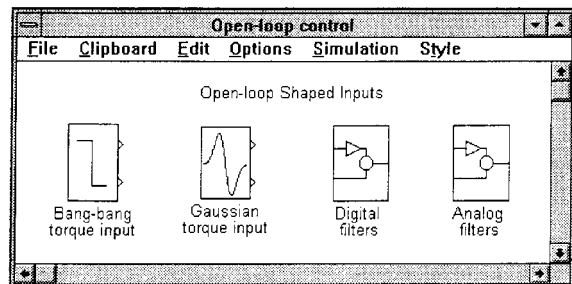


Fig. 9. Library of open-loop shaped inputs.

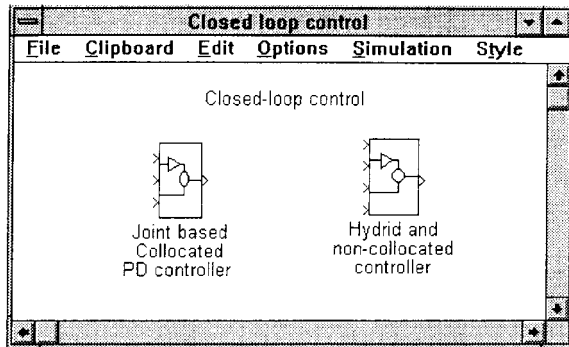


Fig. 10. Closed-loop control strategies for the flexible manipulator system.

labelled 'Position command' generates a hub displacement value in meters from a given command of an angular displacement in degrees. K_p and K_v are the proportional and derivative gains respectively. 'Position feedback' and 'Velocity feedback' are the feedback signals from flexible manipulator system for hub position and hub velocity respectively. Ac is the gain of the motor/amplifier set. This is considered as a linear gain. In the hybrid collocated and non-collocated type, the control structure utilised has two feedback loops: one using the filtered end-point acceleration as input to a control law, and the other using the hub displacement and hub velocity as input to a separate control law. These two loops are then summed and the result passed through the amplifier/motor gain to produce torque input to the manipulator. As in the previous case the motor/amplifier behaviour is considered to be linear. The first loop which comprises the hub displacement and hub velocity signal is considered as the rigid body motion control loop. The flexible motion of the manipulator is controlled by using the end-point acceleration feedback through a PID controller. The values of proportional, derivative and integral gains as indicated earlier, are adjusted using the Ziegler-Nichols tuning method [13, 14]. The hybrid control block, shown in Fig. 12, is constructed by approximating the derivative with the first-order difference form and the integration using the trapezoidal rule.

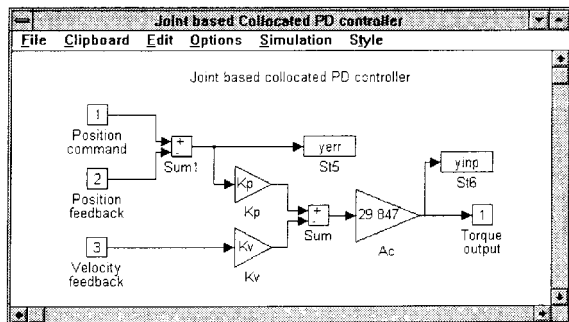


Fig. 11. Connection diagram of joint-based collocated controller.

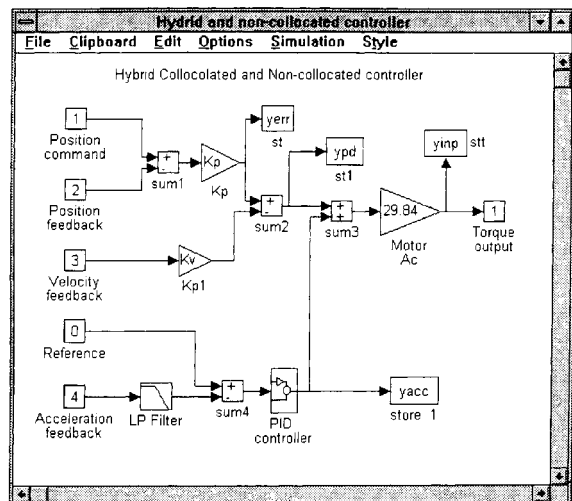


Fig. 12. Connection diagram of the hybrid collocated and non-collocated controller.

Various tools: Subgroups containing frequently used blocks needed for developing flexible manipulator systems in various configurations. These are shown in Fig. 13.

CASE STUDIES

A number of case studies are presented in this section to demonstrate the utilisation of the SIMULINK library. These will help the user to understand the behaviour of the flexible manipulator system with user-defined specifications as well as visualisation of the effectiveness of controllers on the performance of the manipulator. The controllers tested are included in the flexible manipulator library. Users, however, can develop and test their own controllers using this environment. The environment developed runs under version 4.2 of MATLAB with version 1.2 of SIMULINK or above. It is assumed that the

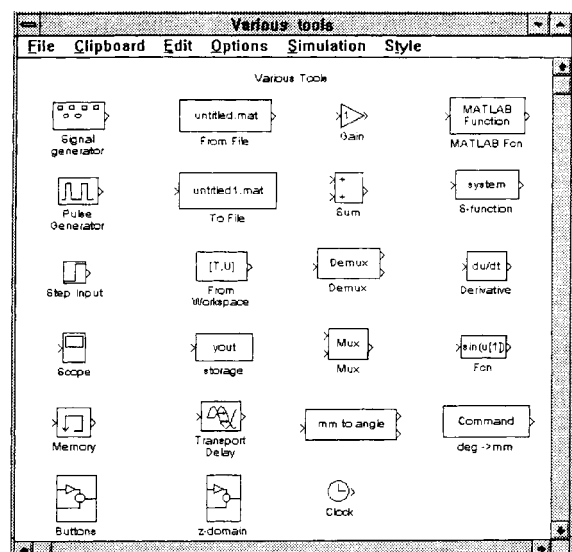


Fig. 13. Commonly used blocks utilised for developing flexible manipulator systems.

reader is familiar with these MATLAB and SIMULINK packages [15, 16].

Case study 1: open-loop response of the flexible manipulator

To use the environment for development of a system, it is required to be in the directory where the simulation library (**alibrary**) resides. At this level, by entering the command **SIMULINK** at the MATLAB prompt the main block library opens. This command displays a new window containing icons for the sub-system blocks that make up the standard library. By clicking on the SIMULINK window, and selecting **Open . . .** from the **File** menu on its menu bar the flexible manipulator simulation library (**alibrary**) opens. Choosing 'ALIBRARY.M' and clicking OK at this level will display a SIMULINK window with icons of the flexible manipulator simulation library as shown in Fig. 8. To develop a new system call a new SIMULINK window by selecting **New . . .** from **File** menu. Drag the required blocks from library (ALIBRARY.M) to the new SIMULINK window and develop the system. In this case, for example 'Bang-bang torque' input from 'Open-loop control' block, 'Flexible manipulator', 'Data control' and 'To workspace' from 'Various Tools' can be placed to the new window and wired one after another as shown in Fig. 14. This working model can be saved on disk as a MATLAB M-file through **Save . . .** in **File** menu and named as 'abang.m'. This M-file contains all the commands necessary to describe the developed flexible manipulator system with bang-bang input torque. One could run this file to observe the behaviour of the flexible manipulator with bang-bang input. To allow this, **Parameters . . .** of each of the block can be selected from **Simulation**, if necessary, and then default values of **Stop time . . .** and both **Max step size . . .** and **Min step size . . .** with value of time step of the system under consideration can be replaced with user specified values.

To run the developed system abang.m it is required to calculate some initial parameters. These can be done by running a program **aset.m** at the MATLAB prompt. This enables to set the manipulator specifications and boundary conditions, such as hub inertia and payload conditions in an interactive manner. Following the setting of

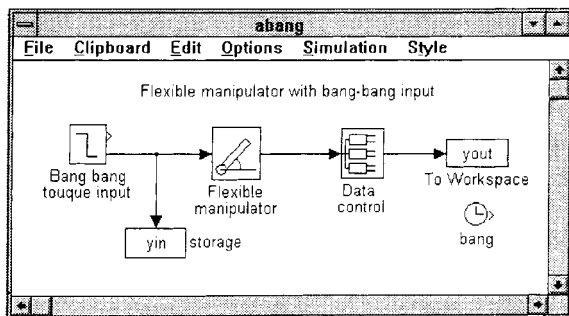


Fig. 14. Flexible manipulator with bang-bang input torque.

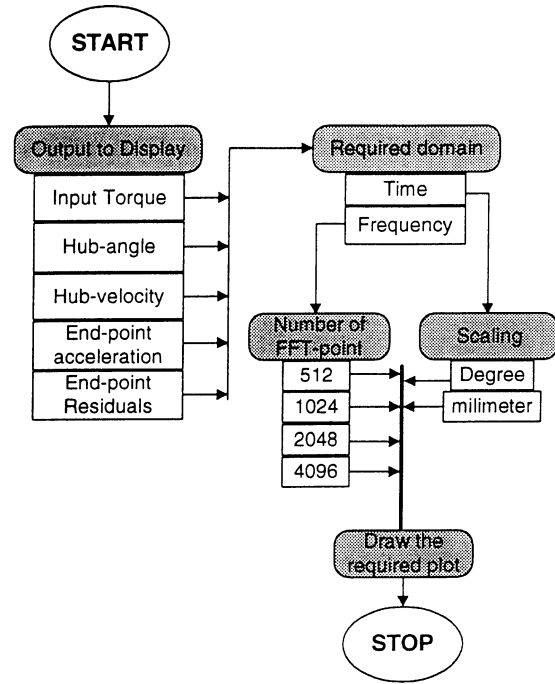


Fig. 15. Display of inputs and outputs after simulation.

initial values, simulation of the developed system could be performed by choosing the **Start . . .** command in the **Simulation** menu. The input and outputs are stored in the MATLAB workspace. These can subsequently be used for monitoring and documentation. Stored input and outputs of the system and their spectra can be observed by entering **display** at the MATLAB prompt. This will guide the user through various options (see Fig. 15) and display the required output in a MATLAB figure window. System input and various outputs from the above simulated system and their spectral analysis obtained through this process are shown in Figs 16 to 25.

Case study 2: flexible manipulator with filtered input torque

This example demonstrates the utilisation of band stop filtered bang-bang input torque with

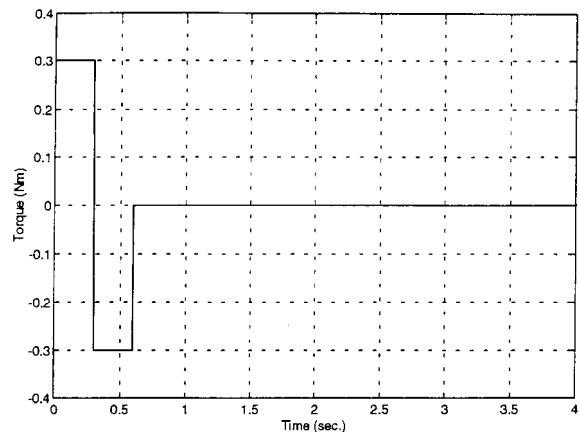


Fig. 16. Bang-bang torque input to the flexible manipulator system.

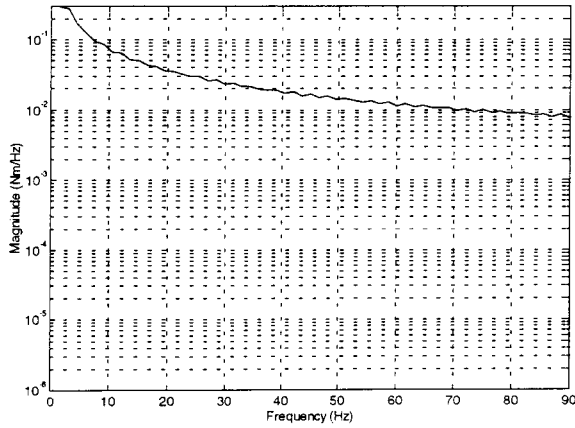


Fig. 17. Spectral density of the bang-bang torque input.

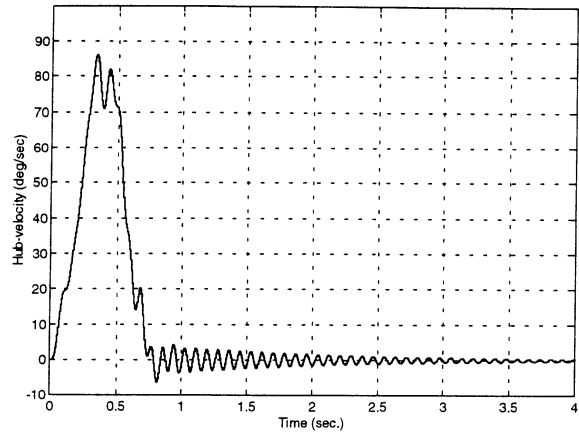


Fig. 20. Hub velocity with bang-bang torque input.

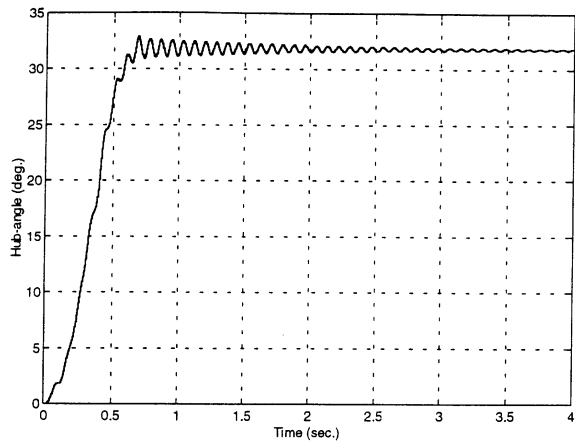


Fig. 18. Hub displacement with bang-bang torque input.

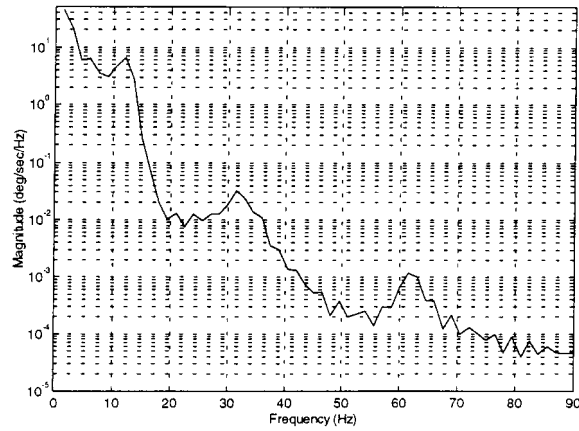


Fig. 21. Spectral density of hub velocity signal with bang-bang torque input.

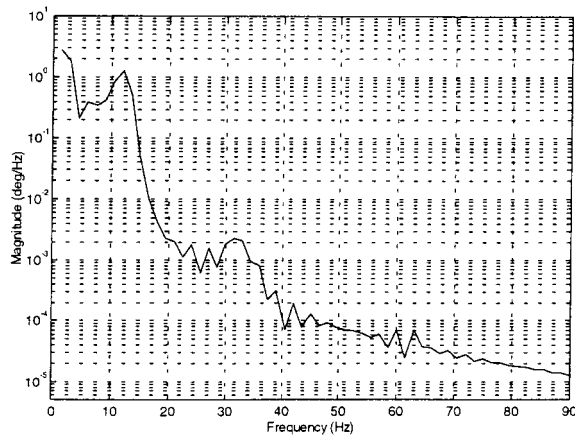


Fig. 19. Spectral density of hub displacement signal with bang-bang torque input.

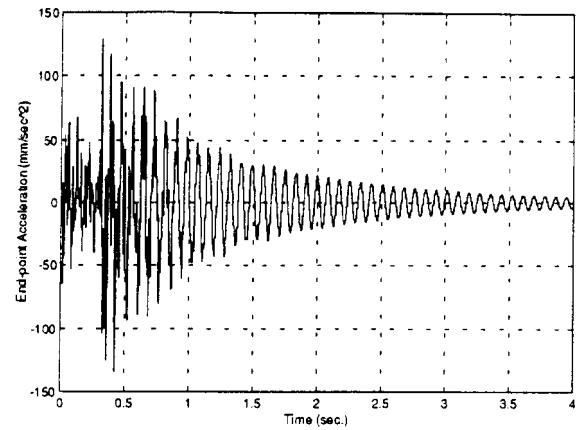


Fig. 22. End-point acceleration with bang-bang torque input.

the flexible manipulator system. Figure 26 shows the developed system using blocks from ALIBRARY.M using similar steps as described above. The purpose of this type of input is to provide vibrationless motion of the system by filtering out the spectral energy near the natural modes. The first three modes are considered in this case with three Butterworth filters, each of 4th order and a stop band of 5 Hz around the natural

frequencies of the manipulator. The first three natural frequencies are at 14.2 Hz, 32.3 Hz and 62.2 Hz respectively. These frequencies are inherent to the behaviour of the system and depend on the physical properties and characteristics of the flexible manipulator. The effect of blocking spectral energy around the natural modes of the flexible manipulator system is clearly shown in the corresponding responses in Figs 27 to 36.

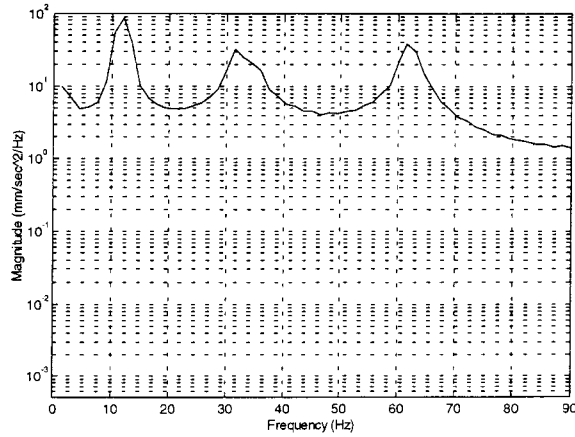


Fig. 23. Spectral density of the end-point acceleration with bang-bang torque input.

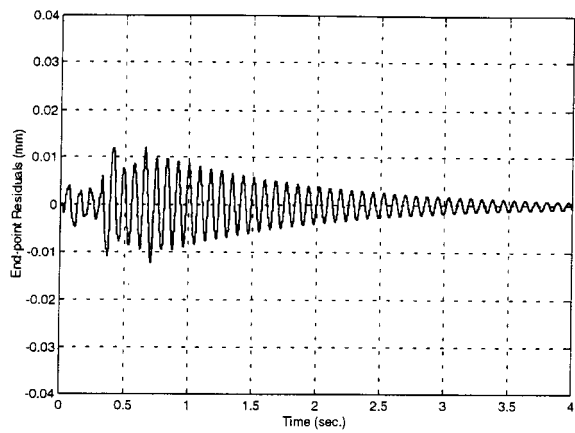


Fig. 24. End-point residual motion with bang-bang torque input.

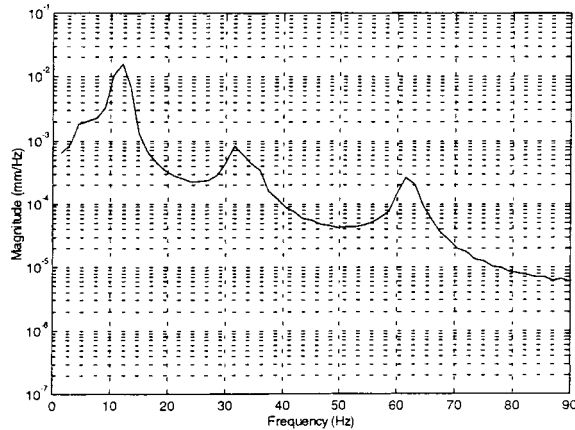


Fig. 25. Spectral density of end-point residual motion with bang-bang torque input.

Case study 3: closed loop (hybrid) control

This case study demonstrates the utilisation of hybrid collocated and non-collocated controller with the flexible manipulator system. Figure 37 shows the developed system using blocks from ALIBRARY.M in a similar manner as described above. The control structure utilised has two feedback loops: one using the filtered end-point acceleration as input to a control law, and the other

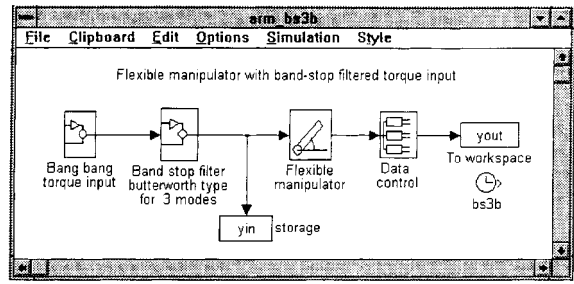


Fig. 26. Flexible manipulator with band-stop filtered input torque (first three modes).

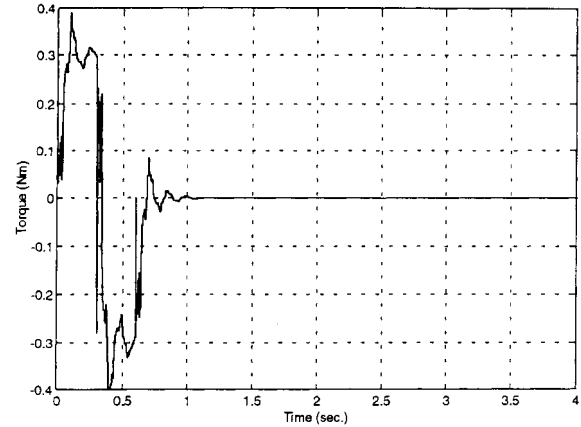


Fig. 27. Input torque with bandstop filtered (Butterworth) input torque for the first three modes.

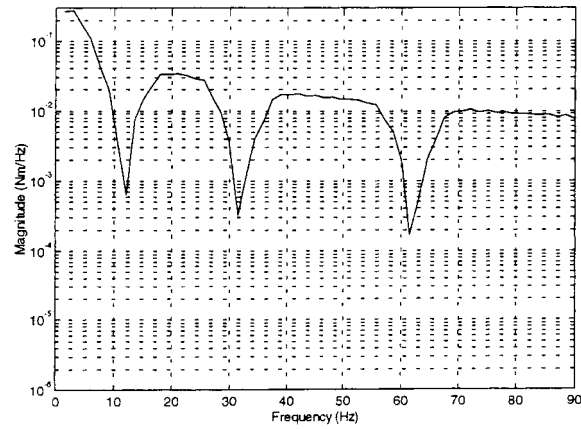


Fig. 28. Spectral density of input torque with bandstop filtered (Butterworth) input torque for the first three modes.

using the hub displacement and hub velocity as input to a separate control law. These two loops are then summed and passed through the amplifier/motor gain to produce torque input to the manipulator (see Fig. 12). The performance of the system with this type of control is demonstrated in Figs 38–47.

CONCLUSION

The development of an interactive and user-friendly environment for simulation and control of

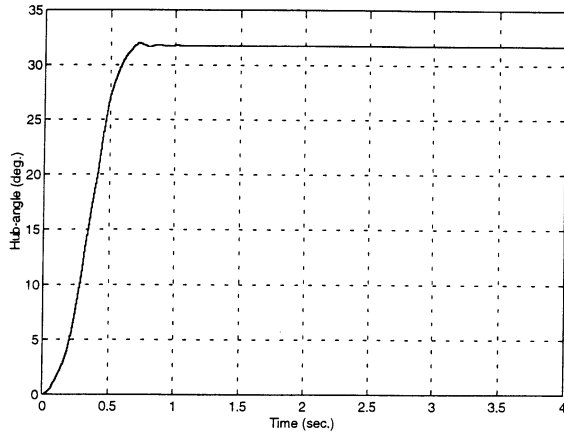


Fig. 29. Hub displacement with bandstop filtered (Butterworth) input torque for the first three modes.

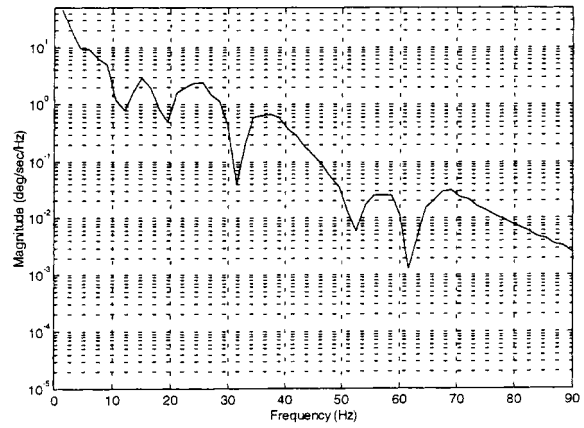


Fig. 32. Spectral density of hub velocity with bandstop filtered (Butterworth) input torque for the first three modes.

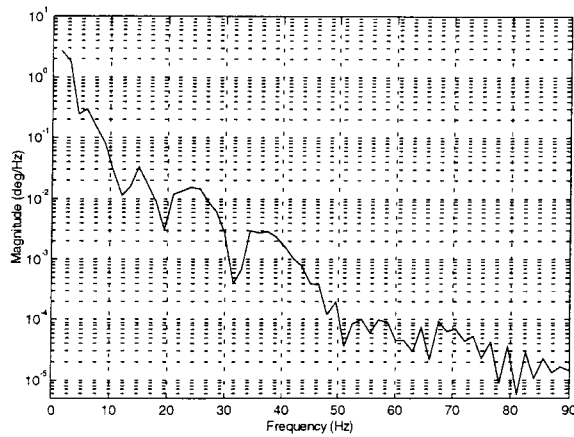


Fig. 30. Spectral density of hub displacement with bandstop filtered (Butterworth) input torque for the first three modes.

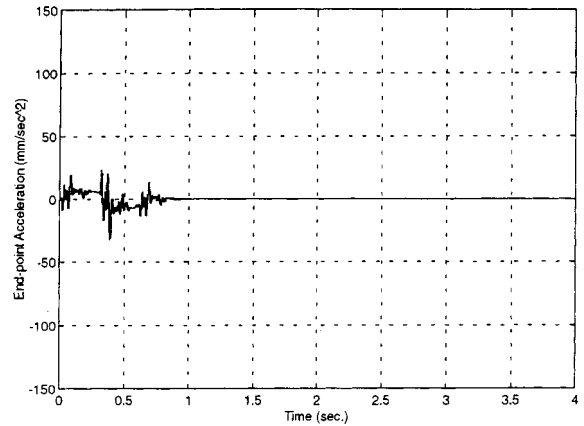


Fig. 33. End-point acceleration with bandstop filtered (Butterworth) input torque for the first three modes.

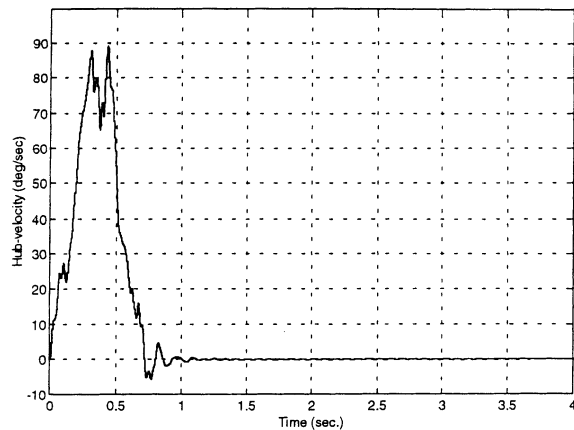


Fig. 31. Hub velocity with bandstop filtered (Butterworth) input torque for the first three modes.

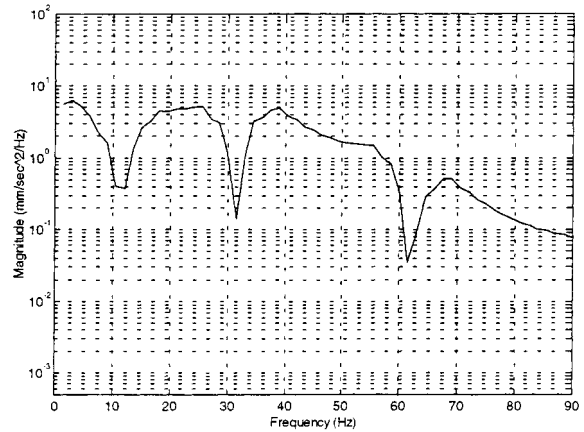


Fig. 34. Spectral density of end-point acceleration with bandstop filtered (Butterworth) input torque for the first three modes.

flexible manipulator systems has been presented. A finite-difference simulation of a single-link flexible manipulator has been developed and realised within SIMULINK. Various open-loop and closed-loop control strategies have been developed for the system and incorporated into a library within the environment. The use of the developed library has also been demonstrated through several selected case studies.

This environment has proven to be a valuable education tool for understanding the behaviour of flexible manipulator systems and development of various controller designs. In this manner the package can easily be used as a computer-aided teaching facility for analysis and design of control systems for flexible manipulators. The environment can also be used as a testbed for newly designed controllers, where

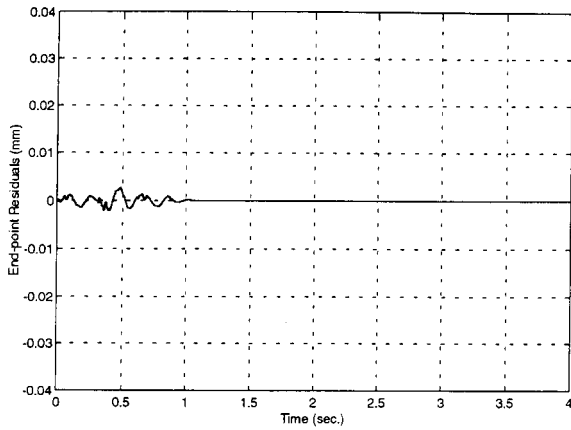


Fig. 35. End-point residual movement with bandstop filtered (Butterworth) input torque for the first three modes.

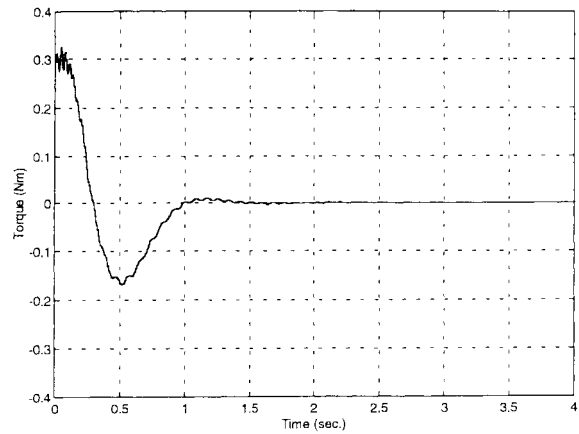


Fig. 38. Input torque with hybrid collocated non-collocated controller.

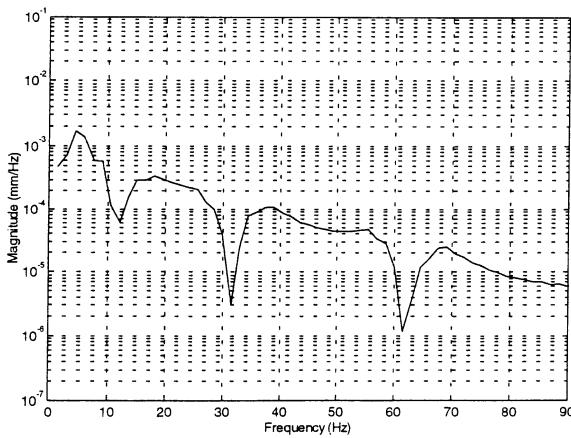


Fig. 36. Spectral density of end-point residual movement with bandstop filtered (Butterworth) input torque for the first three modes.

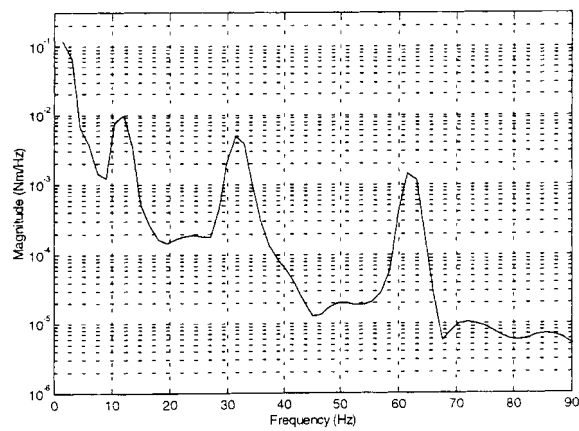


Fig. 39. Spectral density of input torque with hybrid collocated non-collocated controller.

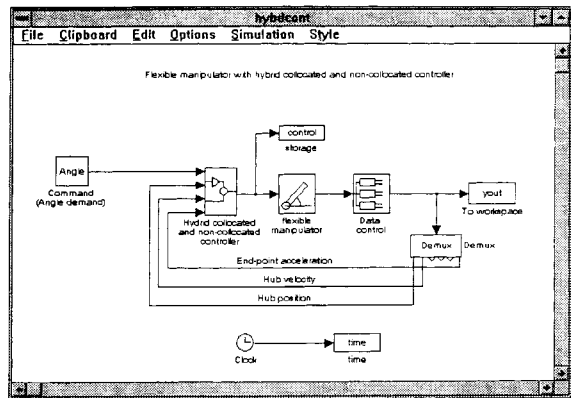


Fig. 37. Flexible manipulator system with hybrid collocated and non-collocated controller.

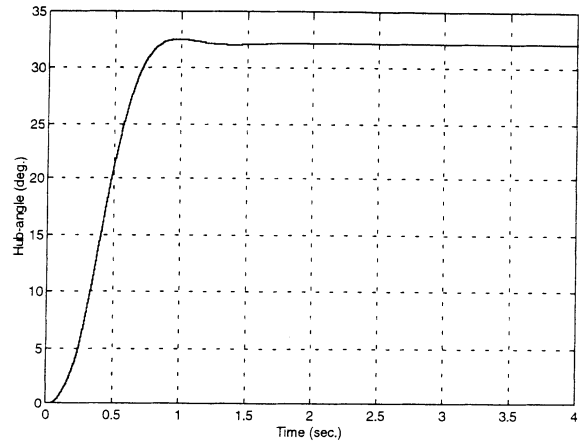


Fig. 40. Hub displacement with hybrid collocated non-collocated controller.

control engineers and researchers could test their controllers for vibration and positioning control of the flexible manipulator and study their performance.

The SIMULINK blocks can be utilised to investigate various other aspects of vibration control in flexible manipulator systems. Users can design their own SIMULINK blocks and couple them with specific requirements. In this

manner, the incorporation of closed-loop control methods of a fixed and adaptive nature are currently being investigated by the authors. In addition to this, a data analysis provision is also provided within the package, to enable users to analyse their data obtained from a test run. This makes the environment more user friendly and saves the time and effort to transfer the data to another environment for analysis.

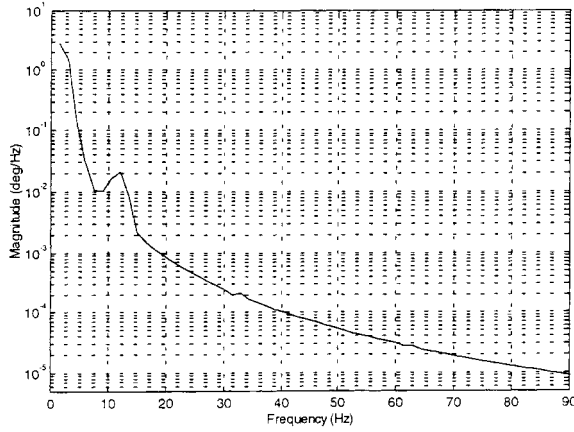


Fig. 41. Spectral density of hub-displacement with hybrid collocated non-collocated controller.

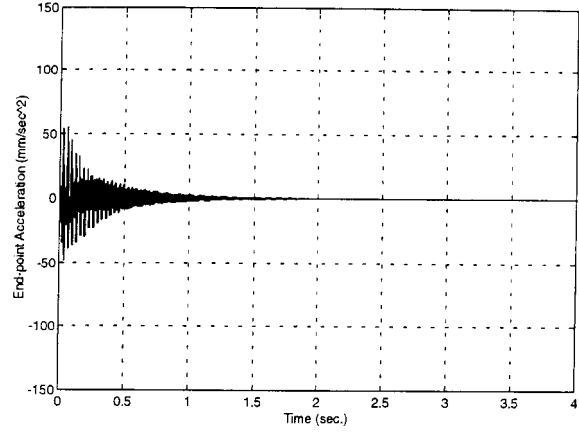


Fig. 44. End-point acceleration with hybrid collocated non-collocated controller.

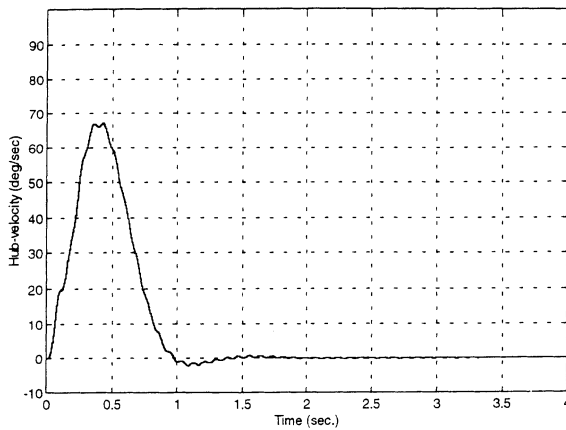


Fig. 42. Hub-velocity with hybrid collocated non-collocated controller.

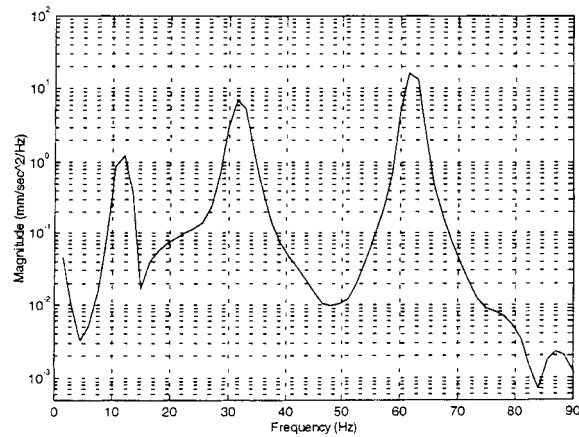


Fig. 45. Spectral density of end-point acceleration with hybrid collocated non-collocated controller.

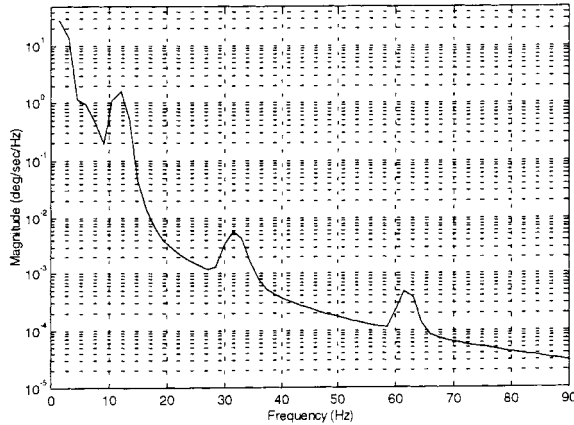


Fig. 43. Spectral density of hub-velocity with hybrid collocated non-collocated controller.

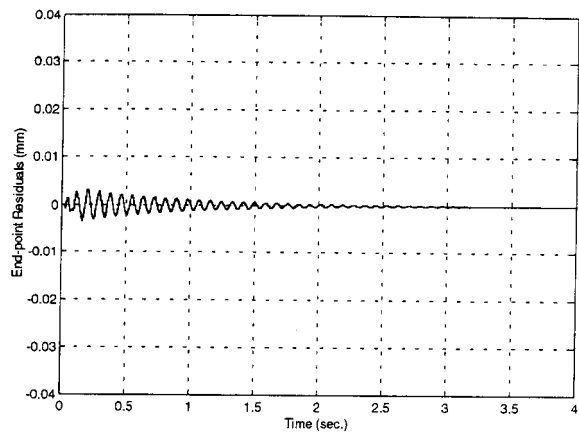


Fig. 46. End-point residual movement with hybrid collocated non-collocated controller.

REFERENCES

1. W. J. Book and M. Majette, Controller design for flexible distributed parameter mechanical arms via combined state-space and frequency domain techniques, *Trans. ASME Journal of Dynamic Systems, Measurement and Control*, **105**, (1983) pp. 245–254.
2. M. O. Tokhi, H. Poerwanto and A. K. M. Azad, Dynamic simulation of flexible manipulator systems incorporating hub inertia, payload and structural damping, *Machine Vibration*, **4**, (1995) pp. 106–124.
3. M. O. Tokhi and A. K. M. Azad, Active vibration suppression of flexible manipulator systems—closed-loop control methods, *Int. J. Active Control*, **1**, (1995) pp. 79–107.

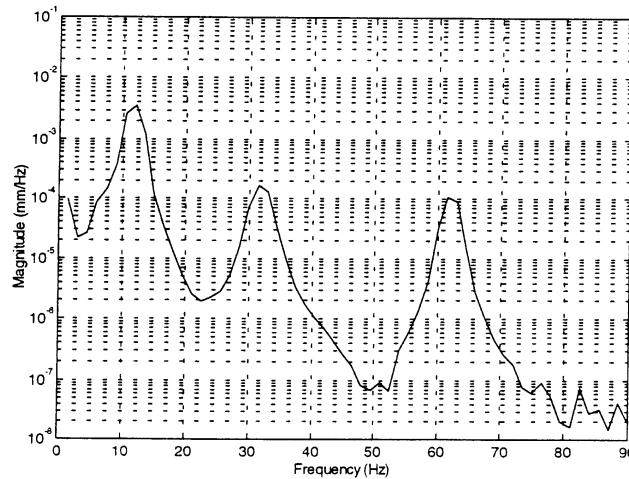


Fig. 47. Spectral density of end-point residual movement with hybrid collocated non-collocated controller.

4. M. O. Tokhi and A. K. M. Azad, Control of flexible manipulator systems, *Proc. IMechE-I: J. Systems and Control Engineering*, **210**, (1996) pp.113–130.
5. M. O. Tokhi and A. K. M. Azad, Collocated and non-collocated feedback control of flexible manipulator systems, *Machine Vibration*, **5**, (1996) pp.170–178.
6. S. Choura, Reduction of residual vibration in a rotating flexible beam with a moving payload mass, *Proc. IMechE—I: Journal of Systems and Control Engineering*, **211**, (1997) pp.25–34.
7. F. S. Tse, I. E. Morse and T. R. Hinkle, *Mechanical Vibrations Theory and Applications*, Allyn and Bacon Inc. (1980).
8. P. K. Kourmoulis, *Parallel processing in the simulation and control of flexible beam structures*, Ph.D. Thesis, University of Sheffield, Department of Automatic Control and Systems Engineering (1990).
9. A. K. M. Azad, *Analysis and design of control mechanisms for flexible manipulator systems*, Ph.D. Thesis, University of Sheffield, Department of Automatic Control and Systems Engineering (1994).
10. R. L. Burden and J. D. Faires, *Numerical Analysis*, PWS-KENT Publishing Company, Boston (1989).
11. G. S. Hastings and W. J. Book, A Linear dynamic model for flexible robotics manipulator, *IEEE Control Systems Magazine*, **7**, (1987) pp.61–64.
12. V. V. Korolov and Y. H. Chen, Controller design robust to frequency variation in a one-link flexible robot arm, *J. Dynamic Systems, Measurement and Control*, **111**, (1989) pp.9–14.
13. K. Warwick, *Control Systems: An Introduction*, Prentice-Hall Inc., UK (1989).
14. K. Ogata, *Modern Control Engineering*, Prentice-Hall Inc., Englewood Cliffs (1990).
15. The Mathworks Inc, *MATLAB for Sun Workstations, Users Guide*, The Mathworks Inc. Natwick (1991).
16. The Mathworks Inc, *SIMULINK, Users Guide (for X Window System)*, The Mathworks Inc. Natwick (1992).

Osman Tokhi obtained his B.Sc. (electrical engineering) from Kabul University (Afghanistan) in 1978 and Ph.D. (control engineering) from Heriot-Watt University (UK) in 1988. He has worked as lecturer in Kabul University and Glasgow College of Technology (UK) and as sound engineer in industry. He is currently employed as senior lecturer in the Department of Automatic Control and Systems Engineering, The University of Sheffield (UK). His research interests include active noise and vibration control, real-time signal processing and control, parallel processing, system identification and adaptive/intelligent control. He is a Chartered Engineer, Corporate Member of the IEE, Senior Member of the IEEE and Member of the IIAV.

A. K. M. Azad obtained his Diploma (electronics engineering), B.Sc. and M.Sc. in Applied Physics and Electronics, all from the University of Dhaka (Bangladesh) in 1978, 1984 and 1986 respectively and Ph.D. (control engineering) from the University of Sheffield (UK) in 1994. He has worked as electronics engineer in industry, lecturer at the University of Dhaka and Research Fellow at the University of Sheffield. He is currently a Research Associate at the University of Portsmouth (UK). His current research interests include mechatronics, modelling and simulation of engineering systems and control. He is an Associate Member of the IEE, Member of the IEEE and Senior Member of the ISA.

H. Poerwanto obtained his M.Sc. (control systems) in 1994 and Ph.D in 1998 both from the Department of Automatic Control and System Engineering, University of Sheffield, UK. His research interests include modelling, simulation and control of flexible robot manipulators.