Design of Command Shaper using Gain-delay Units and Particle Swarm Optimisation Algorithm for Vibration Control of Flexible Systems

M. S. Alam

Department of Applied Physics, Electronics and Communication Engineering, The University of Dhaka, Bangladesh

M. O. Tokhi†

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

(Received 19 November 2006; accepted 29 June 2007)

Conventional command shaping methods involve convolving a desired command with a sequence of impulses that may prove computationally expensive or unsuitable for online applications. Moreover, *a priori* knowledge of the system parameters, such as resonance frequencies and associated damping ratios, is required to design the exact sequence of impulses to produce a command that results in zero residual vibration. A new command shaping method is proposed in this paper using gain and delay units. Assuming that, no prior information is available about the system, a new variant of particle swarm optimisation (PSO) algorithm is proposed and used to derive the gain and delay values. The effect of the total number of delay and gain units is also analysed. Moreover, an adaptive control strategy is developed based on the proposed command shaping technique where the PSO is used to adjust the controller parameters online. A twin rotor (experimental flexible) system is used for assessment and evaluation of the control strategy. The effectiveness of the technique is assessed both in the time domain and the frequency domains.

† Member of the International Institute of Acoustics and Vibration (IIAV)

1. INTRODUCTION

Flexible systems are lighter, faster and less expensive than rigid ones, but they pose various challenges as compared to their rigid counterparts. These include system design, vibration control and structural optimisation. The advantages of such systems are often marred by the structural vibration originating, mainly, from system flexibility while in operation. In order to achieve high-speed and accurate positioning, it is necessary to control the system's vibratory response in a cost-effective manner. Numerous techniques have been proposed to effectively control flexible systems. These can be divided into two broad categories, namely feedback control and feedforward control. A good literature review can be found in reference.[1]

A number of feedforward control approaches have been proposed for flexible systems. Command shaping based on various filtering techniques such as low-pass, band-stop and notch filters have been reported and proved to be effective in practical systems.[2] In this approach, the input torque is modified or shaped so that it contains as little as possible energy at and around the natural frequencies of the system. A feedforward control scheme based on the input command shaping, introduced by Singer and Seering,[3] has been applied to the control of different types of flexible systems for vibration reduction or trajectory tracking or occasionally both.[4,5]

Singh and Vadali[6] presented a method to minimise residual vibration of structures or lightly damped servomechanisms using multiple time delays in conjunction with a proportional part. In order to increase robustness of the controller, they included a basic single time-delay control unit in cascade. In another work, the authors presented a design procedure of open-loop controllers to reduce residual vibration in flexible structures using a multiple step inputs delayed in time.[7] The controller attenuated the residual vibration by cancelling the complex poles of the system and robustness was achieved by locating additional zeros at the cancelled poles of the system. The paper also investigated a design procedure of robust time-delay controllers for multiple modes with user selected time delays. Moreover, a design method for the minimum time-delay controller was also proposed where step input magnitude values were constrained within 0 and 1.[7,8] A single-link flexible arm robot was used to illustrate the effectiveness of the controller.

Bodson[9] presented an algorithm for the tuning of two input shaping methods to prevent the excitation of oscillatory modes in resonant systems. Efforts have been made to make a command shaper to adapt to unknown system parameters using adaptive filters,[10] but conventional adaptive algorithms have the drawback of local minima which may result in poor vibration reduction. Alam et al.[11] investigated the potential of genetic algorithms (GAs)[12] in designing uni-modal command shapers for a twin rotor system that resulted in a significant reduction in vibration at the cost of introducing a long response delay. The command shaping technique, in practice, causes delay in the system's response while it reduces vibration and the amount of reduction in vibration and rise time are often in conflict with one another. The weighted sum of these two competing objectives was selected heuristically and used to reduce system vibration as well as to obtain a satisfactory response rise-time.[13] The success of the approach depends on suitable selection of the weight vector, for which prior knowledge of the system characteristics is required. As an alternative, a multi-objective genetic algorithm was employed to design a command shaper where the algorithm provided a wide range of solutions that tradeoff between these two conflicting objectives.[14]

Particle swarm optimisation (PSO)[15,16] is one of the relatively new bio-inspired computation techniques, which has become increasingly popular mainly due to its simplicity, low memory requirements, low computational cost, faster convergence and good overall performance.[17] Since its introduction, PSO has been successfully used as an optimisation technique in a variety of applications, such as training neural networks,[18] modelling of dynamic systems,[19] tuning and designing controllers for power systems[20] and optimising biological and biomechanical systems.[21]

Although a large volume of work involving PSO has been reported in recent years, the literature is rather poor in the use of PSO for designing controllers for vibration reduction in flexible systems. Alam et al.[22] presented a new method in designing command shapers where PSO was employed to derive amplitude and time locations of impulses which, later on, were convolved with a reference input to form the shaped signal. The attempted applications clearly reveal the potential of PSO in control engineering and provide motivation to researchers to adopt PSO for designing vibration control systems. This paper presents a new command shaping method using gain and delay units where a new variant of the PSO algorithm is developed and used to optimise the gain and delay values so as to reduce vibration of a flexible system. The effect of the total number of delay and gain units is also analysed. Moreover, an adaptive control strategy is developed based on the proposed command shaping technique where PSO is used to adjust the controller parameters online. The control strategy is applied on a scaled and simplified version of a practical helicopter, namely a twin rotor multi-input multi-output system (TRMS).[23]

The paper is organised as follows: Section 2 gives a description of the experimental set-up; Section 3 describes the PSO algorithm; Section 4 describes the proposed command shaping approach. Implementation of the strategy is provided in Section 5; the effect of the number of gain and delay units in designing the command shaper is presented in Section 6; and an adaptive control strategy based on the proposed command shaping technique is presented in Section 7, and Section 8 provides conclusions of this work.

2. EXPERIMENTAL SET-UP

Rotary wing aerial vehicles, such as helicopters, have distinct advantages over conventional fixed wing aircraft in surveillance and inspection tasks. A scaled and simplified version of a practical helicopter, namely a twin rotor multi-input, multi-output system (TRMS), as shown in Fig. 1, is often used for aerodynamic modelling[19,24] and control problems[11,22,24] due to its size, cost, ease of operation and interfacing facilities with a personal computer. The TRMS consists of a beam pivoted on its base in such a way that it can rotate freely in both its horizontal and vertical planes producing two rotating movements around yaw and roll axes, respectively. The schematic diagram of the experimental TRMS[23] is shown in Fig. 2. There are two propellers driven by DC-motors, one at each end of a beam and pivoted on a base. The controls of the system are the supply voltages to the motors. A change in the voltage value results in a change in the rotational speed of the propeller, which in turn results in a change in the corresponding position of the beam.

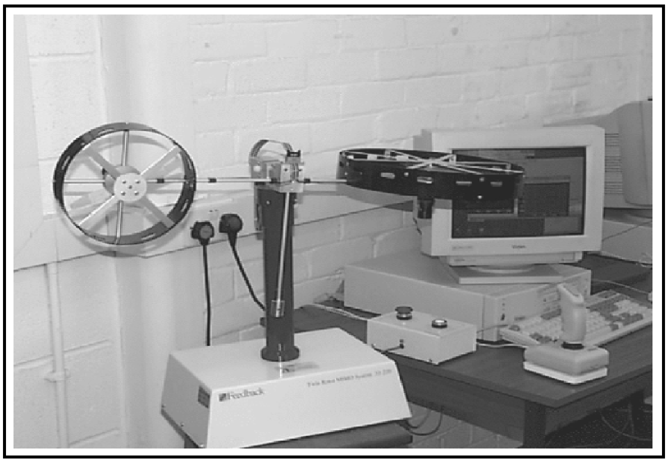

Figure 1. The twin rotor MIMO system.

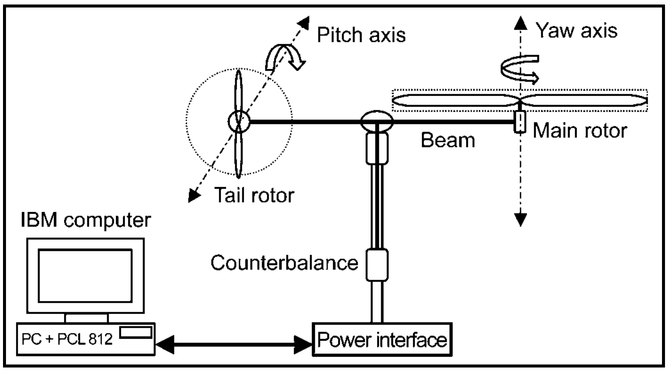

Figure 2. Schematic diagram of TRMS.

The system is interfaced with a personal computer through a data acquisition board, PCL-812PG. The measured signals are: angular positions of the beam in the horizontal and vertical planes and angular velocities of the rotors. The angular velocities of the beam are obtained through the software by differentiating and filtering the measured position angles of the beam. The flexible motion of the system, due to asymmetrical mass distribution of the TRMS system, causes structural vibration while in operation. Moreover, when the rotors move, the rig structure undergoes deflection in the horizontal or vertical or both directions, due to aerodynamic forces, and as a result it may prove difficult to track a desired trajectory. Furthermore, once the system has reached a set point, the residual vibration will degrade the positioning accuracy and may cause a delay in the system's response. The command shaping technique has widely been employed in aircraft[25] and helicopter control.[26]

As far as vibration control of the system is concerned, the vertical channel (motion in the vertical plane) poses more challenges compared to horizontal channel due to the higher physical diameter of the main rotor and higher aerodynamic force. Operation of the TRMS in the vertical plane resembles the behaviour of a practical helicopter in the hovering mode, which is vital for a variety of flight missions such as load delivery, air-sea rescue, etc. Accordingly, this mode of operation of the TRMS is considered in this paper.

3. PARTICLE SWARM OPTIMISATION

PSO is a population-based search algorithm. The algorithm is initialised with a population of random solutions, called particles, and particles fly through the search space with velocities, which are dynamically adjusted according to their historical behaviours. The original PSO algorithm is described as:[15,16]

$$v_{id} = v_{id} + c_1 r(p_{id} - x_{id}) + c_2 R(p_{gd} - x_{id}); \tag{1}$$

$$x_{id} = x_{id} + v_{id}, \tag{2}$$

where c_1 and c_2 are positive constants, r and R are two random functions in the range $[0,1]$; $X_i = (x_{i1}, x_{i2}, ..., x_{id})$ represents the i-th particle; $P_i = (p_{i1}, p_{i2}, ..., p_{id})$ represents the best previous position (the position giving the best fitness value) of the i-th particle; the symbol g represents the index of the best particle among all the particles in the population; and $V_i = (v_{i1}, v_{i2}, ..., v_{id})$ represents the rate of the position change (velocity) for particle i. The fitness of each particle is then evaluated according to a user defined objective function. At each generation, the velocity of each particle is calculated according to Eq. (1) and the position for the next function evaluation is updated according to Eq. (2). Each time, if a particle finds a better position than the previously found best position then its location is stored in the memory.

The first new parameter added into the original PSO algorithm is the inertia coefficient, ω, which is used to balance between the global and local search abilities.[27,28] The introduction of the inertia weight also eliminates the requirement of carefully setting the maximum velocity V_{max}. Equation (1) is thus modified as:

$$v_{id} = \omega v_{id} + c_1 r(p_{id} - x_{id}) + c_2 R(p_{gd} - x_{id}). \tag{3}$$

3.1. Proposed Variant of PSO: A Global Version of PSO With Fitness Sharing Based Replacement Strategy

The commonly used PSOs are either a global or local version of the PSO.[29] The global version (gbest) of the PSO is relatively simpler, faster and requires less computation compared to the local (lbest or pbest) model, but the particles may lose diversity after a certain number of generations.[29,30] As a result, the searching process may get trapped at local minima, especially in the case of multi-modal problems. In order to maintain diversity in the swarm (population), a fitness sharing based replacement strategy is introduced within the gbest model. The algorithm works as a conventional gbest version of the PSO with time-varying inertia coefficient, ω, and constant acceleration coefficients, c_1 and c_2. After a certain number of generations, the shared fitness of each solution is calculated as described below.

Calculation of shared fitness value. Fitness sharing method[31] lowers each population element's fitness by an amount nearly equal to the number of similar individuals in the population. Typically, the shared fitness f_i' of an individual i with fitness f_i is simply:[31,32]

$$f_i' = \frac{f_i}{m_i}, \tag{4}$$

where m_i is the niche count which measures the approximate number of individuals with whom the fitness f_i is shared. The niche count is calculated by summing a sharing function over all members of the population;[31,32]

$$m_i = \sum_{j=1}^{N} sh(d_{ij}), \tag{5}$$

where N denotes the population size and d_{ij} represents the distance between individuals i and j. For phenotypic sharing, the Euclidean distance between two decision variable vectors $X^{(i)}$ and $X^{(j)}$ of an n-dimensional problem can be calculated as d_{ij}:[32]

$$d_{ij} = \sqrt{\sum_{k=1}^{n} \left(x_k^{(i)} - x_k^{(j)}\right)^2}. \tag{6}$$

If normalised distance values are used, the following normalised sharing parameter value can be employed:[32]

$$d_{ij} = \sqrt{\sum_{k=1}^{n} \left[\left(x_k^{(i)} - x_k^{(j)}\right) / \left(x_k^{(U)} - x_k^{(L)}\right) \right]^2}. \tag{7}$$

The sharing function, $sh(d_{ij})$, is given as:[32]

$$sh(d_{ij}) = \begin{cases} 1 - \left(\frac{d}{\sigma_s}\right)^{\alpha}, & \text{if } d \leqslant \sigma_s; \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where σ_s denotes the threshold of dissimilarity (the niche radius) and α is a constant parameter which regulates the shape of the sharing function. Although α does not have too much effect on the performance of the sharing function method, the parameter σ_s must be chosen correctly in order to define the niche size of the optimum. In most applications an $\alpha = 1$ or 2 is used.[32]

Replacement strategy. In the fitness sharing technique, particles in the crowded region reduce the fitness values of one another and thus the shared fitness value is reduced significantly, depending on the value of niche radius, σ_s.[31] As a result, particles with lower shared fitness values indicate that those particles belong to crowded regions in the solution space and larger shared fitness values indicate that the particles remain in less crowded regions. A certain percentage of the total population, say 25%, in the most crowded region, are identified based on their lower shared fitness values. Then these particles are removed and the same number of new particles is introduced into the swarm. At the same time, velocities of the newly introduced particles are initialised and the corresponding pbest values are reinitialised. This process is repeated after every predefined number of generations. It is important to note that, the global best solution, gbest, is always preserved and passed to the next generation for further computations involving that term. As a result, the algorithm converges to the best value, found in the whole optimisation process. This resembles the elitist strategy of an evolutionary optimisation process.

Key features of the proposed PSO algorithm.

1) The global version of PSO is employed.
2) The inertia coefficient ω is varied from a higher value to a lower value (0.9 to 0.4) as the algorithm progresses.
3) The acceleration coefficients, c_1 and c_2, are assumed to be constant.
4) After every predefined number of generations, say 10 or 20, calculate the shared fitness of each particle.
5) Save the gbest solution found so far in order to follow an elitist strategy.
6) Sort the particles based on shared fitness values in ascending order.
7) Identify a certain number (say 25%) of particles with lower shared fitness values.
8) Replace these particles with new particles generated randomly within the specified range.
9) Initialise the velocities of the corresponding particles with randomly generated values within the specified range.
10) Initialise the pbests of the corresponding particles with very big/small numbers for minimisation/maximisation problem.

4. THE PROPOSED COMMAND SHAPING TECHNIQUE

Realisation of conventional command shaping involves convolving a desired command with a sequence of impulses,[3,33] and the amplitudes and time locations of the impulses are calculated based on the natural frequencies and damping ratios of the system. A new command shaping method is proposed, as shown in Fig. 3, using the gain and delay elements to shape the reference input. The unshaped reference signal is passed through multiple delay units, Δ_i, and then multiplied with gain factors, K_i. The shaped command is formed by summing up the delayed and weighted components. The effectiveness of the proposed method depends on a suitable selection of the number of delay and gain elements and their corresponding values. For reasons of simplicity, the number of delay units and gain elements are kept the same. Assume that the number of gain elements and delay units are each represented by n. In order to achieve the same level in the system's response with the shaped command as with the unshaped reference, the gain values are selected in such a way that their sum is equal to unity,[3,33] i.e.

$$\sum_{i=1}^{n} K_i = 1\,. \qquad (9)$$

In order to minimise the delay in system's response, the first delay unit is set to zero, i.e., $\Delta_1 = 0$. The values of the remaining delay units, $\Delta_2, \Delta_3, \ldots \Delta_n$, and all gain values, $K_1, K_2, \ldots, K_n$, may be derived analytically as in a conventional command shaper.[3,33] Assuming that, no prior information is available about the natural frequencies and associated damping ratios of the system, the PSO is used to optimise the values of the gain and delay units so as to reduce the vibration of the system. A schematic diagram of the proposed method is shown in Fig. 3.

4.1. Finding Values of Gain and Delay Units

In the proposed command shaping scheme, as shown in Fig. 3, a PSO algorithm is employed to find the values of gain and delay units. The procedure can be described as follows:

Step 1. Set a value for numbers of gain and delay units n.
Step 2. Set the number of particles as $(2n-1)$ (gain elements $= n$; delay units $= n-1$; assuming $\Delta_1 = 0$).
Step 3. Create an initial population of dimension $(2n-1)N$, where N is the number of individuals. The initial population is created randomly within a range of $[0, 1]$.
Step 4. The first $n-1$ real numbers are converted into integer values by a conversion factor and rounding to nearest value and then assigned to $\Delta_2, \Delta_3, \ldots \Delta_n$, respectively.
Step 5. The remaining numbers are normalised and assigned to $K_1, K_2, \ldots, K_n$. Here normalisation is done in order to maintain: $\sum_{i=1}^{n} K_i = 1$.
Step 6. Once all gain and delay values are calculated, they are passed to the model, shaped command is formed and applied to the system (for, open loop control, see Fig. 4). The error signal is calculated as: $e(t) = d(t) - y(t)$, where $d(t)$ is the desired response and $y(t)$ is the system response.
Step 7. Using this error signal, an objective function can be formed as: $f(x) = f(e(t))$.
Step 8. The PSO process is formulated so as to minimise the objective function $f(x)$.

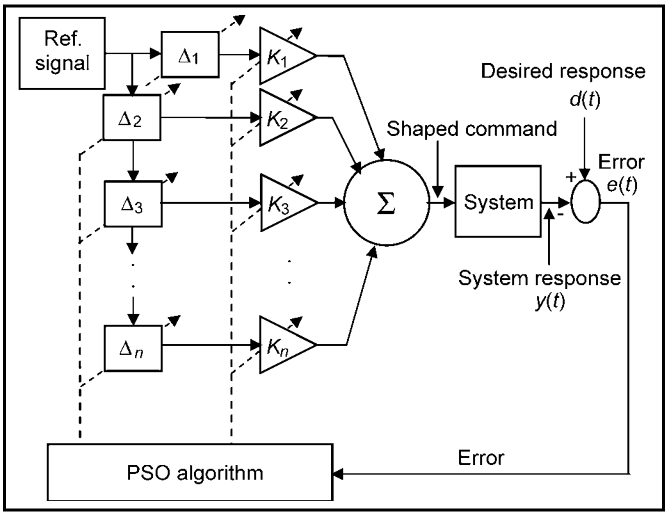

Figure 3. PSO-based command shaping scheme using gain and delay elements.

5. IMPLEMENTATION

The control strategy was implemented in a Simulink[34] environment as shown in Fig. 4. The PSO process was encoded in Matlab .m files.[35] An interface was created so that the gain and delay values were calculated with the PSO and passed to the Simulink environment, and after completion of the simulation, the system response was recorded and again passed to the PSO for further computation, and the process was repeated based on the initial population and total generation of the optimisation process. For $n = 3$, the number of the gain elements is 3 and number of the delay units is effectively 2, since the first delay unit is set to zero. The three gain units are termed as Gain 1, Gain 2 and Gain 3, and the corresponding values are indicated as K_1, K_2 and K_3. The two delay units, associated with Gain 2 and Gain 3, are indicated as Delay 1 and Delay 2, respectively. The value of n was selected intuitively to keep a resemblance with the number of impulses in a zero vibration derivative (ZVD)[33] type command shaper. A fourth order continuous transfer function, $H(s)$, characterising the vertical movement of the TRMS was extracted offline[22] and utilised in this work (see Fig. 4). This is given as:

$$H(s) = \frac{y(s)}{u(s)} = \frac{-0.08927s^3 + 2.249s^2 - 45.57s + 595.1}{s^4 + 3.469s^3 + 519.6s^2 + 35.95s + 2189}. \quad (10)$$

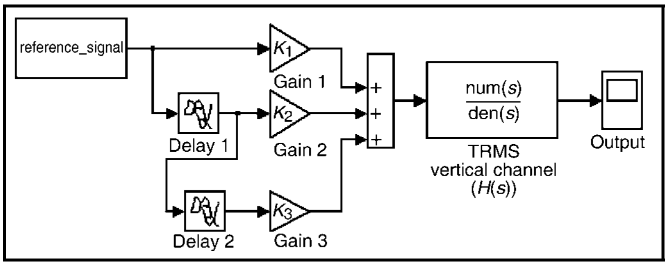

Figure 4. Simulink model of command shaping using gain and delay units.

5.1. Parameter Encoding

The PSO algorithm begins with a population of real numbers called a swarm. Each row represents a solution set called particle. A swarm of ten particles with five elements each, i.e., 10×5 is created randomly within the range of $[0, 1]$. The first three elements of each individual are normalised and assigned to K_1, K_2 and K_3. In Matlab/Simulink,[34,35] the delay units are usually represented in terms of the number of samples, which is an integer value. Thus, the remaining two elements of each individual are converted into integer numbers with a conversion factor of 0.01 followed by a rounding operation and then assigned to Delay 1 and Delay 2 as indicated in the above discussion. The process was run for a maximum number of 200 generations.

5.2. Selection of Objective Function and Niche Radius

For an evolutionary based design procedure, the search capability of the algorithm is directly affected by the nature of the objective function. Moreover, the design objective can only be achieved upon suitable selection of the objective function for the optimisation process. Commonly used objective functions are the sum of the absolute error (SAE), the sum of the squared error (SSE), the mean squared error (MSE), the root mean squared error (RMSE) and the time weighted sum of absolute error (TSAE). The main aim of the optimisation process is to reduce the vibration of the TRMS under the operation of the vertical channel. In such a case, the desired response of the system, $d(t)$, is set to zero in order to achieve zero vibration while the system is in operation. Thus, the system response, $y(t)$, is considered as the error signal, $e(t)$, which in turn is used to formulate the objective function for the PSO algorithm. In the proposed PSO algorithm, 25% of the total number of particles is replaced with randomly generated new particles after every 10 generations, based on the shared fitness values of the particles. In a fitness sharing technique, the value of the niche radius, σ_s is crucial along with the objective function in selecting which particles are to be replaced. Moreover, the diversity in the population as well as the convergence to a global solution depends on a suitable selection of σ_s. The algorithm was run on this problem with different values of σ_s and with different objective functions, such as, SAE, MSE, RMSE and the weighted sum of multiple objectives (weighted sum of the normalised rise-time and settling-time). The design process was run in a Matlab/Simulink simulation environment, and time-domain performance measures of the system's (model of vertical channel of the TRMS as stated in the Eq. (10)) response with shaped commands derived with different niche radii and objective functions were recorded. These are presented in Table 1.

Table 1. Performance measures of output response due to shaped commands designed with different objective functions and niche radii.

Objective function	Niche radius (σ_s)	Over-shoot (%)	Rise time (s)	Settling time (s)	Steady-state error
Sum of absolute error (SAE)	0.1	0.0007	0.6	0.9	0
	0.5	0.0006	1.2	1.5	0
	0.9	0.0203	1.5	1.9	0
Mean squared error (MSE)	0.1	6.0779	1.6	29.1	0
	0.5	3.45	1.5	12.9	0
	0.9	12.303	2.5	42.3	0
Root of mean squared error (RMSE)	0.1	3.476	1.5	13.6	0
	0.5	2.808	1.7	11.6	0
	0.9	3.654	1.6	16.1	0
Weighted sum	0.1	1.524	0.5	0.8	0
	0.5	1.5608	0.4	0.7	0
	0.9	1.5604	0.5	0.8	0

The output responses of the vertical channel due to shaped commands obtained with different objective functions but with a fixed $\sigma_s = 0.1$ are shown in Fig. 5. For clarity, only the leading edge of the system response is shown in order to highlight the differences among the various responses in terms of system overshoot, rise time, and settling time. It is evident that using an objective function with different values of σ_s results in different outputs. Similarly, using a fixed σ_s with different objective functions results in different outputs.

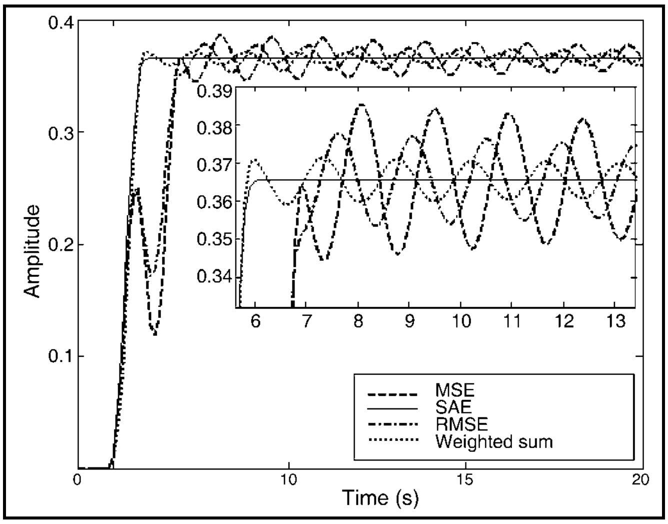

Figure 5. Response of vertical channel (leading edge only) with shaped commands for different objective functions (niche radius = 0.1).

The system responses due to the shaped commands obtained with the weighted sum as the objective function produced better results compared to the other objective functions in terms of rise-time and settling-time, although there was little overshoot associated with the response. For all the objective functions, except RMSE, a better system response was recorded at $\sigma_S = 0.1$ in terms of the various performance measures, as shown in Table 1. Among these, the system response due to the shaped command obtained with SAE as the objective function was better in terms of overall system performance; almost zero overshoot with satisfactory rise-time, settling-time and steady-state error.

5.3. Solution and Shaped Command

The algorithm was run for 200 generations in a Matlab/ Simulink[34,35] simulation environment. The resulting gain values were: $K_1 = 0.4015, K_2 = 0.2785, K_3 = 0.32$ and delay units were: Delay $1 = 50$ and Delay $2 = 90$. Here the delay units are represented in terms of the number of samples. These values were obtained using SAE as the objective function with $\sigma_S = 0.1$. A bang-bang signal was used as the reference signal. Both the bang-bang input and its corresponding shaped signal are shown in Fig. 6. For clarity, the leading edge is enlarged in Fig. 6. The time-domain responses of the vertical channel due to bang-bang and shaped inputs are shown in Fig. 7.

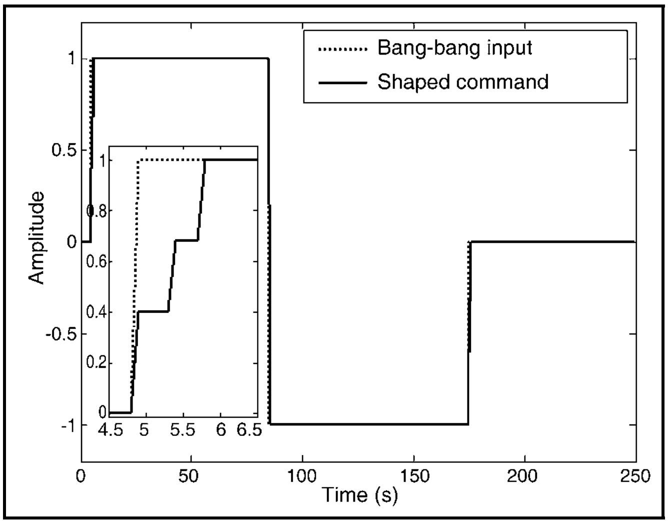

Figure 6. Bang-bang input and shaped command (time-domain).

A relatively large amount of oscillation is observed in the vertical channel in the response to the bang-bang input signal and it takes the system a relatively long time to settle to a steady position. It is clearly evident that the system response oscillation completely diminishes with the shaped command and the system settles quickly to a steady state. This shows that the shaped command can improve several time domain performance measures, such as overshoot and settling time. Although the reduction of oscillation in system's response is directly related to the reduction of the system vibration, a frequency domain representation of the response is so as to highlight the dominant modes of the system and the corresponding vibration reduction at those modes. The frequency-domain representations of the bang-bang input and the shaped command are shown in Fig. 8 and the corresponding system's responses are shown in Fig. 9.

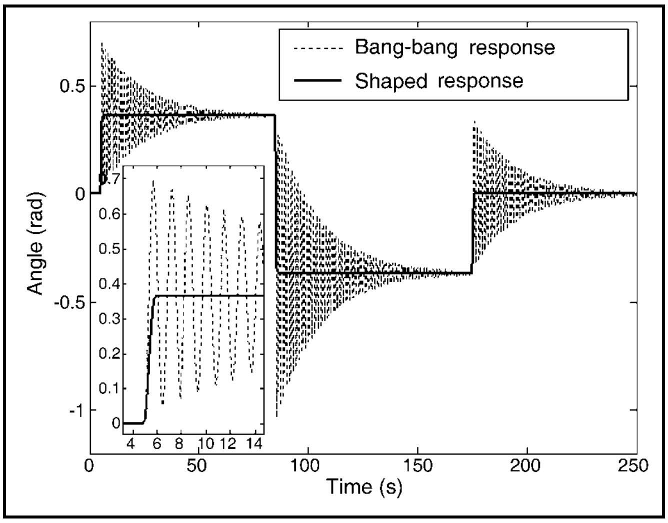

Figure 7. Response of vertical channel due to bang-bang signal and shaped command (time-domain).

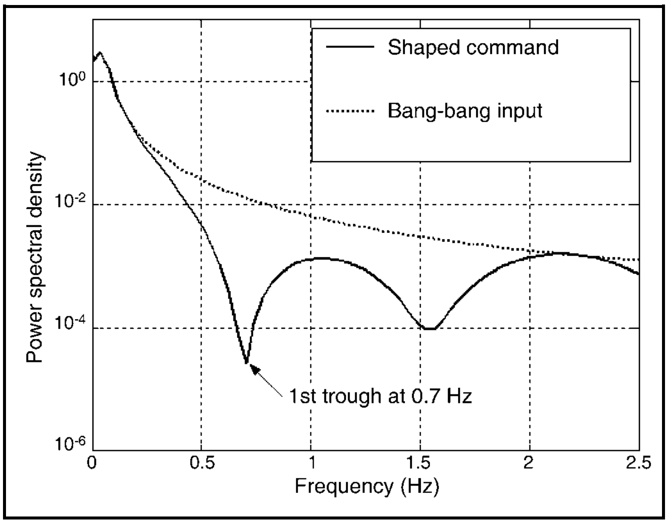

Figure 8. Bang-bang input and shaped command (frequency-domain).

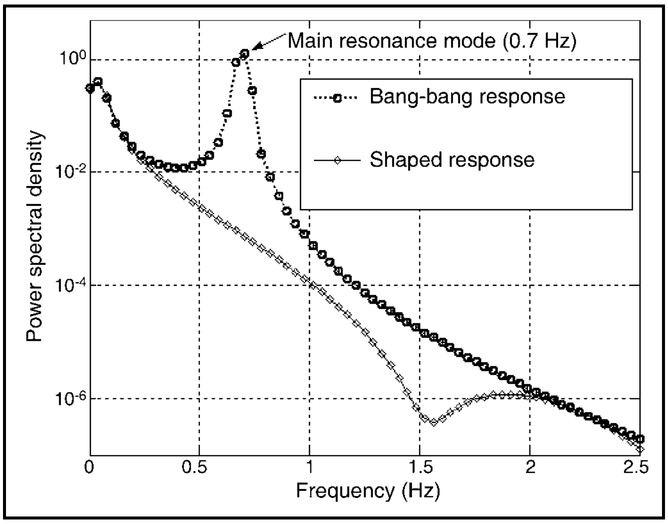

Figure 9. Response of vertical channel due to bang-bang signal and shaped command (frequency-domain).

It is noted in the system's response due to the bang-bang signal that the system has only one dominant mode (the peak in the frequency-domain representation) which appears at 0.7 Hz.

For the bang-bang input, the total energy seems to be evenly distributed along the frequency scale, although it is higher near the dc level. On the contrary, several troughs occur in the frequency-domain representation of the shaped command indicating a decrease in energy level at those frequencies. Most importantly, the first trough occurs exactly at 0.7 Hz where the main resonance mode of the system lies. As a result, the shaped command, when applied to the system, does not excite the system at its resonance mode to the extent that the bang-bang input does. A spectral attenuation of 31.3 dB was recorded at the dominant mode (0.7 Hz) of the system with the shaped command as the input, compared to that with the bang-bang input. This large amount of attenuation in vibration clearly demonstrates the effectiveness of the algorithm and the proposed control strategy in the control of vibration of flexible-structure systems.

6. EFFECT OF NUMBER OF GAIN AND DELAY UNITS

In order to investigate the effect of the number of gain and delay units in the command shaper on vibration reduction of a flexible system, the design procedure was repeated with different numbers of gain and delay units and the system performance was assessed in the objective domain. The number of impulses in a theoretical zero vibration (ZV) based command shaper is two.[3] For an extra insensitive (EI) based command shaper, the number of impulses is four.[33] In order to emulate conventional ZV-based and EI-based command shapers, two command shapers were designed where the number of gain elements is chosen as two and four, respectively. Moreover, a further command shaper was designed where the number of gain units was arbitrarily chosen as 10, which is quite high compared to those of the others. The PSO algorithm was utilised with the same parameters and same objective function for obtaining the optimal solutions for these three command shapers. The responses of the vertical channel due to the shaped signals obtained with the command shapers are shown in Fig. 10 and the corresponding time-domain performance measures are recorded.

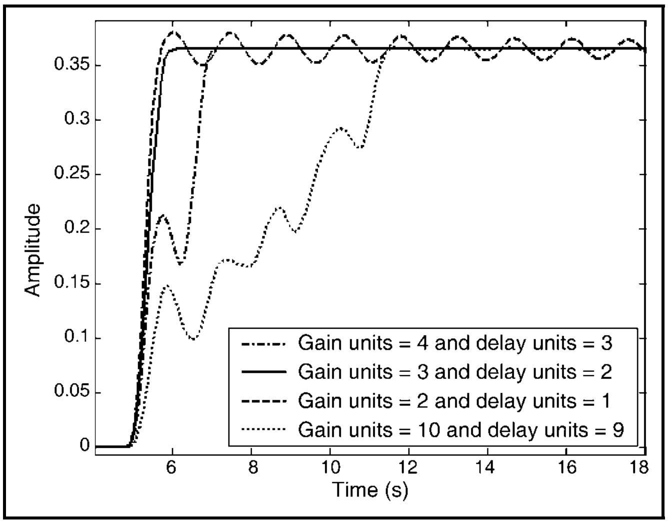

Figure 10. Leading edges of output response with command shapers with different numbers of gain and delay units.

For the command shaper with two gain units and one delay unit, the overshoot, rise-time and settling-time were recorded as 0.5%, 0.49 s and 0.7 s, respectively. For a large number of gain and delay units (gain units $= 10$ and delay units $= 9$), the overshoot, rise-time and setting-time of the system's response were recorded as 0.25%, 5.9 s and 6.4 s, respectively. It is noted in Fig. 10 that the system's response due to shaped signal obtained with command shaper having minimum number of gain and delay units (gain unit $= 2$ and delay unit $= 1$) is the fastest but with the cost of overshoot, whereas for a higher number of gain and delay units the system's response suffers from relatively longer delay with initial oscillations. It is clearly evident that the system response due to shaped signal obtained with command shaper having moderate number of gain and delay units (gain units $= 3$ and delay units $= 2$), is satisfactory in terms of the various performance measures.

7. ONLINE ADAPTIVE REALISATION OF THE PROPOSED COMMAND SHAPER

Due to the convolution process, a conventional command shaping technique may prove to be impractical in online applications. In order to demonstrate the applicability of the proposed control strategy in an adaptive context, the controller was designed in such a way that the values of gain and delay units are adjusted online based on the error between the desired and actual response of the system. A proposed PSO with a small swarm size was employed to adjust the values of gain and delay units so as to minimise the error function. The whole control strategy was realised in Simulink while the PSO algorithm was incorporated as a Matlab function, CS_PSO.m as shown in Fig. 11.

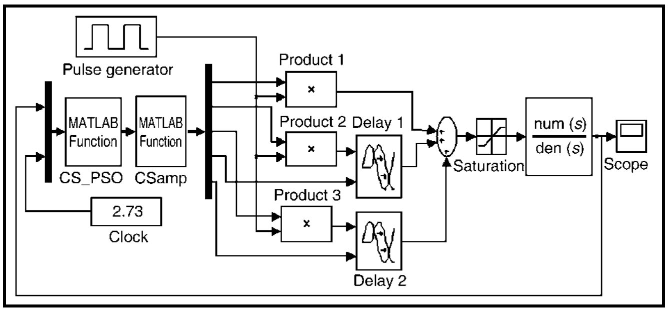

Figure 11. Simulink model of adaptive command shaper using PSO.

Three gain units were implemented, using Simulink product units, namely, Product 1, Product 2 and Product 3, while two delay units were implemented with two-input Simulink block transport delay unit, namely, Delay 1 and Delay 2. In order to reduce the delay in system's response, the first delay unit (following Product 1) was assumed to be zero. A pulse generator was used to generate the unshaped reference signal. This was applied simultaneously to all the gain and delay units. The gain and delay units were designed as variable units with two inputs; one is the signal and the other is the gain/delay value that is loaded from the PSO algorithm. The variable gain values were assigned as K_1, K_2 and K_3 whereas the variable delay units are assigned as Delay 1 and Delay 2. The PSO algorithm begins with a population of five particles with five elements each, i.e., 5×5, created randomly within the range of $[0, 1]$. The first three elements of each particle are normalised and assigned to gain elements, K_1, K_2 and K_3. In Simulink, the delay units are represented in terms of number of samples. Thus, the remaining two elements of each in-

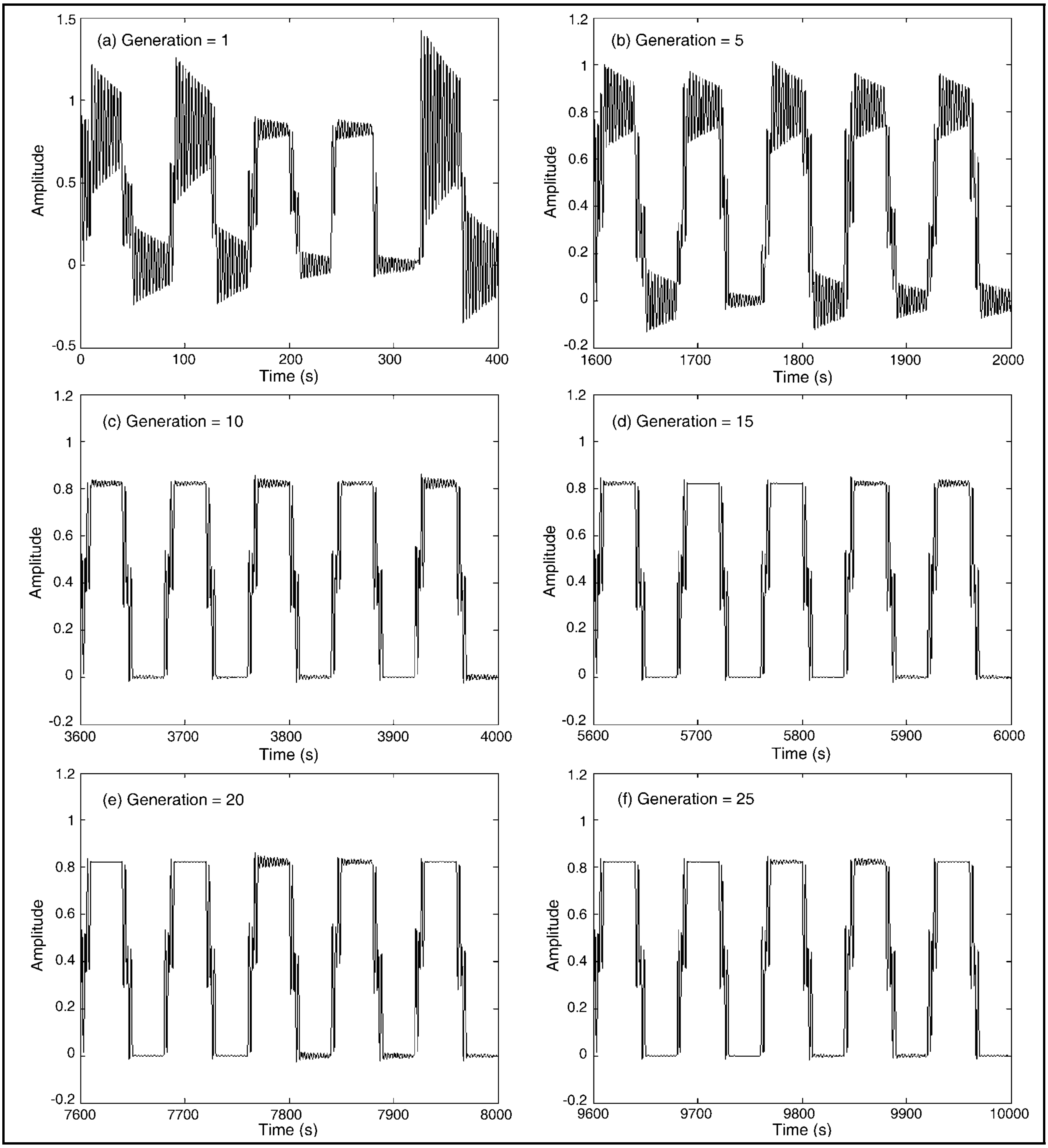

Figure 12. System response with the adaptive command shaper at different generations.

dividual are converted into integer numbers with a conversion factor of 0.01 following a "round" operation and then assigned to Delay 1 and Delay 2 and the model is simulated for a pulse width of 80 s with a duty cycle of 50%. The Matlab function CS_PSO.m has two inputs: clock and error signals. In this example the sum of absolute error (SAE) was used as the objective function of the PSO optimisation process. The five output signals of CS_PSO function; values of 3 gain and 2 delay units were passed to another Matlab function, CSamp.m which acted as a buffer in order to adjust the timing sequence of the Simulink solver.

For each particle/individual, the model was simulated for a full pulse duration of 80 s. Since the swarm constitutes five particles, it takes 400 s (40×5) to complete one generation. Figure 12 shows the adaptation process of the PSO based on the adaptive command shaper at different generations. Since PSO is a population-based global search technique and the initial population is created randomly, a relatively large level of oscillation is observed at the first few generations (see Figs. 12(a) and (b)). A saturator unit was used at the input of the system in order to protect the system from unusually large inputs that may cause damage or drive the system into its nonlinear dynamic region. It is observed that the system response gradually improves as the algorithm proceeds. It is interesting to note that, at generation 15, the oscillation is reduced significantly and in subsequent generations; the system appears to perform better. This clearly demonstrates the effectiveness of the PSO algorithm and the adaptive command shaping strategy in vibration control applications. To highlight the vibration reduction at the dominant mode, the system response is presented in the frequency-domain in Fig. 13. It is observed that the dominant mode appears at 0.66 Hz and reduction is quite significant at this point. It is noted that, from generation 1 to generation 5, the reduction is comparatively low and for subsequent generations, the reduction is considerably higher. The frequency domain responses at generations 10, 15, 20 and 25 appear to overlap each other which reveal insignificant performance improvement beyond generation 10 in this case.

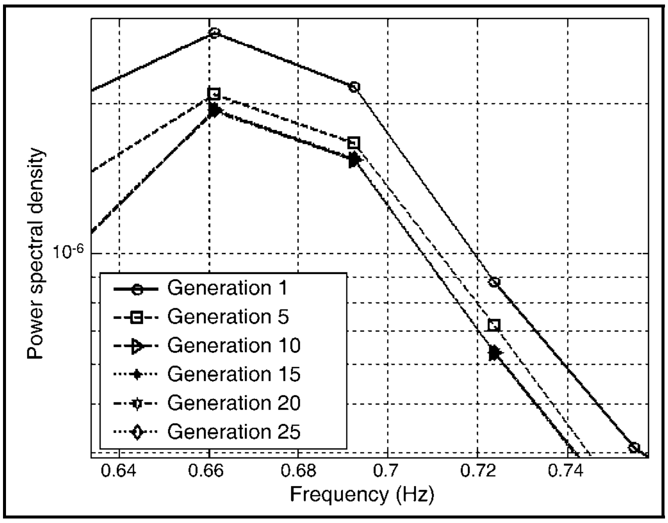

Figure 13. Frequency-domain system response with the adaptive command shaper at different generations.

8. CONCLUSION

A new command shaping technique based on gain and delay units has been presented for vibration reduction in flexible-structure systems. Assuming that no prior information is available about the system, a new variant of the PSO algorithm has been developed and used to optimise the values of gain and delay units of the command shaper.

A large amount of vibration reduction has been achieved with satisfactory performance measured in terms of time-domain response parameters such as overshoot, rise-time, settling-time and steady-state error. Moreover, an adaptive control strategy has been developed and presented based on the proposed command shaping technique where the PSO is used to adjust the controller parameters online. The performance of the proposed algorithm, both online and offline, demonstrates the potential of PSO in controller design. The results have clearly demonstrated the effectiveness of the proposed control strategy in the vibration reduction of flexible-structure systems both in offline and online adaptive environments. Moreover, the control strategy may be extended to complex multi-input, multi-output systems due to its simplicity in the design formulation, ease of implementation and overall performance.

REFERENCES

1. Benosman, M. and Vey, L. Control of flexible manipulators: A survey, *Robotica*, **22**, 535-545, (2004).
2. Shaheed, M. H., Tokhi, M. O., Chipperfield, A. J., and Azad, A. K. M. Modelling and open-loop control of a single-link flexible manipulator with genetic algorithms, *Journal of Low Frequency Noise, Vibration and Active Control*, **20** (1), 39-55, (2001).
3. Singer, N. C. and Seering, W. P. Preshaping command inputs to reduce system vibration, *Trans. ASME, J. Dynamic Systems, Measurement and Control*, **112** (1), 76-82, (1990).
4. Mohamed, Z. and Tokhi, M. O. Vibration control of a single-link flexible manipulator using command shaping techniques, *Proceeding of Inst. Mech. Eng., Part I, J Systems and Control Engineering*, **216**, 191-210, (2002).
5. Singhose, W., Singer, N., and Seering, W. Improving repeatability of coordinate measuring machines with shaped command signals, *Precision Engineering*, **18**, 138-146, (1996).
6. Singh, T. and Vadali, S. R. Robust time-delay control, *ASME Journal of Dynamic Systems, Measurement, and Control*, **115**, 303-306, (1993).
7. Singh, T. and Vadali, S. R. Robust time-delay control of multimode systems, *Proceedings of 13th American Control Conference*, Baltimore, MD, USA, 29 June-1 July, 681-685, (1994).
8. Singh, T. and Vadali, S. R. Robust time-delay control of multimode systems, *Int. Journal of Control*, **62**, 1319-1339, (1995).
9. Bodson, M. An adaptive algorithm for the tuning of two input shaping methods, *Automatica*, **34** (6), 771-776, (1998).
10. Sungsoo, R. Adaptive command shaping using adaptive filter approach in time domain, *Proceeding of American Control Conference*, California, USA, 81-85, (1999).
11. Alam M. S., Aldebrez, F. M., and Tokhi, M. O. Adaptive command shaping using genetic algorithms for vibration control, *Proceedings of IEEE SMC UK-RI 3rd Workshop on Intelligent Cybernetic Systems*, Ulster, UK, 7-8 Sept., (2004).
12. Holland, J. H. *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor MI, (1975).
13. Tokhi, M. O., Alam, M. S., Zain, M. Z. M. and Aldebrez, F. M. Adaptive command shaping using genetic algorithms for vibration control of a single link flexible manipulator, *Proceedings of 12th International Congress on Sound and Vibration*, Lisbon, Portugal, 11-14 July, (2005).
14. Alam, M. S. and Tokhi, M. O. Designing feedforward command shapers with multi-objective genetic optimisation for vibration control of a single-link flexible manipulator, *Engineering Applications of Artificial Intelligence*, (2007) (in press).
15. Kennedy, J. and Eberhart, R. Particle swarm optimization, *Proceeding of IEEE International Conference on Neural Networks (ICNN), IV*, Perth, Australia, 1942-1948, (1995).
16. Kennedy, J. and Eberhart, R. *Swarm intelligence*, Morgan Kaufmann Publishers, (2001).
17. Shi, Y. and Eberhart, R. Comparison between genetic algorithms and particle swarm optimization, *Evolutionary programming vii: proc. 7th ann. conf. on evolutionary conf.*, Springer-Verlag, Berlin, San Diego, CA, (1998).
18. Eberhart, R. and Shi, Y. Evolving artificial neural networks, *Proceeding of International Conference on neural networks and brain*, Beijing, P. R. China, PL5-PL13, (1998).
19. Alam, M. S. and Tokhi, M. O. Modelling of a twin rotor system: a particle swarm optimisation approach, *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, **221**, 353-375, (2007).

[20] Zwe-Lee G. A particle swarm optimization approach for optimum design of PID controller in AVR system, *IEEE Transactions on Energy Conversion*, **19** (2), 384-391, (2004).

[21] Huq, M. S., Alam, M. S., and Tokhi, M. O. Particle swarm optimization of spring brake orthosis, *Proceedings of IEEE SMC UK-RI 5th Chapter Conference on Advances in Cybernetic Systems*, Sheffield Hallam University, Sheffield, UK, 7-8 Sept., (2006).

[22] Alam, M. S., Shaheed, M. H., and Tokhi, M. O. Modelling and vibration control of a twin rotor system: a particle swarm optimisation approach, *Proceedings of 13th International Congress on Sound and Vibration*, Vienna, Austria, 2-6 July, (2006).

[23] Feedback Instruments Ltd. Twin Rotor MIMO System Manual 33-007-0, Sussex, UK, (1996).

[24] Ahmad, S. M., Chipperfield, A. J., and Tokhi, M. O. Dynamic modelling and linear quadratic Gaussian control of a twin rotor MIMO system, *Proceedings of IMechE-I: Journal of Systems and Control Engineering*, **217** (I3), 203-227, (2003).

[25] Livet, T., Fath, D., and Kubica, F. Robust autopilot design for a highly flexible aircraft, *Proceedings of IFAC World Congress*, San Francisco, California, July, 279-284, (1996).

[26] Landis, H. K., Davis, J. M., Dabundo, C., and Keller, J. F. Advanced flight control research and development at Boeing Helicopter, *Advances in Aircraft Flight Control* (Ed. M. B. Tischler), Taylor and Francis, London, 103-141, (1996).

[27] Shi, Y. and Eberhart, R. A modified particle swarm optimizer, *Proceeding of IEEE Int. Conf. Evolutionary Computation*, 69-73, (1998).

[28] Shi, Y. and Eberhart, R. Empirical study of particle swarm optimization, *Proceeding of IEEE Int. Congress Evolutionary Computation*, **3**, 101-106, (1999).

[29] Kennedy, J. and Mendes, R. Neighborhood topologies in fully informed and best-of-neighborhood particle swarms, *IEEE Transactions on Systems, Man and Cybernetics, Part C*, **36** (4), 515-519, (2006).

[30] Clerc, M. and Kennedy, J. The particle swarm: explosion, stability, and convergence in a multi-dimensional complex space, *IEEE Transactions on Evolutionary Computation*, **6**, 58-73, (2002).

[31] Goldberg, E. and Richardson, J. Genetic algorithms with sharing for multimodal function optimization, *J. Grefenstette, (Ed.), Proceedings of the Second International Conference on Genetic Algorithms*, Hillsdale, NJ: Lawrence Erlbaum Associates, 41-49, (1987).

[32] Deb, K. *Multi-objective optimization using evolutionary algorithms*, Wiley, New York, (2001).

[33] Singhose, W. E. Command generation for flexible systems, PhD thesis, S.B.M.E, Massachusetts Institute of Technology, USA, (1992).

[34] *SIMULINK Reference Guide*, The Math Works, Inc., (2006).

[35] *MATLAB Reference Guide*, The Math Works, Inc., (2006).