

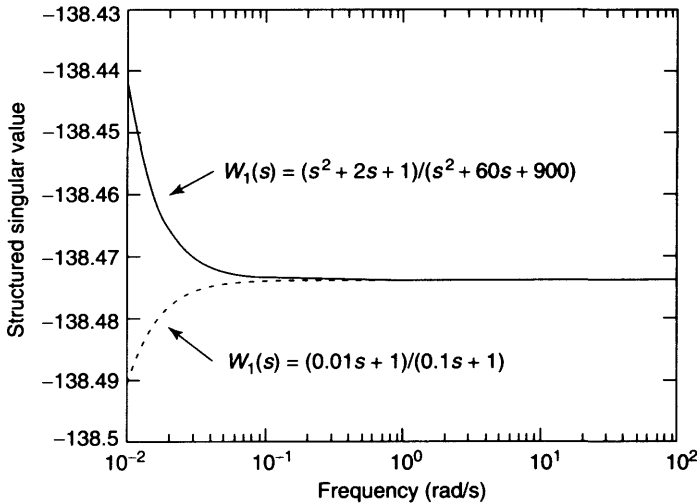


Figure 9.3 Structured singular value (dB) plots of the two optimal μ -synthesis controllers for active flutter-suppression of an aircraft wing

Many illustrative examples of how various control design techniques handle the problem of structured singular value synthesis, especially for robust flight control, can be found in Magni, et al. [6].

9.4 Time-Optimal Control with Pre-shaped Inputs

A topic of current interest is the development of *pre-shaped* inputs for controlling well modeled plants. The term *pre-shaped inputs* (or *input shaping*) refers to an open-loop control system, where the input vector, $\mathbf{u}(t)$, is pre-determined from certain strategy other than feedback. As an alternative to feedback control, input shaping offers a simpler control mechanism for implementation. However, being an open-loop design, input shaping is more sensitive to errors in modeling the plant dynamics when compared to a feedback design. Hence, input shaping requires a better modeling of a physical plant compared to the closed-loop control systems.

A common strategy for generating pre-shaped inputs is the *time-optimal control*. Consider a linear, time-invariant plant described by the following state-equation:

$$\dot{\mathbf{x}}^{(1)}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (9.31)$$

The time-optimal control refers to moving the system from an *initial state*, $\mathbf{x}(0)$, to a *final state*, $\mathbf{x}(t_f)$, such that the time taken, t_f , is the minimum. In other words, the time-optimal control input vector, $\mathbf{u}(t)$, minimizes the following objective function:

$$J(0, t_f) = \mathbf{x}^T(t_f) \mathbf{V} \mathbf{x}(t_f) + \int_0^{t_f} d\tau \quad (9.32)$$

subject to the constraint that the state-vector, $\mathbf{x}(t)$, must obey Eq. (9.31). Since there is usually a physical limit on the magnitude of control inputs that can be generated by a given actuator, each element, $u_j(t)$, of the input vector, $\mathbf{u}(t)$, must obey an additional constraint given by

$$|u_j(t)| \leq m_j \quad (9.33)$$

where m_j denotes the upper bound on $u_j(t)$. A solution to the time-optimal control problem given by Eqs. (9.31)–(9.33) is obtained by applying the *minimum-principle* [7] (also see Section 8.11), which minimizes $J(0, t_f)$ only at a *finite number of discrete* time points [8] (rather than in continuous time). Hence, the solution to the time-optimal control problem posed above results in a *digital* input – even for an analog plant. If the final state is zero, i.e. $\mathbf{x}(t_f) = \mathbf{0}$, then each element of the time-optimal control input, $\mathbf{u}^*(t)$, can be written as follows [8]:

$$u_j^*(t) = -m_j \operatorname{sgn}[\mathbf{B}_j^T \mathbf{p}^*(t)] \quad (9.34)$$

where $\operatorname{sgn}[\cdot]$ denotes the *vector signum function* (defined as 1 when the corresponding element of the vector within square brackets is *positive*, and -1 when the corresponding element of the vector within square brackets is *negative*), $\mathbf{B}_j^T$ denotes the j th row of $\mathbf{B}^T$, and $\mathbf{p}^*(t)$ is the optimal *co-state vector*, which is defined as the solution of the following *co-state equation*:

$$\dot{\mathbf{p}}^{*(1)}(t) = -\mathbf{A}^T \mathbf{p}^*(t) \quad (9.35)$$

Equation (9.34) indicates that each input vector element, $u_j^*(t)$, assumes values of $\pm m_j$ depending upon the sign of $\mathbf{B}_j^T \mathbf{p}^*(t)$, which oscillates about 0. Such a time-optimal control thus exerts the *maximum possible* control input magnitude until $\mathbf{x}(t_f) = \mathbf{0}$ is reached, and is given the name *bang-bang input*. An alternative way of expressing the bang-bang input sequence given by Eq. (9.34) is by a *sampling switch* (see Section 8.1), which turns on and off at a fixed sampling interval, T . The bang-bang input sequence is thus regarded as an *ideal sampler* which produces a series of *pulses*, $u_j^*(t)$, weighted by the input value, $f_j(kT)$, according to Eq. (8.2), re-written as follows:

$$u_j^*(t) = \sum_{k=0}^{\infty} f_j(kT) \delta(t - kT) \quad (9.36)$$

where $f_j(kT) = -m_j \operatorname{sgn}[\mathbf{B}_j^T \mathbf{p}^*(kT)]$.

Considering the fact that the bang-bang input is constant within each sampling interval, Eq. (9.36) is said to *convolve* a series of impulses, $A(k)\delta(t - kT)$, with a step input, $m_j u_s(t)$, where $A(k)$ is the amplitude of the impulse applied at the k th sampling instant. Such a statement can be used with the MATLAB function *kron* which multiplies *two sequences*, in order to obtain the bang-bang input sequence as follows:

$$u_j^*(t) = m_j u_s(t) * A(k)\delta(t - kT) \quad (9.37)$$

where $*$ denotes the *convolution* of the step function, $m_j u_s(t)$, with the weighted impulse sequence, $A(k)\delta(t - kT)$, and $A(k)$ is given by

$$\begin{aligned} A(k) &= 1; & k &= 0, t_f/(kT) \\ A(k) &= 2(-1)^k; & k &= 1, \dots, t_f/(kT) - 1 \end{aligned} \quad (9.38)$$

It can be shown [8] that if the eigenvalues of $\mathbf{A}$ are real, then each component of the optimal control vector, $\mathbf{u}^*(t)$, can switch from m_j to $-m_j$, or from $-m_j$ to m_j , at most $(n - 1)$ times, where n is the order of the plant. For example, a second order plant requires an optimal bang-bang input sequence that switches only once between $t = 0$ to $t = t_f$. Since the sampling interval is a constant for the bang-bang input, it follows that for a second order plant, $T = t_f/2$. Hence, switching occurs at the middle of the control input sequence, and the impulse magnitudes at the various values of k are given as follows for a second order plant:

$$\begin{bmatrix} k \\ A(k) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 2 \\ 1 & -2 & 1 \end{bmatrix} \quad (9.39)$$

Note that the bang-bang input sequence only moves the plant to a final zero state. If the final state, $\mathbf{x}(t_f)$, is non-zero, the bang-bang input sequence must be modified.

A practical application of the time-optimal pre-shaped inputs is in the control of flexible structures, where it is required to make the vibration in the structure (i.e. velocity and acceleration of flexible modes), zero at the end of the input sequence. There are many possible ways to develop such vibration suppression pre-shaped inputs, which has been the subject of active research for many years. However, since input shaping is an open-loop control, only those input shapes can be successfully implemented that are robust with respect to variations in the plant's model. Singer and Seering [9] proposed the following robust, weighted impulse sequence, $A(k)\delta(t - kT)$, based on linear dynamics to reduce the residual vibration of a flexible structure with one rigid-body mode and N flexible modes, and of total mass, m :

$$u_j^*(t) = m_j u_s(t) * A(k_0)\delta(t - k_0 T_0) * A(k_1)\delta(t - k_1 T_1) * A(k_2)\delta(t - k_2 T_2) * \dots * A(k_N)\delta(t - k_N T_N) \quad (9.40)$$

where k_0 and $A(k_0)$ are the bang-bang impulse sequence given by Eq. (9.39), and k_i and $A(k_i)$ denote the sampling instants and impulse amplitudes, respectively, for the i th flexible mode of the structure given by

$$\begin{bmatrix} k_i \\ A(k_i) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 2 \\ 1/4 & 1/2 & 1/4 \end{bmatrix} \quad (9.41)$$

$T_0 = t_f/2$, and $T_i = 2\pi/\omega_i$, where ω_i is the natural frequency of the i th flexible mode. Clearly, t_f is a summation of all the sampling instants of the input sequence given by Eq. (9.40). The convolutions required in Eq. (9.40) can be carried out using MATLAB with the function *kron*. An M-file which generates the input sequence of Eq. (9.40) for a single-input plant, assuming a maximum input magnitude of unity, is *inshape.m*, listed in Table 9.1. The M-file *inshape.m* requires the natural frequencies of the structure as a column vector, $\mathbf{w}$, and the final time, t_f , and returns the input sequence as an amplitude vector, $\mathbf{u}$, and the discrete time vector, $\mathbf{t}$:

```
>>[u,t]=inshape(w,t_f) <enter>
```

The M-file *inshape.m* assumes a maximum input magnitude of unity. If the maximum input magnitude is not unity, you should multiply the input sequence, $\mathbf{u}$, by the maximum

Table 9.1 Listing of the M-file *inshape.m* for generating pre-shaped inputs for vibration suppression of a linear flexible structure with a rigid body displacement*inshape.m*

```

function [a,t]=inshape(w,tf)
%Input shaping for robust, time-optimal control of nth order system
% w= vector of natural frequencies of the structure (rad/s)
% (first element of w must be zero, denoting the rigid body mode)
% tf=final time
% a = returned vector of input magnitudes
% t = returned vector of discrete times for the input sequence
% Copyright (c) 2000 by Ashish Tewari
%size of w vector is n
n=size(w,1);
p=n-1; % p is the no. of flexible modes
% Calculate the time-periods of the flexible modes:-
for i=2:n;
T(i-1)=2*pi/w(i);
end
% Assign the flexible mode impulse sequences:-
for i=1:p;
F1(i,:)= [0 0.5*T(i) T(i)];
ef1(i,:)=exp(F1(i,:));
F2=[0.25 0.5 0.25];
end
% Convolve the impulse sequences for flexible modes:-
a1=zeros(1,3^p);
et=zeros(1,3^p);
size(a1)
if p>1
a1=[kron(F2,F2) zeros(1,3^p-9)];
et=[kron(ef1(1,:),ef1(2,:)) zeros(1,3^p-9)];
else
a1=F2;
et=ef1(1,:);
end
for i=2:p-1;
a1=[kron((nonzeros(a1))',F2) zeros(1,3^p-3^(i+1))];
et=[kron((nonzeros(et))',ef1(i+1,:)) zeros(1,3^p-3^(i+1))];
end
% Assign impulse sequence for the rigid-body mode:-
Ftf=[0 0.5*tf tf];
Fa=[1 -2 1];
% Convolve rigid-body impulse sequence with that of the flexible
% modes:- a0=kron(a1,Fa)';
eft=exp(Ftf);
et1=kron(et,eft);
% Generate input sequence:-
t0=log(et1)';
[t0,i]=sort(t0);

```

(continued overleaf)

Table 9.1 (continued)*inshape.m*

```

a0=a0(i);
m=size(t0,1);
a2(1)=0;
for j=2:m+1;
a2(j)=a2(j-1)+a0(j-1);
end
eps=1e-10;
t(1)=0;
t(2)=eps;
for j=2:m-1;
t(2*j-1)=t0(j)-eps;
t(2*j)=t0(j);
end
t(2*m-1)=t0(m);
t(2*m)=t0(m)+eps;
a(1)=a2(1);
a(2)=a2(2);
for j=2:m;
a(2*j-1)=a2(j);
a(2*j)=a2(j+1);
end

```

magnitude. It is not required that the frequency vector, $\mathbf{w}$, for *inshape.m* should contain *all* the natural frequencies of the plant. Instead, it is a common practice to select a few dominant modes (usually the ones with the *smallest* natural frequencies) for input shaping. However, the higher frequency modes will not be suppressed by an input shaper that is based only on the first few modes. Let us examine the multi-mode input shaping for a simple system with a smooth frequency spectrum – a *train* of n equal masses.

Example 9.3

Consider a simple system of a train of n identical masses, m , connected by identical springs of stiffness, k . A force, $f(t)$, applied on the *first* mass is the input to the system, while the output is the displacement of the center of mass, $x = 1/n \sum x_i$, where x_i is the displacement of the i th mass. A state-space representation of the plant can be obtained with the state-vector $\mathbf{x}(t) = [x_1(t); x_2(t); \dots; x_n(t); x_1^{(1)}(t); x_2^{(1)}(t); \dots; x_n^{(1)}(t)]^T$. An M-file called *nmass.m*, which generates the state-space representation for a train of n -masses, is given in Table 9.2. Let us select $n = 13$, $m = 1$ kg and $k = 10^5$ N/m, and generate the state-space model and the natural frequencies of the plant in increasing magnitudes as follows:

```

>>[A,B,C,D]=nmass(13,1e5); freq=sort(damp(A))' <enter>

freq =
Columns 1 through 7
0.0000 0.0000 76.2341 76.2341 151.3565 151.3565 224.2718

```

Table 9.2 Listing of the M-file *nmass.m* for the state-space model of a train of n identical masses, connected by $(n - 1)$ identical springs

```
nmass.m
```

```
function [a,b,c,d]=nmass(n,k)
% system of n unit masses connected with n-1 springs of
% stiffness k, with input force applied to the first mass
% and displacement of center of mass as the output
K=zeros(n);
K(1,:)=k*[1 -1 zeros(1,n-2)];
K(n,:)=k*[zeros(1,n-2) -1 1];
for i=2:n-1;
K(i,:)=k*[zeros(1,i-2) -1 2 -1 zeros(1,n-i-1)];
end
a=[zeros(n) eye(n);-K zeros(n)];
b=[zeros(1,n) 1 zeros(1,n-1)]';
c=[ones(1,n) zeros(1,n)]/n;
d=0;
```

Columns 8 through 14

224.2718 293.9167 293.9167 359.2757 359.2757 419.3956 419.3956

Columns 15 through 21

473.3998 473.3998 520.5007 520.5007 560.0116 560.0116 591.3562

Columns 22 through 26

591.3562 614.0775 614.0775 627.8442 627.8442

The 26th order plant has one rigid and 12 flexible second-order modes with frequencies 76.2, 151.4, 224.3, 293.9, 359.3, 419.4, 473.4, 520.5, 560.0, 591.4, 614.1, and 627.8 rad/s. Since there is a double eigenvalue at $s = 0$ (denoting the rigid-body mode), the plant is unstable. The damping ratios associated with the flexible modes are all zeros, indicating an undamped plant. Although the plant is undamped, a Bode gain plot of the system would be *smooth* with no resonant peaks. This is because the plant is *minimum phase*, and has all imaginary-axis poles cancelled by the transmission zeros. The control objective is to cancel the residual velocity and acceleration at the end of the input sequence, while keeping the motion of the center of mass, $y(t)$, negligible.

To see the vibration suppression achieved by an input shaper obtained by taking only the first few modes in Eq. (9.40) instead of all 12 flexible modes, let us calculate the input sequence with only the first two flexible modes for $t_f = 0.1$ s, assuming the maximum input magnitude of unity:

```
>>w=freq(1:2:6) '<enter>
```

```
w =
    0.0000
   76.2341
  151.3565
```

```
>>[u,t]=inshape(w, 0.1); % input profile with only first
    two flexible modes <enter>
```

The displacement of the center of mass to the pre-shaped input sequence is obtained using *lsim* as follows:

```
>>sys=ss(A,B,C,D); [y,t,X]=lsim(sys,u,t); <enter>
```

whereas the velocity of the first mass is calculated as follows:

```
>>sys1=ss(A,B,[zeros(1,13) 1 zeros(1,12)], 0);
    [x1,t,X]=lsim(sys1,u,t); <enter>
```

The pre-shaped input sequence, $u(t)$, the resulting center of mass displacement, $y(t)$, and the velocity of the first mass, $dx_1(t)/dt$, are plotted in Figure 9.4. Note that the vibration suppression is evident with zero velocity of the first mass at the end of the input sequence. In a practical implementation, input shaping can be combined with a *feedback* tracking system which moves the train by a given displacement, or achieves a given velocity of the center of mass, while suppressing the residual vibration using pre-shaped inputs as a *feedforward controller*.

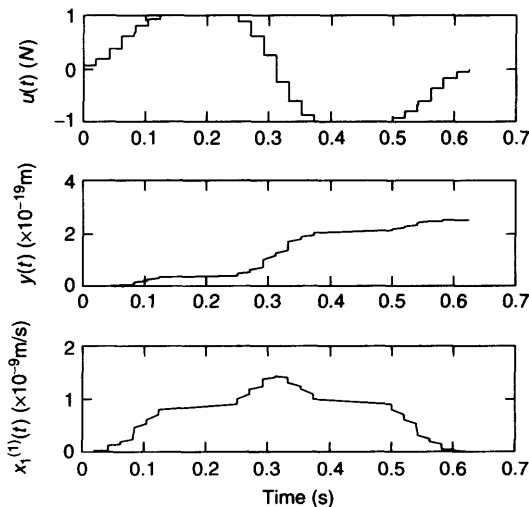


Figure 9.4 Pre-shaped input, $u(t)$, center of mass displacement, $y(t)$, and velocity of the first mass, $x_1^{(1)}(t)$, for a train of 13 identical masses connected by springs

Many investigators [10,11] have applied the input shaping approach to practical problems. Since robustness of input sequences with variations in the plant model is crucial, Singhose, Derezinski and Singer [12] derived an *extra-insensitive* input shaper using on-off reaction jets for time-optimal maneuvers of spacecraft. The robustness of the shaper was addressed [12] by minimizing the *percentage residual vibration* at the end of the input sequence with respect to the frequency of each flexible mode considered.