Available online at www.sciencedirect.com
ScienceDirect
Journal of Sound and Vibration 298 (2006) 1001–1018
JOURNAL OF SOUND AND VIBRATION
www.elsevier.com/locate/jsvi

Wave-echo control of lumped flexible systems

William J. O'Connor*

Mechanical Engineering, University College Dublin, National University of Ireland, Dublin

Received 27 May 2005; received in revised form 30 May 2006; accepted 1 June 2006
Available online 10 August 2006

Abstract

An elegant, generic solution is presented to the problem of point-to-point control by a single actuator of a remote load through an intermediate flexible system, modelled by a system of lumped masses and springs. It is based on new ways of looking at the problem that respect and exploit the underlying dynamics. Under wide-ranging conditions the strategy allows rapid, almost-vibrationless, precise position control of the load, independently of the order of the system, without the need for a detailed system model or ideal actuator. During the start-up, the system itself reveals to the controller how to terminate the motion, so that the real system also acts as the model for the controller. The scheme is very robust to modelling, actuator and sensor errors. The strategy is presented, with some of the motivating ideas reviewed.

1. Introduction

From space structures to disk drive heads, from medical mechanisms to long-arm manipulators, from cranes to robots, there are many contexts in which it is desired to achieve rapid and accurate position control of a load (or system end-point) by an actuator that is separated from the load by an intermediate flexible system. While all systems are to some extent flexible, issues related to flexibility become decisive when designing lighter mechanisms, or more dynamically responsive systems, or deliberately softer devices, or more energy-economical systems, or devices which are long in one direction relative to the other dimensions.

The system's actuator must then attempt to reconcile two demands, namely, position control and active vibration damping. Somehow each must be achieved while respecting the other's requirements. Previous approaches to controlling such flexible systems have included various classical and state feedback control techniques (often using simplified dynamic models); modal control (often considering a rigid-body or zero frequency mode separately from vibration modes); linear quadratic optimal control; sliding mode control; input command shaping; bang–bang control; wave-based control; and control based on real-virtual system models. Each method has special characteristics and drawbacks, discussed in Ref. [1–19]. None is completely satisfactory under all headings: some techniques control (or inhibit) only a few modes, or just one; some require a very good system model or are otherwise highly system dependent; some are not robust to timing or modelling errors or to actuator limitations. Much effort has been expended trying to refine or improve each

*Tel: +353 1 7161887; fax: +353 1 2830534.
E-mail address: william.oconnor@ucd.ie.

0022-460X/$ - see front matter © 2006 Published by Elsevier Ltd.
doi:10.1016/j.jsv.2006.06.010

method, or to mitigate some significant disadvantages, with differing degrees of success. Ref. [19] observes that "to date a general solution to the control problem [of flexible structures] has yet to be found. One important reason is that computationally efficient (real-time) mathematical methods do not exist for solving the extremely complex sets of partial differential equations and incorporating the associated boundary conditions that most accurately model flexible structures."

This paper presents a new approach that avoids most of the shortcomings in previous approaches and offers precisely a "general solution" that has long been sought, one that is applicable to a large class of problems.

2. The problem

The main focus of this paper is the arrangement shown in Fig. 1. A rectilinear mechanical system of lumped masses and springs is controlled by a single actuator at one end, with the "load" (or simply an end-mass) at the far end. The control problem is primarily kinematic rather than dynamic: to determine the best way to move the actuator, $x_0(t)$, so as to manoeuvre the end mass, at x_n, from rest in one position to rest in a new, target position. It is assumed that the final displacement of the end mass is equal to the final displacement of the actuator, so that if, for example, gravity is relevant, its net effect is identical at the beginning and end of the motion.

Although relatively simple in form, the arrangement in Fig. 1 exemplifies most of the inherent challenges in the problem and can be used to model many real systems. It is therefore a good starting point and test case. Note also that distributed systems are not excluded, because by choosing the number and value of the lumped parameters in Fig. 1 the dominant modal shapes and frequencies of a continuous system can be matched and thereby its essential dynamics and control also investigated. In any case, the solution to be presented applies also to distributed systems: in fact the wave ideas behind the proposed controller are actually more clear-cut when the systems are distributed.

Note that the controlled input is position rather than force, suggested in Fig. 1 by the form of the actuator at the left. Such a "kinematic" input is appropriate for many positioning applications (from disc drives to robotics) where supplying the force is not the main issue. The wave-based control ideas presented below apply to either position or force control. One cannot attempt to control both position and force at the input directly and simultaneously: one must choose one, and the system dynamics will determine the other. But the focus of the present paper will be on position input only. It is assumed that the actuator has its own position sensor and sub-controller that works over time, more or less rapidly, to set $x_0(t)$ to the value requested by the main control system, with zero steady-state (final value) error. It is implicitly assumed that it can supply the needed force. The main controller needs to know nothing of the details of this actuator sub-controller.

The control strategy for the simple system in Fig. 1 later proves applicable, with little or no adaptation, to a very broad class of problems. For example, the same strategy works for an arbitrary number of masses and springs; or when the load mass varies between manoeuvres (perhaps in an unknown way, as can happen for example in robotics); or when the system has distributed components or is predominantly distributed in nature; or with arbitrary internal damping (modelled e.g. as dashpots interconnecting any or all masses); or when the masses and springs remote from the actuator change in values; or when nonlinearities arise in the spring and damping characteristics. One strategy serves all cases, and does so remarkably well.

3. Notional separation of actuator motion into two components

The new control strategy involves separation of the actual actuator motion, $x_0(t)$, into two *notional* components, $a_0(t)$ and $b_0(t)$. Thus, formally

$$x_0(t) = a_0(t) + b_0(t). \tag{1}$$

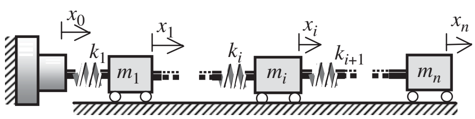

Fig. 1. The representative flexible system, with x_0 attempting to control x_n.

In a way to be explained below, the controller will specify $a_0(t)$ as an input component while determining $b_0(t)$ as a measure of the system response that gives a feedback-like component. It then adds the two components to specify the total input to the actuator, according to Eq. (1).

In this context exactly how a_0 and b_0 are defined is, to a small degree, arbitrary. Three formulations for a_0 and b_0 will be presented below (Section 4). Although not identical they are similar and lead to similar control performances. For the purposes of the control problem, these formulations could be accepted simply as *definitions* of a_0 and b_0, postulated rather than proven, and as such they require no derivation or theory. Nevertheless, some introductory and motivating arguments seem appropriate.

The variables a and b depend to some extent on the idea of "waves" imagined propagating leftwards and rightwards in the flexible system such as Fig. 1. The concept of waves in lumped systems is initially problematic, particularly if the system order is low. In lumped systems all components react immediately and continuously with all other components and many conventional wave concepts, featured in distributed systems, no longer apply. To address this possible concern two illustrative background approaches are offered. The first is more formal, based on operators, leading to "wave transfer functions" in the *s*-domain. The second is more physical, based on relating the real system to notional systems in which travelling "waves" are more identifiable. Neither presentation is intended to be rigorous or comprehensive. The background ideas are topics for further papers now in preparation.

3.1. Operator approach to mechanical wave analysis

One can begin by assuming the validity of the concept: that is, assume that the motion of each mass in Fig. 1 can indeed be expressed as the superposition of two notional wave components

$$x_i = a_i + b_i, \tag{2}$$

where a_i is the component of the position, x_i, of the ith mass, associated with a rightwards propagating wave, and b_i is the component associated with the leftwards wave. In other words, the a_i motion component is associated with motion initiated somewhere to the left and travelling from left to right, as if in a "one-way" system (that is, as if in an imaginary system extending indefinitely to the right). If so, this component motion will be related somehow to the corresponding rightwards component motion in the previous mass to the left. Assume this relationship can be described by some kind of operator $R_i(\cdot)$ so that

$$a_i = R_i(a_{i-1}). \tag{3}$$

Similarly, the b_i component is assumed to be associated with a leftwards wave and will be related to the corresponding component of the next mass to the right, or

$$b_i = L_i(b_{i+1}), \tag{4}$$

where again $L_i(\cdot)$ is an operator.

The proposed operators $R_i(\cdot)$ and $L_i(\cdot)$ defined in part by Eqs. (3) and (4) are a formal expression of the wave postulate. If the postulate is valid, then the operators will have certain features that help define them more completely. Firstly, if the system response must be given by the superposition of $a_i(t)$ and $b_i(t)$, then the superposed motion should obey the differential equations of motion of the system

$$[k_i]x_{i-1} - \left[k_i + k_{i+1} + m_i D^2\right]x_i + [k_{i+1}]x_{i+1} = 0, \tag{5}$$

when the substitutions of Eqs. (2)–(4) are made. Also the two component motions, present in the same system, should obey the same differential equations of motion. So, for example, for the rightwards wave beginning at a mass $i-1$, the operators should be such as to satisfy the equation

$$[k_i]a_{i-1} - \left[k_i + k_{i+1} + m_i D^2\right]R_{i-1}(a_{i-1}) + [k_{i+1}]R_i(R_{i-1}(a_{i-1})) = 0. \tag{6}$$

Finally, the system boundaries will impose further constraints on the nature of the operators R_i and L_i.

It transpires that these constraints define major features of the operators: the steady state gain should be unity; the instantaneous response should be zero; the dynamic response described by the operators should be close to damped second order, with the dominant frequency close to $\sqrt{(k_i/m_i)}$ and the phase lag increasing from zero to π with frequency. But within such constraints slightly different operators, and so different wave models, can perfectly model the dynamics of the lumped system when the a_i and b_i are superposed (Eqs. (2) and (5)).

Under any valid choice of model, the motion of the first mass, x_1, can be expressed as

$$x_1 = a_1 + b_1 \tag{7}$$

$$= R_1(a_0) + L_1^{-1}(b_0), \tag{8}$$

where $L_1^{-1}(\cdot)$ is an operator going one mass to the right for the leftward-going motion component, b_i.

If it is further assumed that R_1 and L_1 are symmetrical, that is, that one is the inverse of the other, then applying R_1 to Eq. (8) gives

$$R_1(x_1) = R_1^2(a_0) + b_0. \tag{9}$$

Combining this with Eq. (1) gives

$$a_0 - R_1^2(a_0) = x_0 - R_1(x_1). \tag{10}$$

From this equation, if $R_1(\cdot)$ is assumed, or is known, and x_0 and x_1 are taken as known or measurable inputs in the physical system, a_0 can be determined, at least implicitly. Then b_0 follows from Eq. (1).

In the Laplace domain the operators can take the form of transfer functions. For the special case of a uniform string of masses and springs, the ambiguity in the wave model can be removed by assuming that all the operators (transfer functions) are equal and that this operator corresponds to the transfer function between two masses in an infinite mass–spring string:

$$G(s) = 1 + 1/2\left(\frac{s}{\omega_n}\right)^2 \mp 1/2\frac{s}{\omega_n}\sqrt{\left(\frac{s}{\omega_n}\right)^2 + 4} \tag{11}$$

with

$$\omega_n = \sqrt{k/m} \tag{12}$$

and k is the spring stiffness and m the mass [11]. For reasons of causality, the negative sign should be chosen, ensuring that the transfer function remains finite at large frequencies and that the phase is lagging, as it must be. The frequency response $G(j\omega)$ confirms that this "wave transfer function" exhibits the features described above.

The non-uniform case is more complicated and the exact wave transfer functions become more complex. Nevertheless, they retain the main characterizing features, and excellent results are still obtained if the system is considered locally uniform, with local stiffness, k_i, and mass, m_i used in determining G in Eqs. (11) and (12).

3.2. Physical model approach

The second motivating approach involves imagining the real system to be expanded and "broken out" into new, notional component systems, with $a_i(t)$ and $b_i(t)$ then identified with motions in these notional component systems.

3.2.1. Wave components physically separated

Fig. 2 shows the original system, Fig. 1, with a second, mirror-image system appended. The load (or end) masses of the two systems are imagined joined to form a single mass which is therefore double the original load, with a single displacement. The rest of the second system (with primed displacement variables) is identical to the first, but in reverse order, ending in a second actuator which is an image of the first. The motion of this image actuator, indicated by x_0, is assumed to follow that of the first actuator exactly, in the sense that if at any instant the first actuator is moving to the right, the second is also moving to the right at the same speed.

Now if the equal motions of the two actuators in Fig. 2 are made identical to that of the actuator in the original system, it is readily shown that the response of the left hand side of the combined system in Fig. 2 will

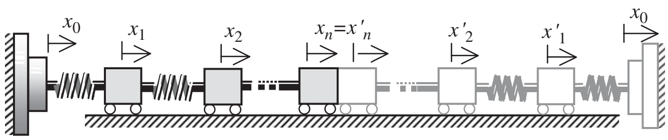

Fig. 2. Original system (left) with mirror-image system appended (right). If the displacement of the mirror-actuator (right) follows the original actuator x_0 then the primed displacements will equal the unprimed.

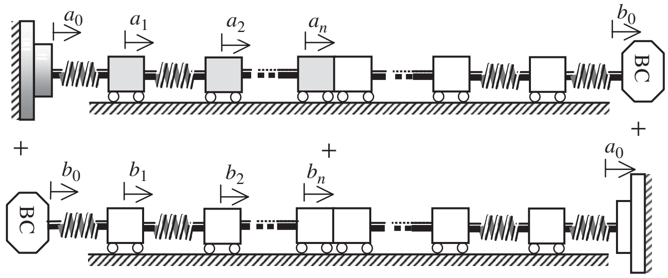

Fig. 3. Two perfectly symmetric systems assumed superposed to produce the displacements in Fig. 2. Thus $a_i + b_i = x_i$, $i = 0,1,\ldots,n$. ''Waves'' move from actuator to BC in both cases: i.e., in the second system, from right to left, the arrows indicating only reference directions for displacement.

be identical to that of the original in Fig. 1. In other words, the addition of the mirror system has no net effect on the motion of the first system.

Taking a further step, it is postulated that the system in Fig. 2 can in turn be considered as the superposition of two such double systems, each a mirror (or reversed-order) image of the other, as depicted in Fig. 3. In other words, the motion of each component in the original system will be given by the sum of the corresponding motions in the two systems in Fig. 3.

Now in the upper system of the pair in Fig. 3, ''disturbances'' of some kind (here loosely called ''waves''), with oscillating displacements, velocities, energies and momenta, can be considered as entering the system from the left-hand actuator. With appropriate boundary conditions BC, these will sooner or later leave the system to the right, having undergone more or less dispersion.

The opposite happens in the lower system in Fig. 3. The ''waves'' enter at the right and leave at the left. They cause the same net displacements as in the upper system, because the two actuators are specified to move simultaneously in the same direction, one pushing where the other pulls.

Understood in the lax sense of the previous paragraphs, the upper system can be considered to have ''one-way'' waves travelling left to right, the lower system having similar ''one-way'' waves travelling right to left, albeit dispersing as they go. These ''waves'' correspond loosely to the component motions a_i and b_i.

The notional ''break-out'' process (Fig. 1 through Fig. 3) can now be reversed. The two systems in Fig. 3 can be superposed, ensuring identical motion of each end, as required on reaching Fig. 2. Then the superfluous mirror system in Fig. 2 can be suppressed to get back to the real system in Fig. 1. By this device, the real system's motion in Fig. 1 can now be considered as the superposition of rightwards- and leftwards-going ''waves'', described by (or made separable and definable by) comparison with the separate, independent motions of the left-hand parts of the two systems of Fig. 3.

3.2.2. Wave-absorbing boundary conditions

For wave-based control, the notional secondary actuators BC should absorb vibrations out of the intermediate system caused by the primary actuators' motion: that is, they should provide damping. Furthermore they should not constrain the final position of the system: that is, if the actuators' motion approaches a steady value (e.g. a constant displacement, or velocity, or acceleration), the steady-state motion of BC should eventually approach the same value. Within these constraints many choices of BC are workable, leading to slightly different definitions of a_i and b_i.

Any boundary acting as a passive, viscous damping will satisfy the general constraints. A good choice of viscous damping constant is $\sqrt{(km)}$ where k and m correspond to the last spring and mass (which are the same as the first). It is conjectured that the "best" choice (giving fastest vibration damping) of BC for a long system is one which mimics a continuation of the string to infinity, beginning with a mirror image of the real system about the free end and repeating periodically. Waves launched at the actuator (causing both vibratory motion and perhaps a net displacement, or DC component) will then pass out of the system at the boundaries, never to return. In other words, the boundary condition maximises energy extraction and minimises energy reflection by minimising the dynamic impedance mismatch at the boundary. It also allows an arbitrary net displacement.

3.2.3. *Fig. 1 considered as superposition*

When the "break-out" process from Figs. 1 to 3 has been reversed, the motions of the two primary actuators and those implicit in the boundaries, BC, in Fig. 3 become incorporated into the (single) real actuator's motion in Fig. 1. The actuator in the real system, Fig. 1, can therefore be considered to be doing at least two jobs, seen more clearly in the left-hand sides of the notional component systems. Firstly it initiates motion, or launches a wave from the actuator into the system, from left to right. Secondly, it terminates the motion, or absorbs the wave returning from right to left. From another perspective, the first job corresponds to "pushing" the flexible system to start its motion, while the second motion, superposed on the first, corresponds to the actuator's being "pulled" by the response of the system in such a way as to dampen vibration.

But to achieve this dual action, the actuator's control system needs to be able separately to identify the notional component waves present at the actuator, that is, a_0 and b_0, by taking measurements in the physical system of Fig. 1.

4. Identifying $a_0(t)$ and $b_0(t)$ in the physical system

Three definitions, or methods of establishing, $a_0(t)$ and $b_0(t)$ within the real system, will now be presented. As already noted, they can be taken either with or without reference to, or dependence on, the ideas of Section 3 above. From here on, a and b (with or without explicit time dependence) will mean $a_0(t)$ and $b_0(t)$, and capital letters will indicate the corresponding Laplace-transformed variables. Subscripts are not needed because the focus will be on resolving only the actuator's motion, $x_0(t)$, into two components.

The first approach is expressed in terms of transfer functions. The s-domain versions of $a(t)$ and $b(t)$ are given by

$$A = X_0 \frac{1}{1-G^2} - X_1 \frac{G}{1-G^2}, \tag{13}$$

$$B = X_0 - A, \tag{14}$$

where G is defined by Eq. (11), using the first spring and mass values in Eq. (12).

While this first formulation is conjectured to be the "best" (at least for long, uniform systems), it gives rise to practical problems. It is very challenging to get the required time-domain values $a(t)$ and $b(t)$ from these equations as they stand. Ref. [11] solved the problem by using convolution with an impulse response corresponding to the time domain version of $G(s)$, truncated in time. This is computationally expensive and slow (although it remains a further option, in addition to those presented here).

A more practical approach is now presented, which will constitute the second option. It involves reformulating Eq. (13), as follows:

$$A = X_0 - GX_1 + G^2 A = X_0 - G(X_1 - GA) \tag{15}$$

with B from Eq. (14) as before. The required variable A is now implicit in Eq. (15) (appearing on both sides). But because the transfer function G on the right-hand side has zero instantaneous response, the time

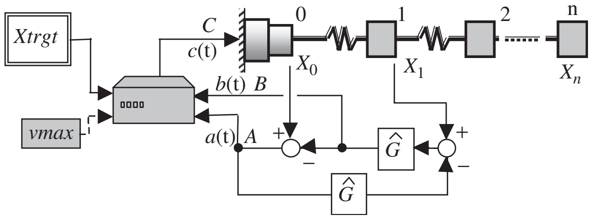

Fig. 4. The control system. $a(t)$ and $b(t)$ are the notional components of $x_0(t)$. The box executes the algorithm listed in Fig. 7 and portrayed in Fig. 6. For an ideal actuator, $x_0(t) = c(t)$.

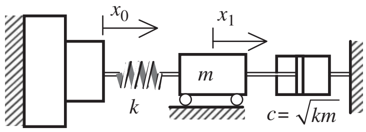

Fig. 5. A simple analogue for $\hat{G}$ (input x_0 and output x_1) for Fig. 4, with the k and m corresponding to the first spring and mass in Fig. 1.

implementation of this equation works perfectly well when a "one-time-increment-old" version of A is used on the right hand side, or when incorporated into a classical control arrangement in block diagram form, as in Figs. 4 and 13.

The wave transfer function G is not easily implemented exactly, especially in real time. If it were possible, an analogue of G could be obtained by making computer models of Fig. 1 extended "to infinity". The function input would then become the position of the actuator x_0, and the output, $G(x_0)$, would be x_1 in the model. In practice, the essential requirements are met by the response of much simpler systems, even down to a one-degree-of-freedom model such as Fig. 5. Two such analogues, denoted $\hat{G}$, are needed to evaluate continuously the time domain equivalent of two G terms in Eq. (15).

A third option, computationally the least demanding, determines $a(t)$ and $b(t)$ simply as (in each case) either

$$a(t) = 1/2\left(x_0(t) + \int \frac{f(t)}{Z}\,\mathrm{d}t\right) = 1/2\left(x_0(t) + \omega_n \int (x_0 - x_1)\,\mathrm{d}t\right), \tag{16}$$

$$b(t) = 1/2\left(x_0(t) - \int \frac{f(t)}{Z}\,\mathrm{d}t\right) = 1/2\left(x_0(t) - \omega_n \int (x_0 - x_1)\,\mathrm{d}t\right), \tag{17}$$

where $f(t)$ is the force at the actuator, acting in the first spring, and Z is an impedance value of $\sqrt{(km)}$, corresponding to the first spring stiffness, k, and first mass, m. This option also meets the control implementation requirements and works surprisingly well when incorporated into the control systems. Depending on which of the two forms is used, the second measured variable will be force, f, or position x_1.

5. Further preliminary ideas

Any of the above definitions of $a(t)$ and $b(t)$ can be taken for what follows. It will be assumed for the moment that the actuator is ideal (a zero order, unity gain response), achieving the position $x_0(t)$ in zero time when requested to do so by the controller.

5.1. At rest, launch and absorbed displacements are equal

For rest to rest manoeuvres, in the absence of extraneous forces, when the system comes to rest again, say at final time t_f, the final values of $a(t_f)$ and $b(t_f)$ will equal each other, and they will each equal $\frac{1}{2}x_0(t_f)$. Or, putting it the other way around, if one ensures that, for times beyond some time t_f,

$$a(t) = b(t) = \tfrac{1}{2}x_0(t) = \text{constant} \quad \forall t > t_f, \tag{18}$$

one has thereby also ensured that the entire system, including the load at x_n, is at rest at this time, at a final constant displacement of $x_0(t_f)$. In words, at steady state the net displacement due to absorbing the return wave is equal to that due to the launch wave. This result is independent of how x_0 attained its final value, that is, it is independent of the time history of $a(t)$ prior to $t = t_f$. It also applies to all the definitions of $a(t)$and $b(t)$ introduced above.

This result, Eq. (18), is implicit in the condition that, in the absence of other external forces, the integral of the actuator force on the system in going from rest to rest must be zero. For example, if the wave definitions in Eqs. (16) and (17) are used, and x_0 set to a constant with the integrals set to zero, Eq. (18) follows. When the waves are defined using G, the argument is more subtle, but the same result can be derived. With reference to the approach in Section 3, if the upper system in Fig. 3 moves from rest in one position to (eventual) rest in a new position, say, 0.5 m to the right, the lower system will do the same, so that when the motion of the two systems is superposed, there will be a net, rest-to-rest motion of 1 m.

5.2. Component $b(t)$ also dampens vibration and lags $a(t)$

Eq. (18) assumes that all motion has ceased, including vibrations. If $a(t)$ is held steady, while the calculated motion $b(t)$ is superposed to produce the actuator motion $x_0(t)$, the $b(t)$ component will have the required effect of damping out all vibratory motion and ensuring stability. This can be seen, for example by differentiating Eqs. (16) and (17) with respect to time, where the effect of superposing b in x_0 is to introduce a viscous damping velocity component $\frac{1}{2}f(t)/Z$ in the motion of x_0. The more dynamic boundary conditions, based on the $G(s)$ definitions of a and b, achieve damping even more effectively. A more physical insight can be obtained from Fig. 3, where the effect of the $b(t)$ component in $x_0(t)$ is to model the boundary condition, BC, which is designed to absorb vibration continuously.

A second feature of $b(t)$ is that it will always lag any assigned value of $a(t)$ by a finite time. This can be seen, for example, by applying the initial value theorem to $G(s)$ or to $\hat{G}(s)$, or, if using Eqs. (16) and (17), by assuming a jump in the acceleration of x_0 which can be shown to leave b instantaneously unchanged. The delay is always significant, and, the longer and more flexible the system, the greater it is. Thus the controller has no practical difficulty in setting the input to the actuator to be the sum of a+b, because it can change a as rapidly as desired (and therefore x_0) without having instantaneously to change b as well. Also, initially x_0 is simply a. As noted, this delay also make Eqs. (10) and (15) effectively explicit.

To summarise so far: the assigned motion, a, *pushes* the system. In the process of simultaneously adding in b (the system response which is delayed with respect to a) the system, through the controller, *pulls* the actuator. But it does so "gently", with just the right amount of "give" to dampen or absorb the vibrations actively. Furthermore, if the push from the a component moves the system a given distance and then holds, the process of adding b eventually doubles the total distance moved while bringing the system to rest. These ideas provide the key to combining position control and active vibration damping. Control systems based on these ideas give great results. But there is one further refinement, based on another discovery, which can if desired provide a small additional improvement to the control performance.

5.3. The "best" launch wave, $a_0(t)$

The form of a chosen by the controller is arbitrary, provided only it has the correct final value. A step, or ramp or even a parabola, up to the desired final value ($\frac{1}{2}$ target), are fine and work very well, as will be seen. But the very arbitrariness gives scope for even further refinement.

One might guess that the "best" final shape of $a(t)$ would involve maximising the final deceleration so as to tend to minimise the transit time. Thus, whereas at the start-up the actuator did its best, in the circumstances, to get the load moving, now it needs to do the opposite in the same circumstances. These "circumstances" include the actuator limitations, the load inertia, and, above all, the flexible nature of the intermediate dynamics that decouple actuator and load. This suggests trying to replicate a time reversal of the start-up, as the best target-arrival performance that is physically possible.

It turns out that, fortunately, precisely the information needed to do this is contained in the system response, $b(t)$, from start-up, which can be recorded easily.

The idea is conveyed in Fig. 6. The controller sets the launch part of the actuator's motion, a, to grow with time, perhaps as a ramp. (Neither the shape nor the slope are critical.) Meanwhile, and throughout, b is determined by knowing x_0 and observing x_1 (or f) and using one of the methods described. The controller adds b to a to determine the actual actuator motion, x_0. It also stores the values of b over time.

At some point, denoted $t = t_1$, the actuator's current position, $x = a+b$, will equal half the target distance (here taken as unity). At this point the launch wave, a, is short of its ultimate value (in this case $\frac{1}{2}$) by the current value of b. The controller then completes the launch wave a by "playing back" the value of b it has recorded up to this point, in reverse, and inverted. This ensures that when the playback has got back to the earliest value of b (which is always zero), the final value of a will be steady at its correct value ($\frac{1}{2}$). This also means that b must reach its correct final value ($\frac{1}{2}$), and so the entire system including the load will stop at the target.

But more importantly (assuming enough of b has been recorded by the changeover time t_1), on arrival at target, the load will stop dead. In fact, the load lands at the target first, stops, and remains stationary while the rest of the system completes the manoeuvre, with the actuator coming to rest last of all. Thus, having strained the system to cause the load to stop on target, by a slight deceleration, the actuator then continues to move to allow the flexible system to "unwind" or "relax" in just the right way, leaving the load undisturbed. The perfection of this action, easily achieved, is a delight to observe in simulation and in experiment.

How much echo has been recorded when t_1 is reached depends on the actuator launch speed, a, the manoeuvre length, and the length of the flexible system. Of these, a is completely controllable, and, if necessary, can be reduced to ensure that sufficient echo has been received at t_1 to allow vibrationless arrival at target. On the other hand, if a is set to grow at the maximum rate possible, regardless, the control system still works splendidly. The only price is a small initial overshoot (typically 5%) and then some small residual vibration on arrival at target, which quickly decays to zero. See Fig. 9, where the effect of different launch speeds, da/dt is illustrated.

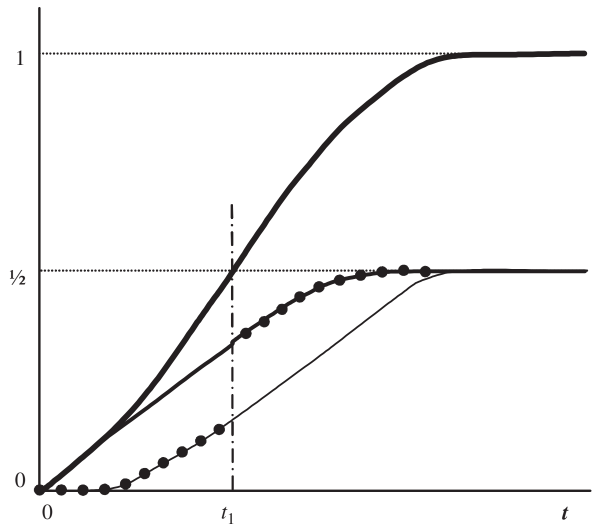

Fig. 6. The notional separation and recombination of displacement x (heavy line) by way of a and b (light lines). Target distance is 1. a is *set* by the controller as a ramp i.e., $a = \frac{1}{2}v_{\max} \times t$,, until $t = t_1$ (when $x = \frac{1}{2}$) then as a replay of previous b, but inverted, i.e. $a = \frac{1}{2}–b(2t_1–t)$ (shown dotted). At all stages b is determined from the system response. Actuator, x: ▬▬ $a(t)$: —— $b(t)$:——*replay*: ●–●–●–●–.

6. The control system algorithm

The strategy conveyed in Fig. 6 does not allow presentation in classical control form. Fig. 4 instead shows a box that executes the corresponding algorithm. It takes $a(t)$ and $b(t)$ as inputs (determined by any of the methods) and produces the actuator position request, $c(t)$, as output. Two further inputs are the target displacement, x_{trgt}, and a maximum actuator velocity, $v_{\max}$. The essence of the algorithm to be evaluated at each time step, t, is as in Fig. 7. The variable $r(t)$ is the assigned actuator input before superposing $b(t)$. For an ideal actuator, $r(t) = a(t)$, and $x_0(t) = c(t)$. (The issue of non-ideal actuator response will be considered briefly below.) Lines 1–10 determine $r(t)$, for all stages, and line 11 adds $b(t)$, again for all stages.

Going through the code, in stage 1, a growing displacement $r(t)$ (here a ramp, the time integral of $\frac{1}{2}v_{\max}$, line 2) is launched into the system until the accumulated echo is sufficient to get the launch position to half the target position, at which point (stage 2, line 6) the echo is played back (or strictly speaking, the final value minus the echo reversed in time). In case the response is not perfect, perhaps due to poor actuator response, and to remove any residual vibration, the algorithm ends by implementing the arrangement described below in Fig. 13 (line 9 with line 11). The "stage" variable (lines 1 and 7) is merely to avoid re-entering the first phase once completed, due to possible small variations in $a(t)$ and $b(t)$. It is not always necessary.

The entire control system shown in block diagram form in Fig. 4, including the analogues for $\hat{G}$, would in practice be modelled within the same controlling computer. So the feedback inputs to the controller will then be $x_0(t)$ and $x_1(t)$ measured on the real system.

7. Sample results

Fig. 8 shows the performance of this algorithm applied to a numerical model of a uniform three-mass system. The actuator and end-mass positions are shown against time expressed in units of the period, T, or $2\pi\omega_n$. The target displacement is 1 m. Also shown are $a(t)$ and $b(t)$.

As can be seen, the response is impressive. The load (end mass) is translated from rest to rest, in a single, controlled movement, with almost no overshoot and with negligible oscillations (and so little or no settling time). Depending how strictly one defines the settling time, the total manoeuvre time is between 3 and 3.5 "periods" of ω_n, This corresponds to about only 1.5 periods of the fundamental mode of the 3-mass system, which is rapid indeed.

The end mass (or load) comes to rest exactly at target, to an accuracy corresponding to that of the actuator position sub-controller. Significantly, it does so sooner than the actuator that is controlling its motion, remotely. Around mid-manoeuvre, the speed of the end-mass (its slope in Fig. 8) is close to that of the actuator: the flexible system is then behaving as if rigid, or almost so. The accuracy of the final load position is limited only by the accuracy of the actuator's final position, which generally can be very high.

Throughout the motion the actuator's movements are smooth and easily achievable, with no necessity to achieve high jerk or even high acceleration. The actuator acceleration can be explicitly limited to realistic values without a noticeable degradation in the overall response. The overall control strategy accepts whatever

```
1   IF    a(t)+ b(t) < ½ xtrgt …    Still in 1st launch stage?
    AND   stage=1                   Without having left it?
2         r(t) = ∫(½ vmax)dt        Assign i/p (launch) position
3         echo(t) = b(t)            Record echo b(t)
4         t1 = t                    t1 marks end of 1st stage
5   ELSE IF (2t1-t)>0               Not yet run out of echo?
6         r(t) = ½ xtrgt            Set i/p = final value minus
                 -echo(2t1-t)       time-reversed echo
7         Stage=2                   Avoids re-entering stage 1
8   ELSE                            End of echo?
9         r(t) = xtrgt – a(t)       Set i/p as in Fig.11
10  END
11        c(t) = r(t) + b(t)        For all i/p "cancel" b(t)
```

Fig. 7. Algorithm executed by the computer in Fig. 4 for rest to rest motion. $a(t)$, $b(t)$ are determined outside this algorithm for each time step t, by any of the methods described. The variable *stage* is initialised to 1 outside the control loop. x_{trgt} and $v_{\max}$ are specified elsewhere.

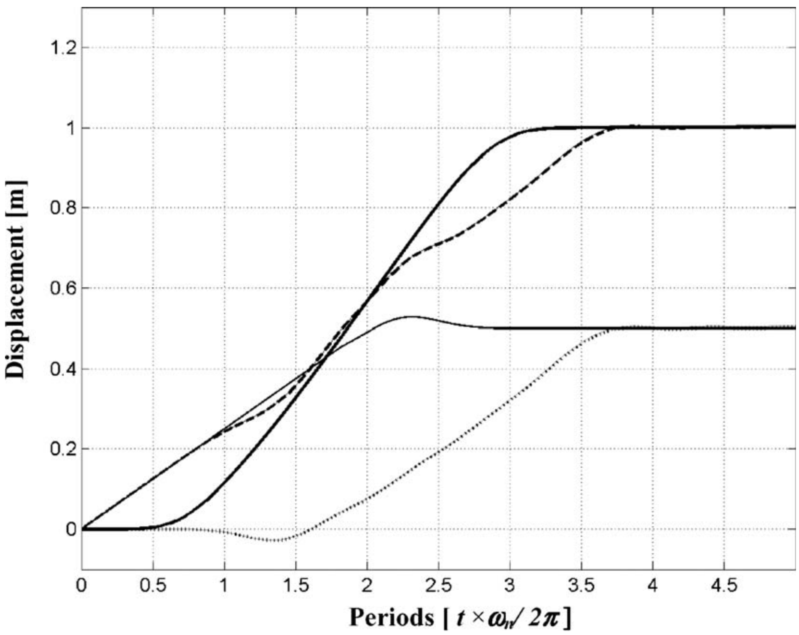

Fig. 8. End point (heavy line) of a uniform, three-mass system, moved 1 m. End point: ▬▬ Actuator: – – – – $a(t)$:—— $b(t)$: ⋯⋯.

the actuator can manage and works within this to reach its goal. The symmetry of the motion also helps the self-compensation of any non-ideal actuator responses.

The variable v_{max} acts as a speed control and sets the slope of $a(t)$ and therefore the speed limit of the response for the main part of the motion. In practice, the actuator "speed limit" may not be a constraining factor. In this case v_{max} can be considered a variable that allows a classical control trade-off, to suit the application, between, on the one hand, the end-mass rise-time, and, on the other, the degree of overshoot and settling time. Fig. 9 shows some examples for a uniform three mass system of the effect of different v_{max} values (and therefore $a(t)$ rise rate).

Setting v_{max} very high shortens the rise-time. But even in the "worst" case with v_{max} effectively unlimited ($100_{trgt}/T$ in Fig. 9), and therefore $a(t)$ taking the form of a step input, the overshoot and the amplitude of the residual oscillations are small and quickly die out. Going in the other direction, reducing v_{max} increases the manoeuvre time, but soon achieves negligible overshoot and negligible residual vibration.

If the actuator speed limit is a physical constraint, then the controller's v_{max} can be set at this value, with $a(t)$ therefore ramping at half this value. For the main part of the motion, with $\mathrm{d}b/\mathrm{d}t = \mathrm{d}a/\mathrm{d}t$, the actuator and load will then approach the maximum actuator speed, or $\mathrm{d}b/\mathrm{d}t + \mathrm{d}a/\mathrm{d}t$, which is the maximum speed it would have if the system were rigid. The absorbing action will still work without "hitting" the speed limit because the maximum and limiting value of $\mathrm{d}b/\mathrm{d}t$ to be added to $\mathrm{d}a/\mathrm{d}t$ will also be half of v_{max}; and there will be even less residual vibration on arrival at target.

Remarkable as all this may seem, the good news does not end there. The new control strategy is found to be surprisingly robust and self-adapting. It has no difficulty coping with significant changes in the system. Fig. 10 for example shows the result with the end/load mass increased by three (case 1) and then reduced to one third (case 2), but without changing a single control parameter from the uniform case of Fig. 8.

The strategy also works for any number of degrees of freedom, large or small. Going for the smallest, Fig. 11 shows a response for a one mass system. The only adjustment to the controller was to set v_{max} to $(1.5)_{trgt}/T$ for the case shown. If preferred, the tiny overshoot and settling can be yet further reduced, while still getting the mass rapidly to target, by simply reducing v_{max} a little (cf. Fig. 9 for 3 dof case).

Other such changes in the internal dynamics of the flexible system, which, with other strategies, often require a complete rethink, are here handled automatically. For example, add internal damping or make the springs beyond the first one nonlinear (hardening or softening), and the same strategy still works very well,

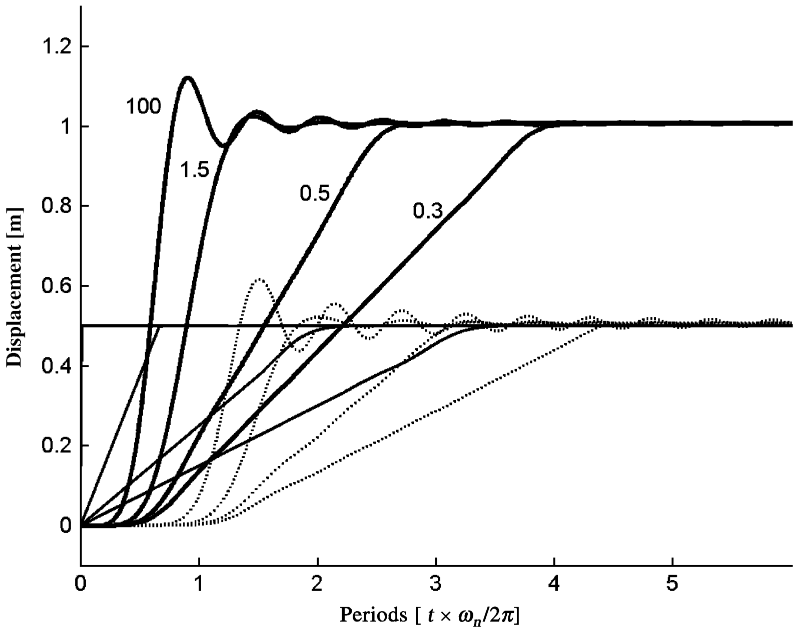

Fig. 9. End mass response (solid heavy line) for values of v_{max} of 100, 1.5, 0.5 & 0.3 times trgt/T. These determine da/dt in Stage 1. $a(t)$ (solid fine) and $b(t)$ (dotted) also shown for the four cases. For largest v_{max}, $a(t)$ is effectively a step input.

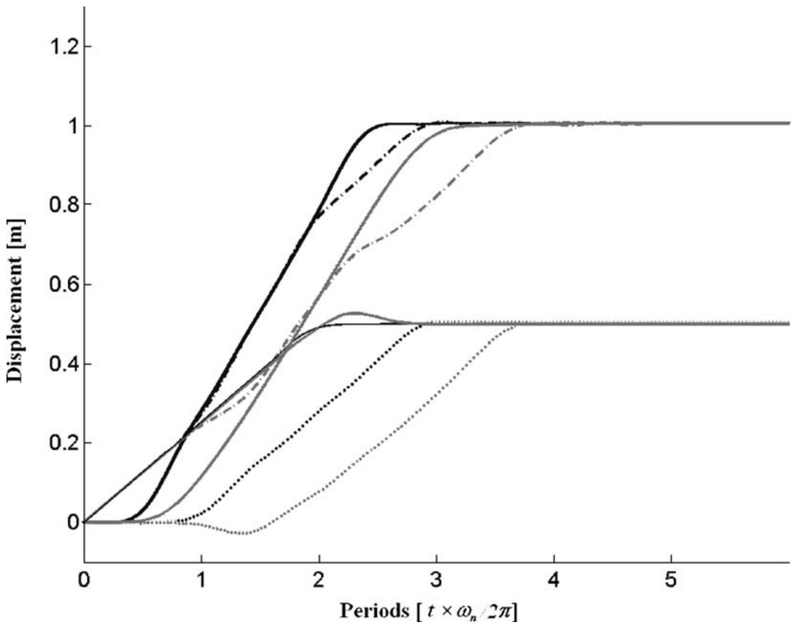

Fig. 10. Three masses with end mass multiplied by 3 (case 1—grey) and by 1/3 (case 2—black) and controller parameters unchanged from those of the uniform case (Fig. 8). End point: ▬▬ Actuator: – – – – $a(t)$:—— $b(t)$: ······.

without even the need to re-tune parameters. The only necessary condition is that the system should return to its initial length at steady state. If part of the system is continuous rather than lumped, again there are no new control issues to grapple with.

For an ideal actuator, $x_0(t) = c(t)$ and $r(t) = a(t)$. But a real actuator will take time to respond, so these equalities will apply only at steady state. A further extraordinary feature of the new algorithm is that it still

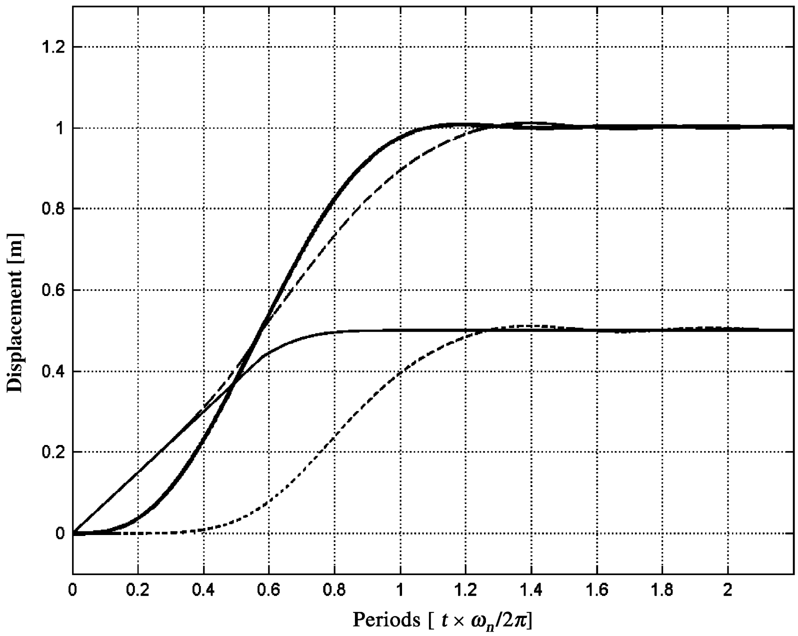

Fig. 11. Response of 1 dof system with $v_{max} = (1.5)_{trgt}/T$.

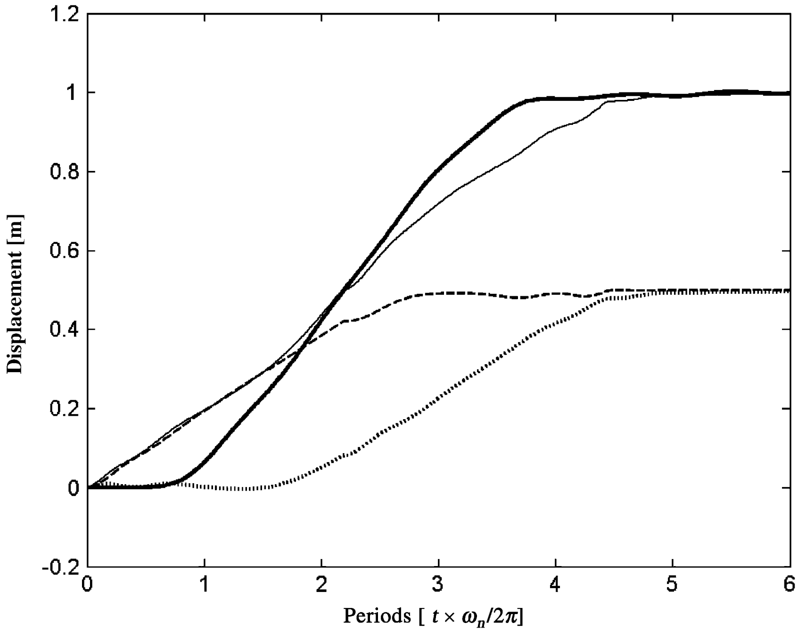

Fig. 12. Response of end mass of 5 dof system, with non-uniform masses, various dampers between the masses, nonlinear springs, a first-order actuator, and a simple approximation for $a(t)$ and $b(t)$. End point: ▬▬ Actuator:—— $a(t)$: – – – – $b(t)$: ⋯⋯.

performs well with far from ideal actuators. Extensive testing was carried out with first and second-order actuators, and in simulation with and without back loading on the actuator from the flexible system dynamics. It was found that provided the actuator steady-state position error is zero (which is easily achieved) and provided its bandwidth is about 15% higher than the highest natural frequency of the system, the performance deterioration in comparison with an ideal actuator is slight and the end conditions are still achieved. (For a uniform system of any length, the highest natural frequency is $2\omega_n$.)

As a final example, illustrating a mixture of such added complexities, Fig. 12 shows the response of a 5 dof system with nonlinear (hardening) springs; variations of the masses of 1, 0.5, 1, 2, 1; damping between the masses of 0, 0.25, 0.1, 0.25, 0, 0.05 times critical damping; an actuator modelled as a first-order system of time constant $1/3\omega_n$; and $a(t)$ and $b(t)$ approximated simply by Eqs. (16) and (17). These values and the system size were chosen almost at random: a similar result is obtained for almost arbitrary choices of these variables.

8. Open-ended, varying input control

The main topic of this paper is controlled motion through a desired distance from rest to rest, which is the most common requirement in practice. But for completeness, it is noted that the strategy outlined above can be adapted for ''open-ended'' control, where the final position may not be known beforehand (as happens for example with hand-and-eye control by an operator or patient); or where the input may be arbitrary or unpredictable; or where the system may not be at rest initially. With the target position now unknown, the full benefit of the wave-echo idea is no longer available, but the underlying wave-based strategy still works very well, retaining most of the featured advantages. Fig. 13 shows an arrangement for this case, in which input $X(s)$ (or $x(t)$) is arbitrary. The response to a step input for this arrangement is almost identical to that of the high $v_{\max}$ curve in Fig. 9: no longer ''vibrationless'', but still an excellent response.

Finally, Fig. 14 shows the corresponding system based on Eqs. (16) and (17), which leads to a particularly simple arrangement that still delivers a remarkable performance [20].

9. Discussion

Many configurations in addition to those shown above have been simulated and comprehensively tested, in all cases giving similarly impressive responses. The same control strategy (or with minor variations of the same basic idea) has also been demonstrated experimentally on a gantry crane model [13,14], again with excellent results. In summary, a control system has been developed that proves powerful, flexible, robust, generic, and that produces near ''perfect'' results. While the reasoning that led to it may involve subtleties, the final control system is remarkably simple to implement and very modest in its hardware requirements.

Because of the novelty of the approach, it is difficult to line it up fully with standard control theory. Some further explanatory comments may however be helpful.

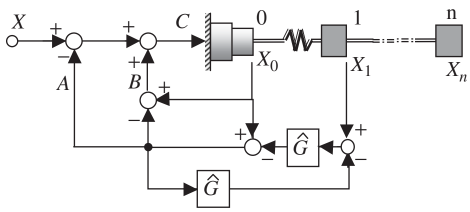

Fig. 13. The strategy adapted for open-ended control, with arbitrary input X.

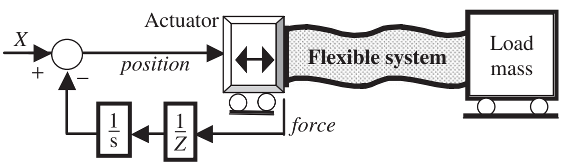

Fig. 14. The same adaptation as Fig. 13, but based on Eqs. (16) and (17).

9.1. Energy

From an energy perspective, the control system contrives to get energy into the system, necessarily as kinetic and potential in about equal measures [21], then works to convert it all to kinetic energy, then extracts it all again involving reconversion to potential and kinetic, arriving at the target at the instant the energy returns to zero.

This method of controlling the energy serves several purposes at once. It provides an energy loading procedure that packs the energy into the system in an orderly, minimally disruptive way. It is one that is easily and predictably reversed ("unpacked") at the end. It dynamically suppresses vibration. Furthermore, it causes the system to reveal its entire dynamic signature as seen from the actuator's perspective. In addition, this revelation is in precisely the form the actuator needs, without further processing, later to stop the load dead, at the target, by allowing the remaining energy to flow back out of the system as the load lands on target, and for a short while afterwards as the system relaxes back to steady state.

9.2. No model needed

As a consequence, little or no system model or system characterisation is needed. Or rather, the system response is itself serving as the controller "model", observed and recorded for later playback. As the system size or parameter values change, so will the recorded signal, automatically. No adjustments are needed. The order of the controller automatically matches the order of the system. In the same way, the controller automatically takes care of many non-ideal and nonlinear effects, such as spring hardening or softening (other than in the first spring), or more or less internal damping, linear or nonlinear. (If G is used to determine a and b, the first spring alone does need to be linear because Eq. (15) assumes linear superposition between x_0 and x_1. However, if Eqs. (16) and (17) are used, even the first spring can be nonlinear.)

The symmetry of the strategy also ensures robustness to limitations in the actuator dynamics, to sensing errors, and to dynamic errors in the determination of a and b. An imperfect dynamic measure of b, and/or imperfect absorption of b by the actuator, will lead to a small degradation in the performance, seen as slightly longer absorption and settling times. But the performance degradation is remarkably gradual, and there will typically be a cancelling error in a, ensuring that Eq. (18) will still apply. This in turn ensures that the overall strategy still gives excellent vibration control and a steady-state position error no greater than that of the actuator's position sensor. (The issue of any *static* offset error in measuring x_1, and therefore in the steady value of b, will be considered below.)

The only element of "tuning" of the controller (or system "modelling") is to attempt to match the dynamics of the flexible system close to the actuator. But even here the overall performance degrades remarkably slowly as the real system values (k_1 and m_1) depart from the assumed values (used to choose ω_n or Z) in the controller. Again, this can be partly explained by a cancelling effect between the estimates of a and b, and by the fact that the only critical requirements are that b dampen the motion while settling to a value equal to a. If these are fulfilled even slowly, achieving the main goal is always guaranteed.

9.3. Waves

While presented here for lumped systems, wave-based control methods work even better for completely distributed (continuous) systems, in which, because of finite propagation delays, wave concepts are unambiguous and more easily measured.

Although the focus above has been on displacement waves, similar results are obtained if one considers force waves, or velocity waves, or acceleration waves. In much the same way, actuator force, velocity, or acceleration can be notionally separated and recombined to effect wave-based control. In this way the control strategy can easily be adapted for use, for example, as a tip force controller, in applications such as robotic machining or assembly.

9.4. Action and reaction

In a perfectly rigid system, the actuator interacts directly and instantaneously with the load. But in a flexible system, the action and reaction of the actuator force (in the Newtonian sense) are only with the part of the flexible system dynamics next to the actuator. Thus the actuator has ceased to interact directly with the load: all the interaction is mediated by the flexible dynamics. Recognising this, the new control strategy focuses entirely on the only part of the flexible system the actuator is controlling directly, but uses this focus to recover control of the remote load over time.

Direct and instantaneous action and reaction between actuator and load in a rigid system are thus replaced by mediated and time-extended action and reaction in the flexible system. Furthermore, the strategy ensures that the accumulated "action", $a_0(t)$, and delayed system "reaction", $b_0(t)$, become equal on completion of the manoeuvre. If considered as force waves, this might be considered a generalisation of Newton's third law, with action and reaction between actuator and load being equal and opposite, but only over time and on reaching steady state.

As the system becomes more and more rigid, the delay between b and a eventually becomes negligible. Then each can be taken as half the actuator's value at all times, which nicely recovers Newton III for the actuator-load interaction, as a limiting case as the flexibility is removed: the actuator now "feels" the load directly, and action and reaction are once again instantaneously equal. This interpretation is clearest when a and b are defined by Eqs. (16) and (17) with $k_1 = \infty = Z$, or $x_0 = x_1$ for all t.

9.5. Transit time

The new strategy "settles" the system at target within, say, $1\frac{1}{2}$ periods of the fundamental mode of vibration of the system. (The exact manoeuvre time depends on the exact definition of "settling time".) As a rest-to-rest manoeuvre time this is very fast.

A point of comparison could be the widely studied time-optimal solution for rest-to-rest motion under a specified maximum actuator acceleration [16–19]. Such a solution is "bang–bang"; that is, the actuator is always at its maximum acceleration, and the control problem becomes one of specifying the exact switching times between full positive and full negative accelerations.

This whole issue is beyond the scope of this paper. Generalised comparisons between time-optimal and wave based strategies are difficult because so much depends on manoeuvre length, system order, system natural frequencies, and assumed acceleration limits. In addition bang–bang control demands precisely zero residual vibrations on arrival, whereas the wave-based control can easily tolerate small residual motion, which it then quickly absorbs. But, in so far as one can claim to be comparing like with like, the wave-based control, within the same acceleration limit, can achieve a manoeuvre time that is typically no more than 5–10% greater than the time optimal (using bands of $\pm 5\%$ or $\pm 1\%$ of final value, respectively, in defining the "settling time"). Yet the wave-based strategy does not need the accurate switching times, nor the high jerk, nor the accurate system model, nor the complicated problem solving, nor the ideal actuator, all required by time optimal solutions.

9.6. Environmentally friendly manoeuvring

In general forces are less in flexible systems than in stiff. In flexible systems, the actuator is pushing a softer system, and does not "feel" the load inertia directly (but over time).

An additional feature here is response to an unexpected external force. When on an experimental rig the end-point is pushed externally, the absorption process enables the system to "give" gracefully, much like power-assisted steering. Depending on the application, the controller can then be programmed either to accept the new position (e.g. in applications such as medical robotics, or with a ship's crewman manually positioning a crane's hanging load), or detect the resulting position error and move to correct it (e.g. in industrial automation).

The wave absorption feature also makes the system inherently stable. It automatically tends to absorb external shocks or vibrations, for example due to an unforeseen collision with an obstacle.

9.7. Errors

Under the assumption that the initial and final extensions are equal, the final, tip position accuracy is determined primarily by the accuracy of the actuator position controller, which in practice can be as good as available technology allows. Also, determining the switching time, t_1, is straightforward: it is when the actuator is half way to the target position. In any case, the exact value is not critical.

The only other variable to be measured is x_1, the position of mass 1 (or equivalently, the force, f, in the first spring, or the relative displacement x_0–x_1). Whereas in simulation no problem arose, in the experimental rigs even a small zero error in the final measured value of x_0–x_1, or of f, was found to cause a slow drift from the target position after arrival. This is due to an integration of the zero error over time, causing $b(t)$ to drift (cf., e.g. Eq. (17)). However, as this drift affects the entire system, it is readily detectable in the actuator position sensor. One solution then is simply to turn off the wave absorption soon after arrival at target. A more elegant solution has recently been developed and will be presented separately.

10. Summary, conclusions and further work

Aspects of the problem of controlling flexible systems have been thought about in new and fruitful ways. These lead to a new control algorithm that performs extraordinarily well. It easily moves a load from point to point, rapidly, yet with negligible residual vibration and negligible overshoot and zero steady-state error. It moves the load at close to the actuator velocity (the ideal), in one controlled motion, without exciting load or system vibrations unnecessarily. The control strategy is robust, applicable to a wide variety of problems, requires minimal system information, little computational overhead, and is very tolerant of limitations in the actuator dynamics. Sensing requirements are also minimal, and all sensing is done at, or close to, the actuator, which is normally the "clean" and accessible part of the system.

Modelling errors hardly feature. System changes are automatically accommodated. The order of the controller automatically matches that of the system, and explicit information, for example about locations of poles (or natural frequencies and damping ratios of modes), is not needed. The real system is also the controller's main "computer", as it "calculates" the exact echo profile, which the controller simply observes and records for later use, most effectively.

The strategy involves minimal dynamic decomposition. Where other approaches focus on n state variables (two for each mass), or on modal decomposition (with or without modal truncation), or on separation into rigid-body and flexible modes (with subsequent concerns about mutual coupling between them), here only the actuator motion is decomposed, and then into only two notional components over time, based on how the system is responding. Then, using a dynamic feedback of one of these components, in a computationally simple scheme, the entire motion becomes one, controlled, almost vibrationless, sweep of the load from rest to rest.

All aspects of the new strategy merit further analysis and investigation. More formal proofs are needed for some of the results. This work is well under way. Work is also advanced on applying and adapting the wave ideas to more complex systems, including systems with multiple actuators, whether in series or in parallel, systems undergoing flexural (beam-like, lateral) vibrations, and systems with significant external forces active during manoeuvres. An adaptation of the wave-based idea is also proving very effective in achieving an actively stabilized platform on a randomly moving base.

In summary, while wave-based control does involve some subtleties and unconventional notions, it is essentially simple, intuitive, powerful and easy to implement, and is proving adaptable to an increasing range of application areas.

Acknowledgements

This work was funded in part by Enterprise Ireland Basic Research Grant, code SC/2001/319/.

References

[1] L. Meirovitch, *Dynamics and Control of Structures*, Wiley, New York, 1989.
[2] A. Preumont, *Vibration Control of Active Structures*, Kluwer Academic Publishers, The Netherlands, 1997.
[3] W.J. Book, Controlled motion in an elastic world, *Journal Dynamic Systems, Measurement, and Control—Transactions of the ASME* 115 (1993) 252–261.

[4] T.L. Vincent, S.P. Joshi, Y.C. Lin, Positioning and active damping of spring–mass systems, *Journal Dynamic Systems, Measurement, and Control—Transactions of the ASME* 111 (1989) 592–599.
[5] S. Jayasuriya, S. Choura, On the finite settling time and residual vibration control of flexible structures, *Journal of Sound and Vibration* 148 (1) (1991) 117–136.
[6] W.J. Book, Recursive Lagrangian dynamics of flexible manipulator arms, *International Journal of Robotics Research* 3 (3) (1984) 87–101.
[7] I. Yamada, M. Nakagawa, Reduction of residual vibration in position control mechanisms, *Journal of Vibration, Acoustics, Stress, and Reliability in Design—Transactions of the ASME* 107 (1985) 47–52.
[8] J.J. Feliu, V. Feliu, C. Cerrada, Load adaptive control of single-link flexible arms based on a new modelling technique, *IEEE Transactions on Robotics and Automation* 15 (5) (1999) 793–804.
[9] T. Singh, G.R. Heppler, Shaped input control of a system with multiple modes, *ASME Journal Dynamic Systems, Measurement, and Control—Transactions of the ASME* 115 (1993) 341–347.
[10] W.E. Singhose, N.C. Singer, W.P. Seering, 1994, Design and implementation of time-optimal negative input shapes, *Proceedings of the 1994 International Mechanical Engineering Congress and Exposition*, Chicago, IL, USA, ASME Dynamic Systems and Control Division (publication) DSC 55-1, 1994, pp. 151–157.
[11] W.J. O'Connor, D. Lang, Position control of flexible robot arms using mechanical waves, *Journal Dynamic Systems, Measurement, and Control—Transactions of the ASME* 120 (3) (1998) 334–339.
[12] Z. Abduljabbar, M.M. El Madany, H.D. Al-Dokhiel, Controller design of a one-link flexible robot arm, *Computers & Structures* 49 (1) (1993) 117–126.
[13] W.J. O'Connor, Gantry crane control: a novel solution explored and extended, in: *Proceedings of the American Control Conference 02*, Alaska, 8 May 2002.
[14] W.J. O'Connor, A gantry crane problem solved, *Journal Dynamic Systems, Measurement, and Control—Transactions of the ASME* 125 (4) (2003) 569–576.
[15] V.V. Karolov, Y.H. Chen, Controller design robust to frequency variation in a one-link flexible robot arm, *Journal Dynamic Systems, Measurement, and Control—Transactions of the ASME* 111 (1989) 9–14.
[16] P. Meckl, W. Seering, Active vibration damping in a three-axis robotic manipulator, *Journal of Vibration, Acoustics, Stress and Reliability in Design—Transactions of the ASME* 107 (1985) 38–46.
[17] L.Y. Pao, Minimum-time control characteristics of flexible structures, *AIAA Journal of Guidance, Control and Dynamics* 1 (1996) 123–129.
[18] M. Muenchhof, T. Singh, Jerk limited time optimal control of flexible structures, in: *Proceedings 2001 ASME International Mechanical Engineering Congress and Exposition*, November 11–16, 2001, New York, IMECE2001/DSC-24560.
[19] R.D. Robinett III, C.R. Dohrmann, G.R. Eisler, J.T. Feddema, G.G. Parker, D.G. Wilson, D. Stokes, *Flexible Robot Dynamics and Controls*, Kluwer Academic/Plenum, New York, 2002 (p. 165 & passim.).
[20] W.J. O'Connor, C. Hu, 2002, A simple, effective position control strategy for flexible systems, International Federation of Automatic Control, *Second IFAC Conference on Mechatronic Systems*, December 2002, Berkeley, California, USA, pp. 153–58.
[21] W.J. O'Connor, The lost energy when two capacitors are joined: a new law? *Physics Education*, Vol. 32 (2), Institute of Physics, London, 1997, pp. 88–91.