



This is a repository copy of *An adaptive polynomial based forward prediction algorithm for multi-actuator real-time dynamic substructuring.*

White Rose Research Online URL for this paper:  
<http://eprints.whiterose.ac.uk/79688/>

Version: Accepted Version

---

**Article:**

Wallace, M.I., Wagg, D.J. and Neild, S.A. (2005) An adaptive polynomial based forward prediction algorithm for multi-actuator real-time dynamic substructuring. Proceedings of the Royal Society of London A., 461 (2064). 3807 - 3826. ISSN 1364-5021

<https://doi.org/10.1098/rspa.2005.1532>

---

**Reuse**

Unless indicated otherwise, fulltext items are protected by copyright with all rights reserved. The copyright exception in section 29 of the Copyright, Designs and Patents Act 1988 allows the making of a single copy solely for the purpose of non-commercial research or private study within the limits of fair dealing. The publisher or other rights-holder may allow further reproduction and re-use of this version - refer to the White Rose Research Online record for this item. Where records identify the publisher as the copyright holder, users can verify any specific terms of use on the publisher's website.

**Takedown**

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing [eprints@whiterose.ac.uk](mailto:eprints@whiterose.ac.uk) including the URL of the record and the reason for the withdrawal request.



[eprints@whiterose.ac.uk](mailto:eprints@whiterose.ac.uk)  
<https://eprints.whiterose.ac.uk/>

# An adaptive polynomial based forward prediction algorithm for multi-actuator real-time dynamic substructuring

BY M. I. WALLACE, D. J. WAGG AND S. A. NEILD

*Department of Mechanical Engineering, University of Bristol,  
Queens Building, University Walk, Bristol BS8 1TR, UK*

Real-time dynamic substructuring is a novel experimental technique used to test the dynamic behaviour of complex structures. The technique involves creating a hybrid model of the entire structure by combining an experimental test piece — the substructure — with a set of numerical models. In this paper we describe a multi-actuator substructured system of a coupled three mass-spring-damper system and use this to demonstrate the nature of delay errors which can first lead to a loss of accuracy and then to instability of the substructuring algorithm.

Synchronisation theory and delay compensation are used to show how the delay errors, present in the transfer systems, can be minimised by online forward prediction. This new algorithm uses a more generic approach than the single step algorithms applied to substructuring thus far, giving considerable advantages in terms of flexibility and accuracy. The basic algorithm is then extended by closing the control loop resulting in an error driven adaptive feedback controller which can operate with no prior knowledge of the plant dynamics. The adaptive algorithm is then used to perform a real substructuring test using experimentally measured forces to deliver a stable substructuring algorithm.

**Keywords:** Substructuring; numerical-experimental testing; delay compensation; synchronisation theory; real-time

## 1. Introduction

In this paper we consider the hybrid experimental-numerical testing technique known as real-time dynamic substructuring. The technique involves creating a hybrid model of the whole structure — the emulated system — by combining an experimental test piece — the substructure — with one or more numerical models. The substructuring technique was originally intended to be applied to situations where accurate numerical models of experimental parts were unreliable. As only part of the structure is experimentally tested it allows engineers to view the behaviour of critical elements under dynamic loading at full scale. So far the technique has been developed successfully using expanded time scales, known as pseudodynamic (PsD) testing (Mahin & Shing 1985; Nakashima *et al.* 1992; Donea *et al.* 1996; Pinto *et al.* 2004). As PsD testing is carried out quasi-statically, any time-dependent behaviour of the test specimen is lost. However, this type of testing is usually applied to rate-independent structures and as a result has achieved a large amount of success in the field of earthquake engineering because the strain-rate

sensitivity of the materials can often be neglected. Implementing the substructuring process in real-time means that the damping and inertial components of the substructure dynamics are retained (Blakeborough *et al.* 2001; Darby *et al.* 2001b; Wagg & Stoten 2001; Sivaselvan *et al.* 2004). The numerical model computation time is restricted due to the real-time constraints and that the actuation devices used must have a high enough dynamic rating to achieve the desired accelerations. As this currently cannot always be achieved for large structures, real-time testing has been used for smaller scale component testing applications, for example Horiuchi *et al.* (1999). A comparative overview of real-time and pseudodynamic substructuring is given by Williams & Blakeborough (2001).

To carry out a substructuring test the component of interest is isolated and fixed into an experimental test system. To link the substructure to the numerical models, a set of *transfer systems* (which act on the substructure) are controlled to follow the appropriate output from the numerical model, which is typically a displacement. At the same time the forces between the transfer systems and the substructure are fed back into the numerical models to give a form of bi-directional coupling. Transfer systems are typically single actuators (electric or hydraulic) including their proprietary (built in) controller, but can also be in the form of a more complex test facility such as a shaking table. Single actuator substructuring has been developed beyond the ‘proof of concept’ stage where experiments on simple substructures have been carried out. Multi-actuator substructuring presents a significant engineering challenge in terms of real-time implementation (Nakashima & Masaoka 1999; Wallace *et al.* 2004; Neild *et al.* 2005).

For a substructured system to be stable a delay compensation scheme is typically used to negate the effect of the actuator dynamics resulting from the proprietary control of the transfer system. Typically, this has been achieved by including additional control algorithms (an *outer-loop controller*). Single step forward prediction approaches to substructuring have already been presented by Horiuchi *et al.* (1999) and Darby *et al.* (2001a), and shown to improve accuracy. The algorithm presented here uses a more generic approach than these single step algorithms giving considerable advantages in terms of flexibility and accuracy. It should be noted that there are other approaches which have been proposed to compensate for the transfer system error. For example, lag compensation by inverting an experimental transfer function estimation of the combined inner-loop controller and actuator dynamics, Gawthrop *et al.* (2005), or via the use of model reference adaptive controllers as an outer-loop control strategy, Neild *et al.* (2002).

In Section 2 of this paper we describe a multi-actuator substructuring model of a coupled three mass-spring-damper system (Wallace *et al.* 2004). We describe how techniques from synchronisation theory can be used to give an online accuracy measure based on the delays between the signals passing through the transfer systems (Ashwin 1998). Additionally, we describe how a measure of the accuracy of the substructuring algorithm can be obtained without having to simulate the complete system.

In Section 3 we present a forward prediction algorithm to compensate for the delay present in the transfer system dynamics, based on performing an online polynomial estimation of the numerical model displacement. We then extend this algorithm by closing the control loop to achieve an error driven adaptive feedback controller capable of achieving high levels of synchronisation over transient and

frequency dependent plant behaviour. This approach uses the transfer system synchronisation error to achieve complete delay compensation, which builds on the fundamental concepts presented by Darby *et al.* (2002). We present a formulation of the adaptive parameters and show that this algorithm can operate from a desired initial condition and with no prior knowledge of the plant dynamics. We then present the results from a substructuring test where the experimentally measured forces from the substructure are used and discuss the implications of local and global accuracy for this testing technique.

## 2. Real-time dynamic substructuring with multiple transfer systems

The entire structure — the emulated system — is represented by a hybrid numerical-experimental model where the dynamics of the numerical model are combined with the dynamics of the experimental component — the substructure. The general principle of substructuring remains the same regardless of the number of transfer systems present in the system, however the problem from a control point of view becomes more complicated for more than a single transfer system due to the introduction of cross-coupling between the control signals. In this paper, we will consider the example of a three mass oscillator system with two diametrically opposing excitation walls as shown in Figure 1. This will allow us to demonstrate the problems of achieving accurate control for multi-actuator substructuring using a conceptually simple example. This example has been discussed in detail by Wallace *et al.* (2004), but for completeness will be briefly described here also. The masses are coupled by four identical linear springs,  $k_i$ , and damped by coupled viscous dampers,  $c_i$ , where  $i = 1, 2, 31, 32$ . The system is excited via two moving supports,  $r_j$ , where  $j = 1, 2$ . The general equation of motion for such a system can be written as,

$$M\ddot{\xi} + D\dot{\xi} + K\xi = Sr(t), \quad (2.1)$$

where,  $M$ ,  $D$  and  $K$  are the mass, damping and stiffness matrices respectively and  $Sr(t)$  is the support excitation.  $\xi$  is a vector which represents the states of the system, such that

$$\xi = [z_1^*, z_2^*, z_3^*]^T, \quad (2.2)$$

where,  $(.)^*$  is used to indicate that these dynamics are based on the ‘perfect’ dynamics of the emulated system. In order to create a substructured model of the system shown in Figure 1, the middle mass,  $m_3$ , and accompanying springs,  $k_{31}$  and  $k_{32}$ , are taken to be the substructure. This leaves both excitation walls and

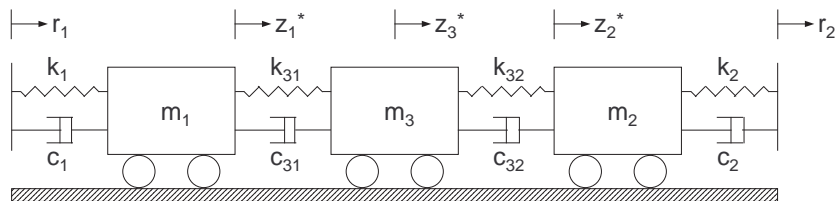


Figure 1. Schematic representation of the three mass system

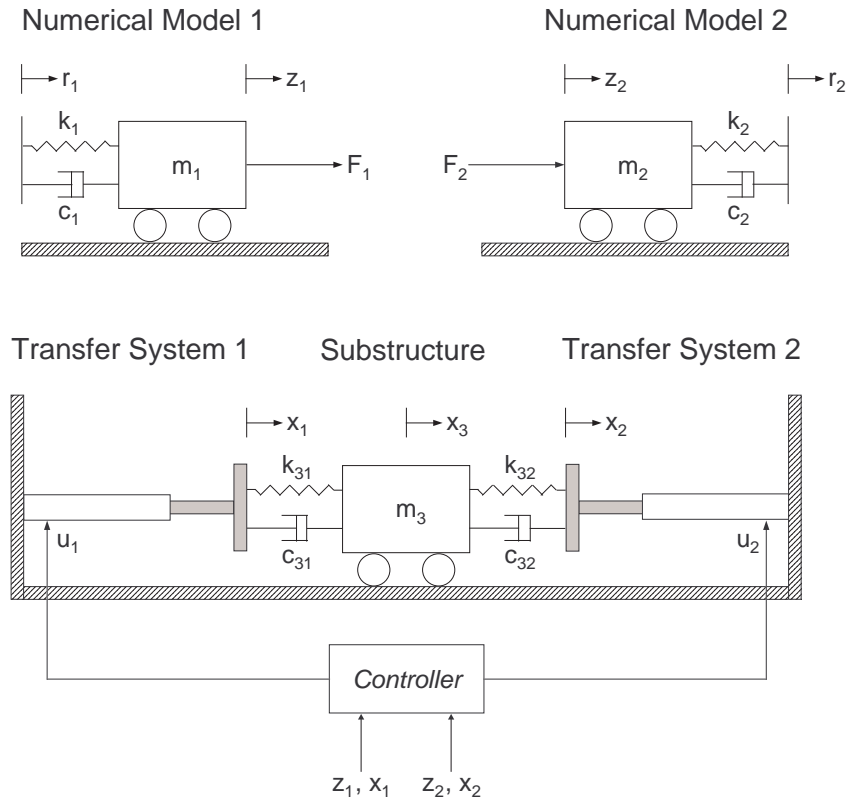


Figure 2. Schematic representation of a *substructured* three mass system with two transfer systems

adjacent masses to be used to create two independent numerical models whose influence is imposed on the substructure by two separate transfer systems (actuators) via two independent control signals,  $u_1$  and  $u_2$ . The influence of the substructure is represented by two autonomous forces,  $F_1$  and  $F_2$ , which are measured experimentally and then imposed back on the numerical models. This new substructured model is shown schematically in Figure 2.

From Figure 2, the dynamics of the two numerical models can be written as,

$$\begin{aligned} m_1 \ddot{z}_1 + c_1(\dot{z}_1 - \dot{r}_1) + k_1(z_1 - r_1) &= F_1, \\ m_2 \ddot{z}_2 + c_2(\dot{r}_2 - \dot{z}_2) + k_2(r_2 - z_2) &= F_2. \end{aligned} \quad (2.3)$$

Due to the linear nature of the example considered here, we know explicitly the forces at each time interval allowing us to measure the accuracy of an individual test, where

$$\begin{aligned} F_1 &= c_{31}(\dot{z}_3 - \dot{z}_1) + k_{31}(z_3 - z_1), \\ F_2 &= c_{32}(\dot{z}_2 - \dot{z}_3) + k_{32}(z_2 - z_3). \end{aligned} \quad (2.4)$$

However, substructuring is specifically aimed at modelling systems where this is not the case (Wagg & Stoten 2001), and in such a case these forces could only

be measured using load cells between the transfer systems and the experimental substructure. This point highlights the difficulty in assessing the accuracy of substructuring tests for complex systems, as the end results of the test can never be compared with results from the emulated system. This point will be discussed in greater depth later in this section and in Section 2(f).

The aim of the control algorithm is to achieve synchronisation between the *desired* interface displacement of the numerical models,  $z$ , and the *actual* position of the transfer systems,  $x$ . However, under just the linear control of the propriety controller, a transfer system will always be subject to some form of delay,  $\tau$ , which can either be characterized as a pure delay or as a frequency dependent delay (lag) depending on the type of actuator. This error will lead to a reduction in the degree of synchronisation of the transfer system and thus a corresponding reduction in the accuracy of the numerical model compared to that of the emulated system (Mosqueda 2003). In fact the nature of this delay error in the substructuring model can be represented by two coupled components which we can write as,

$$e = e_1(z^*, z, t) + e_2(z, x, t), \quad (2.5)$$

where,  $e_1$  is a function which describes the accuracy of the numerical models compared to the appropriate variable in the complete emulated system,

$$e_1 = [(z_1^* - z_1) \quad (z_2^* - z_2)]^T, \quad (2.6)$$

and  $e_2$  represents the degree of synchronisation between each transfer system and its numerical model (the local measure of accuracy for the control algorithm),

$$e_2 = [(z_1 - x_1) \quad (z_2 - x_2)]^T. \quad (2.7)$$

Therefore, by combining  $e_1$  and  $e_2$  we can get a global measure of the accuracy of the substructuring test which relates the emulated system coordinates  $z^*$  to the actual displacement of the transfer systems  $x$ . However, when substructuring complex systems it is not possible to compute  $z^*$ , and the only measure of accuracy is the degree of synchronisation  $e_2$ . We note also that the numerical model coordinates  $z$  are, in effect, a function of the transfer system  $x$ , as the force vector  $F$  will be subjected to the same delay  $\tau$  as the transfer system, such that

$$z = f(r(t), F(t - \tau)), \quad (2.8)$$

This highlights the nature of the coupling between  $e_1$  and  $e_2$ . In the ideal case, achieving perfect synchronisation by removing the delay  $\tau$  from the transfer systems will result in  $e_2 \rightarrow 0$ . This in turn will mean that the correct force vector  $F(t)$  will be added into the numerical models at the correct time, such that  $e_1 \rightarrow 0$  and the substructured system will replicate the dynamics of the emulated system. This argument allows us to propose the following:

**Proposition 2.1.** *If the synchronisation error,  $e_2 = 0$ , for all time  $t \geq 0$  during a substructuring test then  $x = z$  and  $e_1 = 0$  such that the substructured model exactly replicates the dynamics of the emulated system.*

However, in practise the synchronisation error,  $e_2$ , can never be exactly equal to zero in a real substructuring test so the practical interpretation of Proposition 2.1 is

that as  $e_2 \rightarrow 0$  the substructured model more closely replicates the dynamics of the emulated system. The significance of Proposition 2.1 is that it gives an indication of the accuracy of a substructured system using the only measurable quantity of error  $e_2$ , the local control error.

From an accuracy standpoint alone it is clear that the primary control objective should be to minimize  $e_2$ , however the size of the delay  $\tau$  also has a significant effect on the stability of the substructuring algorithm as a whole. Any error in  $e_2$  will result in a corresponding error in  $e_1$  and thus propagate to the next time step leading to potential instability of the substructuring algorithm. This is discussed further in Section 2(b).

(a) *Experimental setup*

To implement substructuring we are using a combination of standard Simulink blocks and custom S-Functions (C modules compiled into \*.dll files using a MEX compiler) to formulate a model and a dSpace DS1104 R&D Controller Board to implement it in real-time. The companion software ControlDesk is used for online analysis and control, providing soft real-time access to the hard real-time application. More detail on the experimental setup is given by Wallace *et al.* (2004).

Figure 3 (a) shows the substructured model setup along with the transfer systems which imposes the interface displacements on the physical substructure. An expanded view is shown in Figure 3(b). Each mass is a constant 2.2kg and connected via three parallel shafts constraining their motion to one degree of freedom. Through system identification the springs were found to have a stiffness of  $k = 9000\text{N/m}$  and damping value of  $c = 30\text{Ns/m}$ . The stiffness of the strings allows significant coupling between the two transfer system, this is discussed further in Section 3(f).

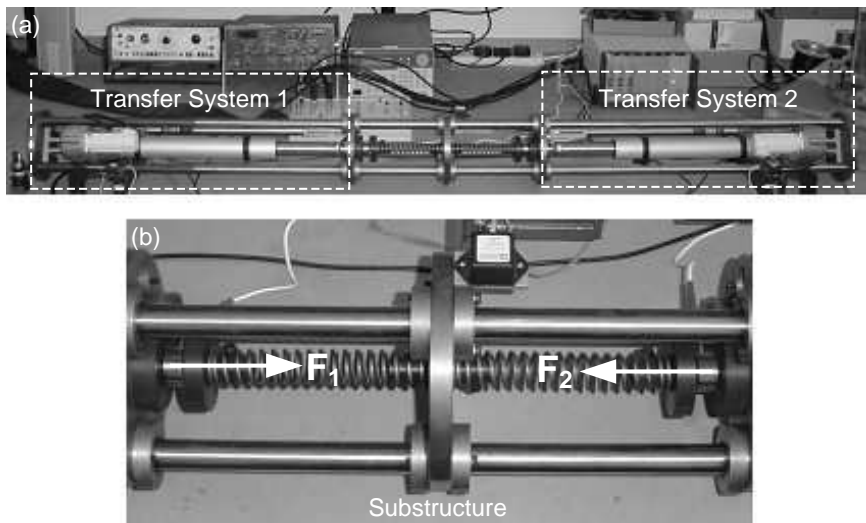


Figure 3. (a) Experimental rig setup of substructured model. (b) Enlarged view of substructure.

(b) *Effect of feedback delay on the substructuring algorithm*

There is an important difference between the difficulties faced in a standard control problem to that faced when performing substructuring. For substructuring, the reference signal (i.e. the control demand) for each transfer system is not known at the start of each time step as in a normal control problem, but must be created in its respective numerical model during each time period. A small delay in the transfer system response (such that  $e_2 \neq 0$ , from Equation 2.7) introduces a corresponding error in the feedback force vector, which can be thought of as adding negative damping to the system (Horiuchi *et al.* 1999). This discrepancy has the effect of reducing the accuracy of the numerical models (compared to the emulated system) until the magnitude of the synchronisation delay increases to such a degree that a sign change for the damping of the overall system occurs. At this point instability of the substructuring algorithm is observed and is characterized by the onset of oscillations with exponential growth (Wallace *et al.* 2004).

(c) *Measuring synchronisation accuracy using subspace plots*

Synchronisation *subspace plots* are used to show the effectiveness of the control algorithm by plotting the *desired* versus *actual* responses, (Ashwin 1998). A subspace plot shows the amplitude accuracy and the magnitude of delay coupled together at any one time interval. Perfect synchronisation is represented by a diagonal straight line with maxima and minima of the reference signal. Any reduction in synchronisation can be seen as a deviation from this idealized line. For periodic wall excitation conditions these plots build up into a repeating periodic pattern, which can appear complex. However, the individual components of amplitude and delay produce their own specific and identifiable patterns if evaluated separately.

The result of varying the amplitude accuracy is to change the angular orientation of the subspace plot compared to the idealized line. Figure 4 shows the result of increasing amplitude accuracy from a 0% to 25%, 50% and then 75% of perfect synchronisation. Continuing this trend, the angular orientation further increases to

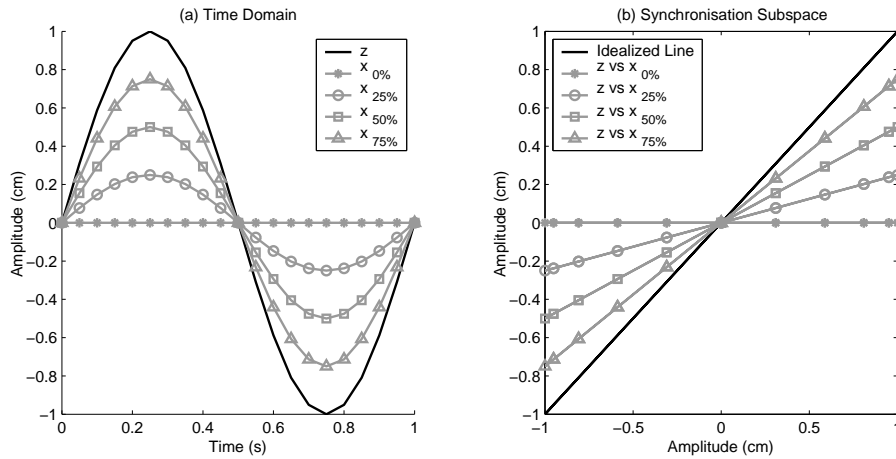


Figure 4. Effect of increasing amplitude accuracy with zero delay.



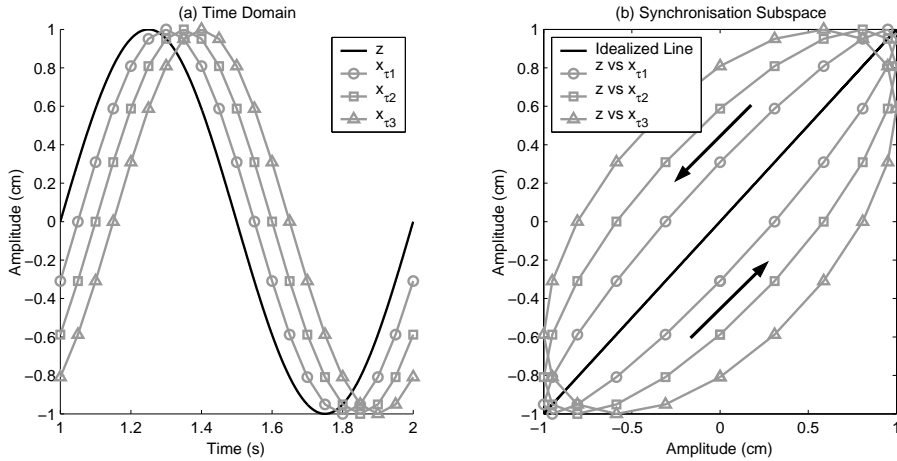


Figure 5. Effect of increasing delay with perfect amplitude accuracy.

a limit of a  $90^\circ$  (vertical) line which is the outcome of an infinite plant response. The consequence of introducing a constant delay between the reference signal and the plant response is to transform the idealized straight line into an ellipse (anti-clockwise implies negative damping, clockwise implies positive damping) as shown by Figure 5(b). The greater the delay, the larger the width of the minor axis of the ellipse, with the change being proportional to the delay magnitude. If the delay is not constant through one period, then the ellipse no longer has a uniform shape.

In a typical subspace plot however, the effect of amplitude and delay are coupled together with both being able to vary independently through a single period. However, despite this, their visual interpretation remains simple instant online guide to the accuracy of an individual test, thus allowing linear controllers to be tuned online and adaption characteristics to be viewed. This technique will allow us to measure the effectiveness of the forward prediction algorithms presented in Section 3.

### 3. Forward prediction

#### (a) Application to forward prediction

The value of being able to forward predict online can be seen from Figure 6. Controlling to an arbitrary reference signal  $z$  we see that there is an inevitable delay  $\tau$  in the dynamic response of the plant  $x$  (expected in any dynamical system under linear control). By predicting forward the same amount as the delay  $\tau$  to create a new reference signal  $z'$ , at any point in time  $t_n$ , and then using this new reference as the transfer system demand, we will be able to eliminate the response delay thus obtaining nominally zero synchronisation error, such that  $e_2 \rightarrow 0$  from Equation 2.7.

Figure 7 shows an example sinusoid reference signal of 10Hz (shown by the solid grey line) which is to be predicted forward. A section of data must be taken to act as control points for the fitted polynomial curve, here a buffer of 20 data points  $n = 20$  starting at time  $t = 1$  has been stored. For example, if we wish to predict forward 18 time steps ( $P = 18$  and  $\Delta t$  is the sample time step size and equal to 1ms

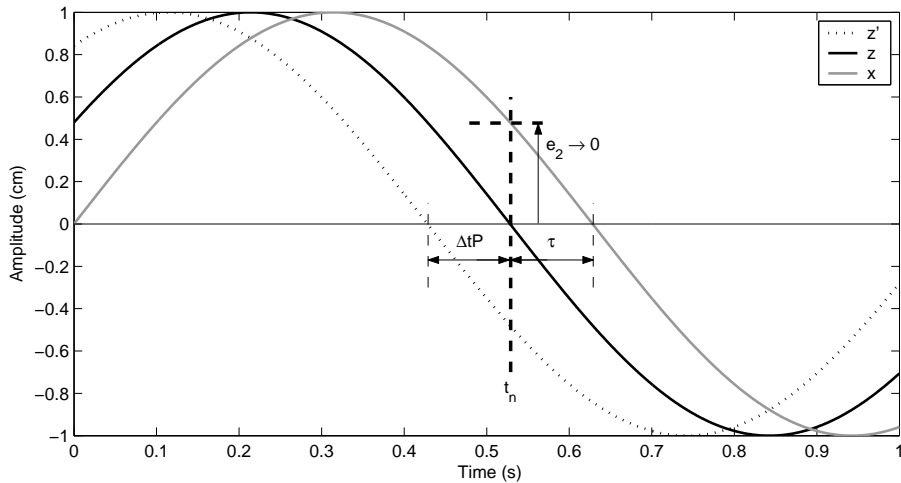


Figure 6. Delay compensation for transfer system dynamics by creating a forward predicted signal  $z'$ .

in this case) Figure 7 shows the differing accuracies obtained by the various order  $N$  of polynomial fitted curves. From this example, we see that both the 8th and 10th order curves have the highest degree of accuracy, whereas the 4th order curve loses coherence much more quickly. However, the higher the order of prediction the more computationally intensive the calculation and the more inherently unstable the predicted signal  $z'$  will become. Therefore, the desired amount of time for which the numerical model should be predicted forward is equal to  $\tau = \Delta tP$  to achieve optimal delay compensation.

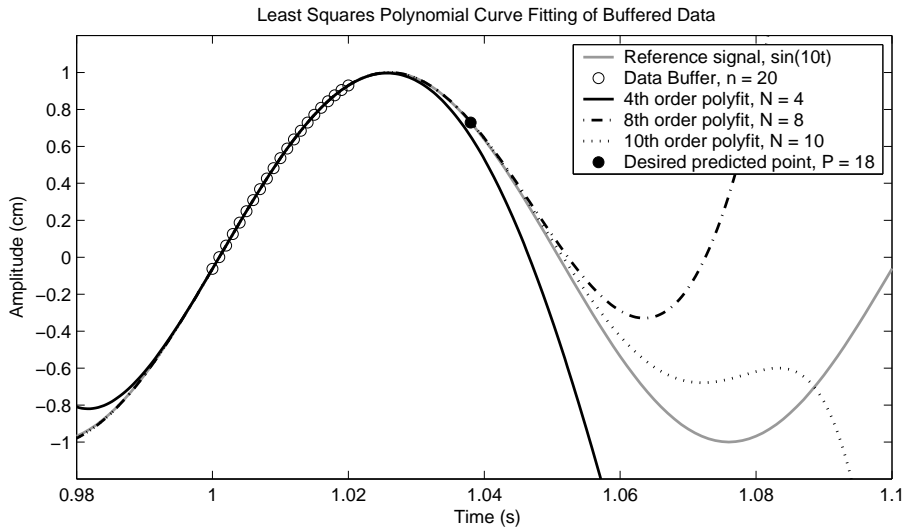


Figure 7. Least squares  $N^{th}$  order polynomial curve fitting of buffered data.

(b) *Least squares polynomial fitting*

One mathematical procedure for finding the best fitting curve to a given set of points is by minimising the sum of the squares of the offsets of the points from the curve (Kreyszig 1999). The linear least squares fitting technique is the simplest and most commonly applied form of linear regression and provides a solution to the problem of finding the best fitting straight line through a set of points. However, due to the dynamics of the numerical models in our system we move from a best-fit line to a best-fit polynomial, where sums of vertical distances are used. A polynomial in  $x$  of order  $N$  with coefficients  $a_i$  (where  $i = 0, \dots, N$ ) is given by,

$$y = a_0 + a_1x + \dots + a_Nx^N. \quad (3.1)$$

Using a standard least squares polynomial derivation (Kreyszig 1999), given  $n$  number of data points  $(x_0, y_0), \dots, (x_{n-1}, y_{n-1})$  with polynomial coefficients  $a_0, \dots, a_N$  the equation of the curve is given by,

$$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^N \\ 1 & x_1 & x_1^2 & \dots & x_1^N \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & x_{n-1} & x_{n-1}^2 & \dots & x_{n-1}^N \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_N \end{bmatrix}. \quad (3.2)$$

Therefore, in matrix notation, the equation for a polynomial fit is given by

$$y = Xa, \quad (3.3)$$

which can be solved by premultiplying by the matrix transpose, such that  $X^T y = X^T X a$ . The matrix can then be inverted directly to give the solution,

$$a = (X^T X)^{-1} X^T y. \quad (3.4)$$

(c) *Single time step forward prediction*

Delay compensation by polynomial extrapolation is not a new concept, single time step prediction techniques have already been proposed in relation to substructuring (Horiuchi *et al.* 1999; Darby *et al.* 2001a), and shown to improve accuracy. These algorithms are based on using predefined coefficients,  $\bar{a}_i$ , for an  $N^{\text{th}}$  order polynomial fit of  $n$  number of control points following the equation,

$$z' = \sum_{i=0}^N \bar{a}_i z_i, \quad (3.5)$$

where  $z_0$  is the present calculated numerical model displacement and  $z_i$  are the previously calculated displacements at  $\Delta t \times i$  units of time ago. Figure 8 shows the forward predicted point  $z'$  being obtained by extrapolating the polynomial function over the present displacement  $z_0$  and  $N$  previous calculated values, thus making the number of control points used,  $n = (N + 1)$ .

For this  $2^{\text{nd}}$  order polynomial fit we attain the following constants  $\bar{a}_0 = 3$ ,  $\bar{a}_1 = -3$  and  $\bar{a}_2 = 1$  (Horiuchi *et al.* 1999). Note that we can only predict one whole time step forward and the correct number of control points  $n$  must used for

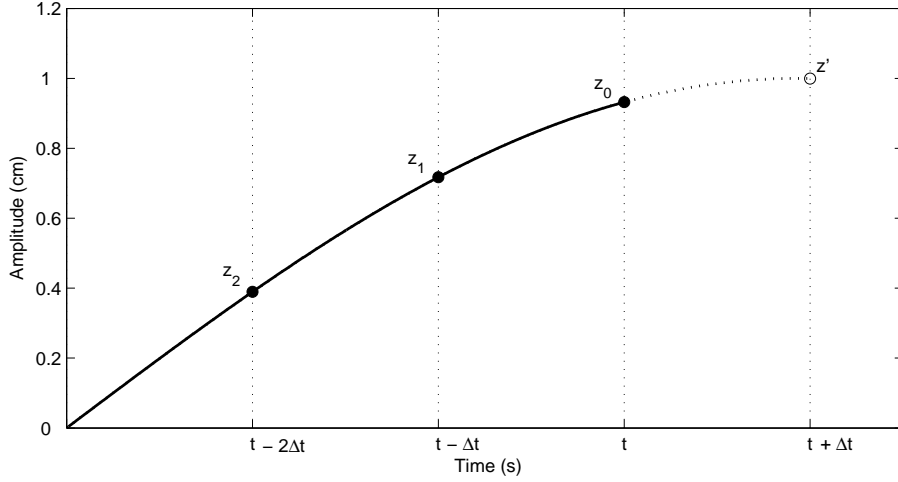


Figure 8. Single time step prediction of  $n = 3$  control points for a  $N = 2$  order polynomial fit.

the polynomial function (to ensure that the  $X$  matrix of Equation 3.3 is square) otherwise the polynomial fit will not hold. To predict further ahead than one time step we can simply apply Equation 3.5 more than once (Darby *et al.* 2001a). For example to predict two time steps forward we see that if

$$z'(\Delta t) = 3z_0 - 3z_1 + z_2, \quad (3.6)$$

then,

$$\begin{aligned} z'(2\Delta t) &= 3z'(\Delta t) - 3z_0 + z_1, \\ &= 6z_0 - 8z_1 + 3z_2. \end{aligned} \quad (3.7)$$

Note that we are still restricted to predicting in whole time steps unless an additional interpolation is carried out between the two points (Darby *et al.* 2001a).

However, if we use Equation 3.4 to solve the  $(X^T X)^{-1} X^T$  matrix components numerically first, we can deduce a more generalised forward prediction algorithm where multiples and fractions of one time step can be predicted in one iteration and we are no longer constrained to  $n = (N + 1)$  number of control points. Increasing  $n$  can help to smooth noise out of the numerical model and thus from the control signal. We define the size of the  $X$  matrix by choosing values for the number of control points  $n$  and the order of the polynomial fit  $N$ . For the case above,  $n = 3$  and  $N = 2$ , thus rewriting the general Equation 3.3 for this specific case gives us,

$$\begin{bmatrix} z_0 \\ z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 1 & t_0 & t_0^2 \\ 1 & t_1 & t_1^2 \\ 1 & t_2 & t_2^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}, \quad (3.8)$$

where,  $t_i$  is the current simulation time for each value of  $z_i$ . Therefore, for a sample time step size  $\Delta t$  as shown in Figure 8,

$$\begin{bmatrix} z_0 \\ z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -\Delta t & \Delta t^2 \\ 1 & -2\Delta t & 4\Delta t^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}. \quad (3.9)$$

The predicted point  $z'$  is given by an adaptation of Equation 3.3,

$$z' = X_P a, \quad (3.10)$$

where,  $X_P$  is the forward prediction vector and given by,

$$X_P = [1 \quad P\Delta t \quad \dots \quad P^N \Delta t^N], \quad (3.11)$$

and  $P$  is the number of time steps to be predicted forward (which does not have to be an integer multiple of  $\Delta t$ ). As  $X$  is a square matrix in this case,  $(X^T X)^{-1} X^T = X^{-1}$ , therefore,

$$z' = X_P X^{-1} z. \quad (3.12)$$

Evaluating  $X_P X^{-1}$  gives an expression which is independent of the sample time step size  $\Delta t$ . Substituting into Equation 3.12 for the case when  $P = 1$  we see that

$$z'(\Delta t) = 3z_0 - 3z_1 + z_2, \quad (3.13)$$

thus matching the coefficients of the one step method in Equation 3.6. Substituting  $P = 2$  we see that

$$z'(2\Delta t) = 6z_0 - 8z_1 + 3z_2, \quad (3.14)$$

matching the coefficients found in Equation 3.7 but in a single operation. Therefore, the coefficients  $\bar{a}_i$  from Equation 3.5 are actually the pre-multiplication of the forward prediction vector  $X_P$  for the special case of  $P = 1$ , such that

$$\bar{a} = X_P [(X^T X)^{-1} X^T]. \quad (3.15)$$

In this way, the coefficients  $\bar{a}$  are predefined and thus fix the level of forward prediction obtained, whereas the coefficients of the polynomial function  $a$  are calculated each time step and therefore allow variable degrees of forward prediction to be achieved by altering the value of  $P$  online.

(d) *Online forward prediction using variable polynomial coefficients*

To achieve online forward prediction a buffer of  $n$  data points of the numerical model displacement  $z$  are stored and updated by a buffer overlap of  $(n - 1)$  each time step. These are then fed into a least squares polynomial fitting sub routine which uses the set of control points  $[z_{n-1}, z_{n-2}, \dots, z_0]$  to calculate the  $N^{th}$  order polynomial fit and therefore find the coefficient vector  $a$  from Equation 3.4 for that time step. The current coefficients are then fed into a reconstruction algorithm that calculates the predicted point  $z'$  according to Equation 3.10 using the forward prediction vector  $X_P$ , such that

$$z' = k_a \sum_{i=0}^N (a_i P^N). \quad (3.16)$$

When using linear control, an additional source of error is the amplitude accuracy. Typically, as the excitation frequency is increased the level of amplitude accuracy of the transfer system reduces. Thus, the forward predicted point  $z'$  is then multiplied by a gain  $k_a$  to remove this error and increase the synchronisation accuracy.

## (e) Adaptive forward prediction (AFP)

The basic forward prediction algorithm can be used to effectively remove the transfer system delay, however, both the magnitude of the forward prediction,  $P$ , and the amplitude gain,  $k_a$ , must be specifically tuned for each different excitation condition, thus making the algorithm, in effect, a feed-forward controller. To remove the need for this tuning and to allow the algorithm to achieve high levels of synchronisation for frequency dependent and transient plant conditions we must close the control loop and use the feedback dynamics of the transfer system. In combination with the existing linear control present in the substructuring algorithm this model structure now represents an error driven adaptive feedback controller (Åström & Wittenmark 1995). We cannot explicitly measure the transfer system delay  $\tau$  as we only have data for the current time step, thus we only know the synchronisation error  $e_2$  at any single point in time. Therefore, to achieve complete delay compensation we can indirectly force  $\tau \rightarrow 0$  by explicitly using a measure of the synchronisation error  $e_2$ . An alternative technique which uses this feedback error to achieve adaptive compensation is presented by Darby *et al.* (2002).

As before, the current coefficients are fed into a reconstruction algorithm that calculates the predicted point  $z'$  but now the magnitude of the forward prediction  $\tau$  is now governed by,

$$\tau = \Delta t(P + \rho), \quad (3.17)$$

where,  $\Delta t$  is the sample time step size,  $P$  is the *fixed* initial number of time steps to be forward predicted and  $\rho$  is the *adaptive* number of time steps to be forward predicted. Likewise, the amplitude accuracy is now governed by,

$$z' = z'(k_a + \sigma), \quad (3.18)$$

where,  $k_a$  is the *fixed* amplitude gain and  $\sigma$  is an *adaptive* amplitude variable which together control the amount of the predicted reference signal to be used.

Setting  $P = 0$  and  $k_a = 1$  will bring about zero initial conditions. The delay compensation can then be completely achieved by the adaptive parameter  $\rho$  and the amplitude error completely removed by  $\sigma$ . Thus, we can use this new adaptive algorithm when we have no knowledge of the plant dynamics and when there is transient or frequency dependent plant behaviour. However, it is not always desirable to start from zero initial conditions, in fact in many cases (and mainly in earthquake engineering) it is important to start the test with the AFP algorithm in a state near to optimal adaptation. These optimal values can be estimated by observing the steady state adaptive values of an uncoupled system identification test for the specific transfer system. This would then allow the initial transient phase of the test to be avoided.

The adaption algorithm works off four triggered states,  $\varphi_{1,\dots,4}$ , which have a null value until their individual trigger conditions are met. Trigger states,  $\varphi_1$  and  $\varphi_2$ , are activated on the condition of sign change of the numerical model displacement, when  $z$  has zero amplitude. The first state represents a rising edge,  $z$  changing from negative to positive, and the second a falling edge,  $z$  changing from positive to negative. The other two trigger states are similar except activated on the condition of sign change of the numerical model velocity,  $\dot{z}$ , with  $\varphi_3$  representing a rising edge and  $\varphi_4$  representing a falling edge. Effectively these two conditions give the time at which the maxima  $\varphi_3$  and minima  $\varphi_4$  of the numerical model displacement occurs.

The adaptive forward prediction parameter  $\rho$  is calculated when either of the first two triggered states  $\varphi_{1,2}$  are met. This allows the delay of the transfer system response to be observed independently from any amplitude error. The value of  $\rho$  is given by,

$$\rho_{n+1} = \rho_n \pm \alpha e_{2,n}^\gamma, \quad (3.19)$$

where,  $\alpha$  is an adaptive gain parameter for the magnitude of forward prediction and  $\gamma$  sets the convergence curve and must be greater than or equal to 1. Note that the  $\pm$  relates to whether the signal is a rising or falling edge. Equation 3.19 shows that when the synchronisation error  $e_2$  is zero,  $\rho_{n+1} = \rho_n$  thus  $\rho$  retains its previous value, indicating that full delay compensation  $\tau$  has been achieved.

Similarly, the adaptive amplitude parameter is calculated when either of the second two triggered states  $\varphi_{3,4}$  are met. This allows the peak of the numerical model to be compared to that of the transfer system once delay compensation has occurred. The value of  $\sigma$  is given by,

$$\sigma_{n+1} = \sigma_n \pm \beta e_{2,n}^\gamma, \quad (3.20)$$

where,  $\beta$  is an adaptive gain parameter for the amplitude accuracy. Note that until full delay compensation has been achieved the parameter  $\sigma$  will give an under estimation of the amplitude error. However, as the delay is by far the dominant factor in the compensation algorithm this condition does not effect the performance of the controller.

We see that by setting both adaptive parameters  $\alpha$  and  $\beta$  to zero, the adaption algorithm can be turned ‘off’ resulting in the basic feed-forward controller, assuming both  $\rho_0$  and  $\sigma_0$  equal zero. Note that both Equations 3.19 and 3.20 include history data, analogous to that of an integrator, such that any steady state error is forced to zero. The choice of  $\gamma$  decides the convergence curve. The synchronisation error should be such that  $e_2 \ll 1$  so the higher  $\gamma$ , the slower the convergence at very low instances of synchronisation error, thus the smoother the steady state values for the adaptive parameters but the less reactive it is to fast transient or frequency dependent plant behaviour. Additionally, as the adaption only occurs at the set trigger conditions,  $\varphi_{1,\dots,4}$ , the AFP algorithm is subject to a persistence of excitation criterion (Åström & Wittenmark 1995).

#### (f) Substructuring using the AFP algorithm

To show the effectiveness of the AFP algorithm, we first perform a substructuring test using forces generated by a model of the *emulated* system, rather than the actual *measured* forces as this will ensure that the numerical models will exactly match the dynamics of the emulated system, such that  $e_1 = 0$  from Equation 2.6. This is analogous to pseudodynamic substructuring where an estimation of the force is used (Donea *et al.* 1996). A typical challenge faced in real-time substructuring is a rapidly varying transfer system delay (or lag depending on the plant characteristics), therefore, Figure 9 shows the subspace plots for a sine sweep excitation test. The level of synchronisation between each respective numerical model and its transfer system are shown for the case when there is no forward prediction (only the inner loop proprietary linear control is active) in Figures 9 (a1) and (b1), and for the case where we use the AFP algorithm as an outer loop strategy in Figures 9

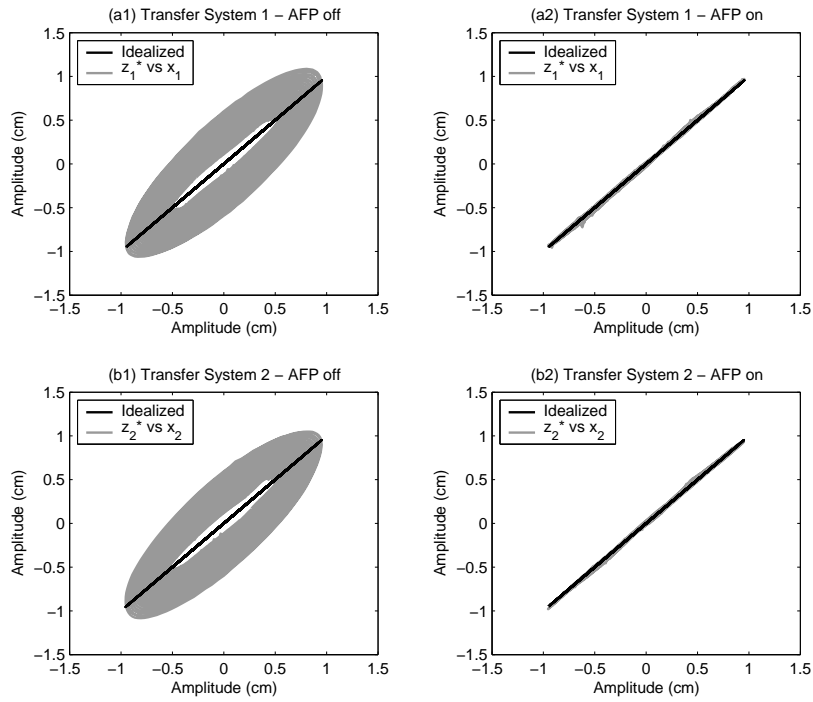


Figure 9. Synchronisation subplots for equal and opposite wall sweep excitation of  $r_{1,2} = 3$  to 10Hz in 5s and then back to 3Hz in 5s. Controller parameters  $N = 4$ ,  $n = 10$ ,  $\alpha_{1,2} = 100$ ,  $\beta_{1,2} = 5$ ,  $\gamma_{1,2} = 2$ ; Adaptive parameters shown in Figure 10.

(a2) and (b2). It is clear that the use of the AFP algorithm results in a significant improvement in the level of synchronisation. The constantly changing conditions means that the delay compensation scheme never reaches steady state values but instead must constantly monitor the synchronisation error in order to maintain a high

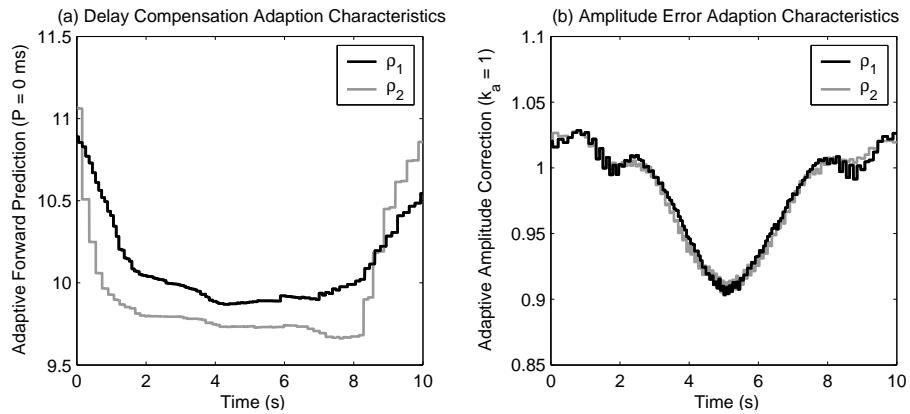


Figure 10. Adaptive parameter characteristics for Figure 9 (a2) and (b2).



level of compensation. Although the excitation demands are equal and opposite and the actuators are of the same type, we can see that the transfer system dynamics vary as shown by the differing adaption characteristics in Figure 10 (a). This is a marked difference to the similarity of the amplitude error shown in Figure 10 (b). This highlights the need for the control of the transfer systems to be decoupled such that their specific mechanical characteristics can be dealt with independently.

However, to achieve a true real-time dynamic substructuring test the *actual* force must be fed back into its respective numerical model rather than the one generated by a model of the *emulated* system. Figure 11 shows the steady state results for both transfer systems for such a test using the AFP algorithm. Firstly, it can be seen that the substructuring algorithm has remained stable due to the high degree of synchronisation between the numerical models and their respective transfer systems, shown by Figure 11 (a2) for Transfer System 1 and (b2) for Transfer System 2. In both cases, virtually all the delay has been removed. The resulting numerical models can then be compared to their respective emulated dynamics — shown in Figures 11 (a1) and (b2). The reduction in synchronisation of the numerical models highlights the fact that although the *local* control error is small a sizeable substructuring error exists. The reason for the extent of the substructuring error in this case is the magnitude of cross coupling between the transfer systems. Figure 12 shows the manufacture’s specification for the actuator capacity envelope for the transfer systems used in these experiments. The experimental data shown is for Transfer

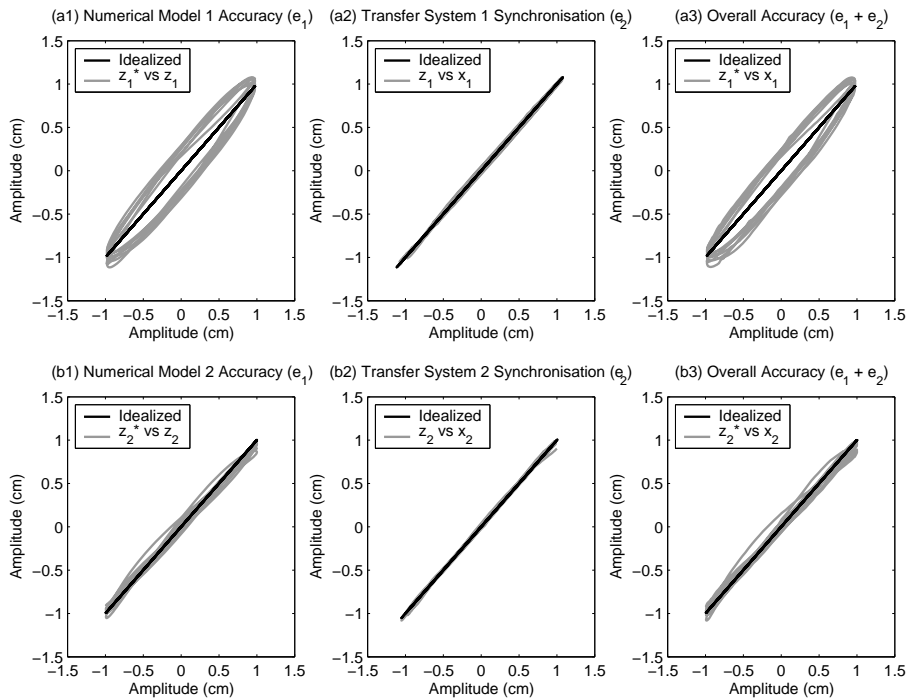


Figure 11. Comparative synchronisation subplots for a real substructuring test for wall excitation of  $r_{1,2} = 8Hz$  (excitation magnitudes are not equal) using the AFP algorithm;  $\bar{\rho}_1 = 9.67$ ,  $\bar{\rho}_2 = 9.53$ ,  $\bar{\sigma}_1 = 0.96$ ,  $\bar{\sigma}_2 = 0.95$ ,  $\alpha_{1,2} = 75$ ,  $\beta_{1,2} = 5$ ,  $\gamma_{1,2} = 2$ ,  $N = 4$ ,  $n = 10$ .

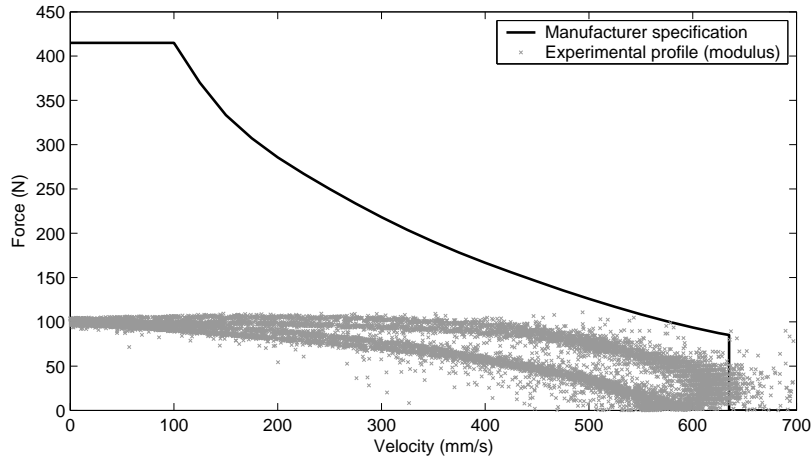


Figure 12. Actuator capacity envelope for Transfer System 1. Experimental data shown for the test of Figure 11(a) (10s test data).

System 1 (although a similar profile would be observed for Transfer System 2) for the test shown in Figure 11(a). It is typical in real-time substructuring that the actuators are operated right up to their maximum performance envelope. It is close to and beyond this envelope that the cross-coupling and other nonlinear effects become significant. However despite this, good local synchronisation is maintained by the AFP algorithm.

Figure 13 shows the case where the AFP algorithm is not used, such that the delay experienced due to a standard linear controller is not removed. The substructuring test is started using the *emulated* force vector to ensure stability and then switched over to the *actual* force vector after approximately 3 seconds of run time. We observe the instability of the substructuring algorithm immediately building in Figure 13 (a). It is important to note that it is not the controller which is becoming unstable as can be seen from Figure 13 (b), which shows a consistent transfer system

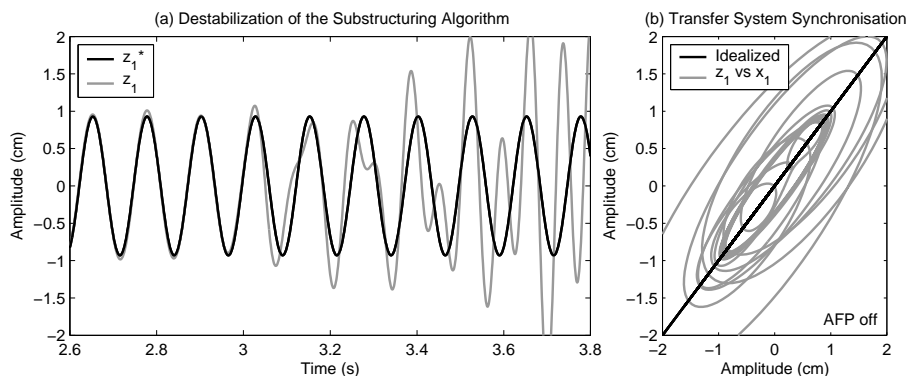


Figure 13. Destabilization of the substructuring algorithm for an identical test to Figure 11 but only using the inner loop linear control with no delay compensation.

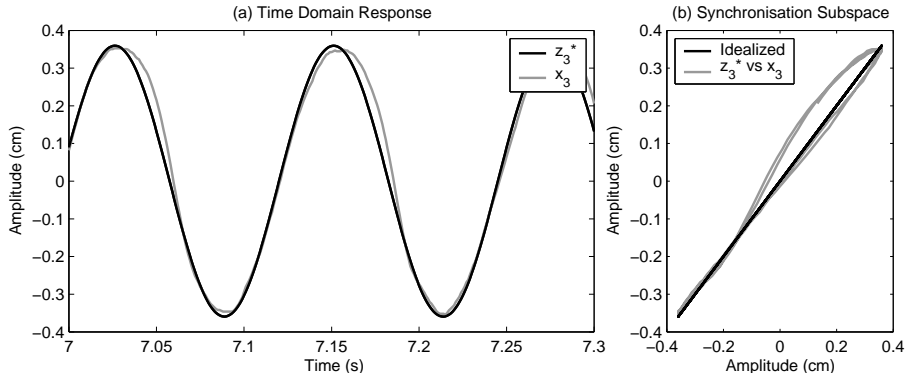


Figure 14. Accuracy of the substructure dynamics for the test of Figure 11.

delay throughout the entirety of the test. This example shows that the removal of the transfer system delay is essential in ensuring a stable substructuring algorithm (Horiuchi *et al.* 1999).

The overall resulting *global* accuracy of the substructuring test using the AFP algorithm is shown in Figures 11 (a3) and (b3), which indicates the level of synchronisation between the emulated system and the resulting transfer system dynamics. As stated in Equation 2.5, the global error is a linear addition of the two sources of error,  $e_1$  and  $e_2$ . However, we note that  $e_1$  often cannot be calculated when performing a complex substructuring test as the emulated dynamics would not be known explicitly. Figure 14 shows the result of the global error in the dynamics of the substructure  $x_3$  compared to that of the emulated system  $z_3^*$  for the example presented in Figure 11. The combination of the local control error and the numerical model error due to the cross coupling of the transfer systems has resulted in a non-linear relationship of the substructure dynamics compared to that of the emulated system. This is an important concept for measuring the accuracy of a substructuring test as the only error that we have a direct influence on is the level of synchronisation achieved,  $e_2$ . The magnitude of the numerical model error,  $e_1$  and thus the resulting global error is dependent on the capacity of the transfer systems used to perform the test. Working well within the actuator’s performance envelope will result in limited coupling between the transfer systems and therefore a correspondingly lower global error. For a complex substructured system this means that the explicit measure of accuracy,  $e_2$  (the local control error), needs to be assessed in combination with the experimental transfer system profile (compared to its actual performance envelope) to gain a more complete measure of the accuracy of a substructuring test.

#### 4. Conclusions

In this paper we have discussed the hybrid numerical-experimental technique of real-time dynamic substructuring. This technique offers a method of testing critical elements of a system under dynamic loading at full scale. The example we have used in this work is a multi-actuator substructuring model of a coupled three mass-spring-damper system. This example has allowed us to demonstrate the nature of delay errors which lead to loss of accuracy in substructure models. We then

described, using synchronisation theory, how these delay errors can be measured. We have described a measure of the accuracy of the substructuring algorithm without having to simulate the complete system. This is a generic result which holds for all substructured systems, although the exact relationship between  $e_1$  and  $e_2$  will be system dependant.

In Section 3(d) we have shown how the resulting delay errors can be removed by using an online forward prediction technique. This is based on a polynomial estimation to compensate for the delay present in the transfer systems. In addition to accuracy, this technique has benefits in achieving a stable substructuring algorithm. In Section 3(c) the algorithm presented here is compared with existing techniques and shown to be a more generic approach than the single step algorithms applied to substructuring thus far. This gives considerable advantages in terms of flexibility and accuracy.

Finally in Section 3(e) we have shown how the basic forward prediction algorithm can be extend to one with adaption by closing the control loop to achieve an error driven adaptive feedback controller. This allows the forward prediction algorithm to accurately cope with frequency dependent plant behaviour and operate with no prior knowledge of the plant characteristics. We show that the AFP algorithm can be used to achieve a stable substructuring algorithm despite the discrepancies in the dynamics of the numerical models compared to those of the emulated system. This is in contrast to the immediate exponential growth observed when the test is carried out using a standard linear controller in isolation.

## Acknowledgements

Two authors would like to acknowledge the support of the EPSRC: Max Wallace is supported via an EPSRC DTA and David Wagg via an EPSRC Advanced Research Fellowship.

## References

- Åström, K.J. & Wittenmark, B. 1995 *Adaptive Control*, Addison Wesley.
- Ashwin, P. 1998 *Non-linear dynamics, loss of synchronisation and symmetry breaking*, Proc. of the Institution of Mech. Engrs, Part G, **212**(3), 183–187.
- Blakeborough, A., Williams, M.S., Darby, A.P. & Williams, D.M. 2001 *The development of real-time substructure testing*, Phil. Trans. R. Soc. Lond., Series A, **359**, 1869–1891.
- Darby, A.P., Blakeborough, A. & Williams, M.S. 2001a *Improved control algorithm for real-time substructure testing*, Earthquake Engng Struct. Dyn., **30**, 431–448.
- Darby, A.P., Blakeborough, A. & Williams, M.S. 2001b *Real-time substructure tests using hydraulic actuator*, J. Engng Mech., Amer. Soc. Civ. Engrs, **125**(10), 1133–1139.
- Darby, A.P., Williams, M.S. & Blakeborough, A. 2002 *Stability and delay compensation for real-time substructure testing*, J. Engng Mech., Amer. Soc. Civ. Engrs, **128**(12), 1276–1284.
- Donea, J., Magonette, P., Negro, P., Pegon, P., Pinto, A. & Verzeletti, G. (1996) *Pseudo-dynamic capabilities of the ELSA laboratory for earthquake testing of large structures*, Earthquake Spectra, **12**(1), 163–180.
- Gawthrop, P.J., Wallace, M.I. & Wagg, D.J. 2005 *Bond-graph based substructuring of dynamical systems*, Earthquake Engng Struct. Dyn., **34**(6), 687–703.

- Horiuchi, T., Inoue, M., Konno, T. & Namita, Y. 1999 *Real-time hybrid experimental system with actuator delay compensation and its application to a piping system with energy absorber*, *Earthquake Engng Struct. Dyn.*, **28**, 1121–1141.
- Kreyszig, E. 1999 *Advanced Engineering Mathematics, 8th Edition*, New York: Wiley.
- Mahin, S.A. & Shing, P.B. 1985 *Pseudodynamic method of seismic testing*, *J. Strut. Engng*, **111**, 1482–1503.
- Mosqueda, G. 2003 *Continuous hybrid simulation with geographically distributed sub-structures*. Ph.D. thesis, University of California, Berkeley.
- Nakashima, M., Kato, H. & Takaoka, E. 1992 *Development of real-time pseudo dynamic testing*, *Earthquake Engng Struct. Dyn.*, **21**, 779–792.
- Nakashima, M. & Masaoka, N. 1999 *Real-time on-line test for MDOF systems*, *Earthquake Engng Struct. Dyn.*, **28**(4), 393–420.
- Neild, S.A., Drury, D., Wagg, D.J., Stoten, D.P. & Crewe, A.J. Implementing real time adaptive control methods for substructuring of large structures *Proc. of the 3rd World Conf. on Structural Control, Como, Italy, 7-12 April 2002*, pp. 811–817.
- Neild, S.A., Stoten, D.P., Drury, D. & Wagg, D.J. 2005 *Control issues relating to real-time substructuring experiments using a shaking table*, *Earthquake Engng Struct. Dyn.*, **34**(9), 1171–1192.
- Pinto, A.V., Pegon, P., Magonette, G. & Tsionis, G. 2004 *Pseudo-dynamic testing of bridges using non-linear substructuring*, *Earthquake Engng Struct. Dyn.*, **33**(11), 1125–1146.
- Sivaselvan, M.V., Reinhorn, A., Liang, Z. and Shao, X. Real-time dynamic hybrid testing of structural systems *Proc. of the 13th World Conf. Earthquake Engineering, Vancouver, Canada, 1-6 August, 2004*, Paper No. 1644, pp. 1–14.
- Wagg, D.J. & Stoten, D.P. 2001 *Substructuring of dynamical systems via the adaptive minimal control synthesis algorithm*, *Earthquake Engng Struct. Dyn.*, **30**, 865–877.
- Wallace, M.I., Wagg, D.J. & Neild, S.A. Multi-actuator substructure testing with applications to earthquake engineering: how do we assess accuracy? *Proc. of the 13th World Conf. Earthquake Engineering, Vancouver, Canada, 1-6 August, 2004*, Paper No. 3241, pp. 1–14.
- Williams, M.S. & Blakeborough, A. 2001 *Laboratory testing of structures under dynamic loads: an introductory review*, *Phil. Trans. R. Soc. Lond., Series A*, **359**, 1651–1669.