A novel control architecture for marginally stable dynamically substructured systems

An Hu[a], Paolo Paoletti[a,*]

[a]*School of Engineering, University of Liverpool, L69 3GH Liverpool, UK.*

Abstract

Dynamic substructuring is a popular hybrid testing technique that mixes physical testing with numerical simulations. Research in this domain has mainly focused so far on structures and decomposition strategies that result in asymptotically stable systems. However, several cases fall outside this family, with either the physical or the numerical part being only marginally stable post-decomposition. A typical example of such scenario is when a mass is split between physical and numerical parts. For generic structures with marginally stable subsystems, current techniques often fail to deliver satisfactory performance or require modifications to the original structure. In this paper, analytical results - valid for generic structures - are derived to highlight the potential shortcomings of current techniques and to motivate the proposal for a novel control architecture to enable dynamic substructuring with marginally stable subsystems. By selecting a suitable set of signals for control design and by augmenting the control strategy via a local controller for the marginally stable subsystem, the proposed method allows accurate hybrid testing without relying on the modifications of the original structure that classical approaches requires. This technique is suitable for all of the mass split structures, and indeed for all marginally stable structures, which means that the results are applicable to a wide range of hybrid testing problems. A simulated benchmark problem of hy-

*Corresponding author

Email addresses: an.hu@liverpool.ac.uk (An Hu), P.Paoletti@liverpool.ac.uk (Paolo Paoletti)

brid testing for vibrating structures is included to demonstrate the effectiveness of the proposed technique.

Keywords: Dynamically Substructured Systems, Hybrid testing, Marginal Stability, Mass split, Vibration.

1. Introduction

Hybrid testing and real-time hybrid simulation are popular approaches to analyse the dynamics of complex structures without having to perform full scale scale physical testing, and to explore vibration control strategies for such structures [1, 2]. These techniques can be applied to the verification of building structure design [3] and the verification of control effect on the rail vehicle [4], just to cite a few examples. A comprehensive guidance for the application of such techniques is also provided in [5]. In this paper, the focus is on dynamically substructured systems (DSS), a family of hybrid testing approaches that combines the advantages of physical experiments and of numerical simulations. The basic approach is illustrated in Figure 1, where an original structure is split into a physical system coupled with a subsystem that is numerically simulated. In order for the decomposed structure to behave as the original structure, kinetic (displacement) and dynamic (force) balances should be satisfied at the interface. However, only one of these constraints can be exactly satisfied and closed-loop control needs to be used to satisfy the other, see discussion below. An actuator is then placed at the interface in the physical system with the aim of making the decomposed structure exhibit the same dynamical response of the original structure.

Two main approaches have been proposed in the literature to achieve this goal: force control and position control. In the force control setting, the displacement of the physical part at the interface is measured and transmitted as an input to the numerical part (indicated by *synchronised signal* S_s in Figure 1). The simulator used in the numerical part then evaluates the force at the interface (*numerical signal* S_n), which is then compared with the measured physical

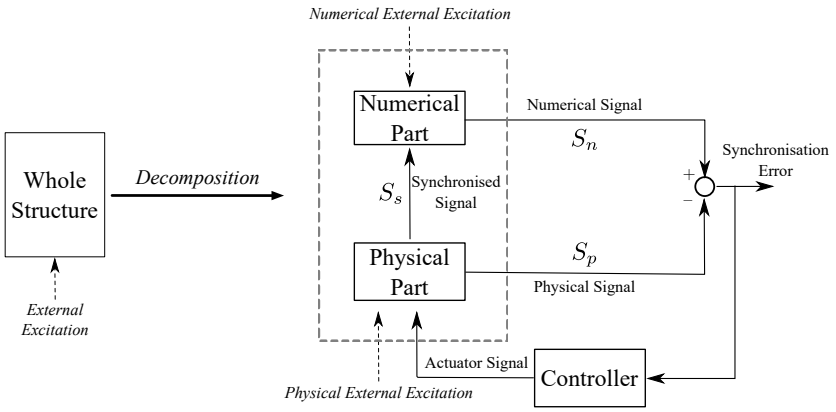


Figure 1: Typical DSS testing approach: the original whole structure shown on the left is split into a numerical part and a physical part. The external excitation applied to the original structure can either be applied to either the numerical or the physical part, according to the chosen DSS decomposition strategy. A synchronisation signal is measured on the physical part and transmitted as input to the numerically simulated subsystem. The goal of the controller is to minimise the error between the numerical and the physical signals, so that the decomposed DSS system (enclosed in the dashed box) behaves as the original whole structure.

force generated by the actuator located at the interface in the physical part (*physical signal* S_p). A feedback controller is designed to minimise the *synchronisation error* e between the simulated force and the actuator force, so that the interface becomes transparent and the decomposed system behaves as the original system. In the position control setting, the role of force and displacement is swapped. A mix of force and position control has also been proposed recently for some specific structures such as bearings [6]. The detailed description of substructuring techniques can be found to [7], whereas some real-world applications of this technique are described in the papers by Yamaguchi [8], Tu [9] and Stoten [10], just to cite a few examples. Similar substructuring techniques have also been recently proposed to understand vibration and uncertainty propagation in complex structures in presence of limited knowledge about the structures themselves [11, 12]. The possibility of using model updating to complete hybrid testing has also been proposed to improve accuracy in these cases [13, 14]. The presence of the actuator at the interface introduces additional challenges due

to the delay and internal dynamics associated with such actuator, which need to be compensated for to obtain good synchronisation performance, see for example [15, 16, 17, 18, 19], as well as [20] where eigenvalue analysis is used to understand the instability caused by delay.

In the vast majority of the literature, the interface separating the physical system from the numerical system is placed across elements such as springs and dampers connecting different masses. Placing the interface across a mass poses additional challenges. In fact, when a mass is split between physical and numerical systems, the action-reaction principle imposes that the physical and numerical forces must be equal [21]. The only way to achieve synchronisation is therefore to generate the physical force via the interface actuator and then use its opposite value as the reaction force at the interface in the simulated subsystem. The selection of the synchronisation error then fall among displacement, velocity or acceleration at the interface. Currently most of the approaches explored in the literature focus on decomposition strategies resulting in stable subsystems, see for example [22, 23, 24]. Wallace [25], Tu [26], Barton [27] and Terkovics [28] discussed the effect of marginally stable decomposition and proposed methods to eliminate the synchronisation error, but these approaches require a very accurate knowledge of the system dynamics to carefully tune the actuator and the synchronising controller, thus severely limiting their applicability in real-world applications. Other approaches have been considered as well. For example, in [29] the original structure was modified to make all subsystems asymptotically stable after the decomposition, with the goal of obtaining better results when synchronising displacements, even at the cost of not representing the original behaviour exactly.

In this paper, a comprehensive analysis of the dynamics of DSS in presence of marginally stable subsystems, resulting for example by placing the interface across a mass, is performed. This analysis is then used to propose a novel control architecture that overcomes the drawbacks described above for current DSS control approaches. The proposed methodology significantly broadens the application of DSS to structures that are either out of range for current techniques

or that would require very accurate (and hence unreliable) tuning. At odd with most of the available literature, the proposed control architecture is suitable for all of the marginally stable structures and not linked to specific experimental implementations.

The paper is organised as follows. A benchmark problem representing a generic vibrating structure is described in section 2 to motivate the need for alternative control schemes in presence of marginally stable subsystems. The challenges of DSS control design in presence of marginally stable subsystems are discussed in section 3, together with the proposed methodology to overcome those challenges. Numerical results are discussed in section 4 to support the analyses and to demonstrate the effectiveness of the proposed approach to control design. Finally, some conclusions and opportunities for future work are discussed in section 5.

2. DSS for vibrating structures with mass-split

In this section, the generic mass, spring and damper structure shown in Figure 2 is used as a benchmark system to motivate the need for alternative DSS control schemes. Note that, despite its simplicity, such system can represent a whole class of vibration testing problem, see for example [30, 26, 25] where this and similar models has been used. Here, a disturbance $d(t)$ is applied as displacement of the support at one end of the structure, whereas an external force $F(t)$ may be applied to the other end (such force will represent, for example, the action of the actuator used in DSS substructuring in the following sections of the paper). Note that when the interface is placed across a mass - a case scarcely studied in the literature - the subsystem on the right is marginally stable. Therefore, such system can be used as a benchmark case to introduce the proposed approach.

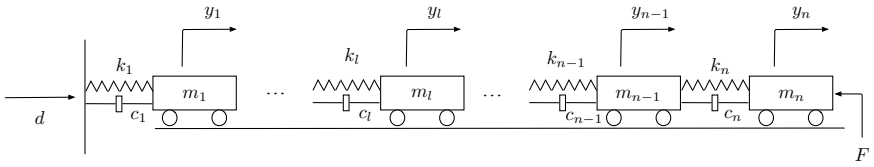


Figure 2: Schematics for a generic lumped parameter vibration problem.

2.1. Original dynamics

The equations of motion describing the behaviour of the original system shown in Figure 2 (sometimes referred to as *whole system* in the DSS literature [9]) can be obtained by imposing force balance at each mass

$$\begin{cases} m_n\ddot{y}_n = -k_n(y_n - y_{n-1}) - c_n(\dot{y}_n - \dot{y}_{n-1}) - F & i = n \\ \quad\vdots & \\ m_i\ddot{y}_i = -k_i(y_i - y_{i-1}) - c_i(\dot{y}_i - \dot{y}_{i-1}) & \\ \qquad + k_{i+1}(y_{i+1} - y_i) + c_{i+1}(\dot{y}_{i+1} - \dot{y}_i) & 2 \le i \le n-1 \\ m_1\ddot{y}_1 = -k_1(y_1 - d) - c_1(\dot{y}_1 - \dot{d}) + k_2(y_2 - y_1) + c_2(\dot{y}_2 - \dot{y}_1) & i = 1 \end{cases} \tag{1}$$

where m_i is the i-th mass, c_i the i-th damping coefficient, k_i is the i-th spring stiffness, y_i is the displacement of the i-th mass and the dot indicates time derivative. The displacement of the support is indicated by $y_0 = d$.

The equivalent form in the frequency domain can then be simply obtained by applying the Laplace transform to (1), thus obtaining

$$y_i = \frac{Num_i(s)}{Den_i(s)} y_{i-1} - \frac{Num1_i(s)}{Den1_i(s)} F, \quad 1 \le i \le n \tag{2}$$

where $Num_i(s)$ and $Den_i(s)$ represent, respectively, the numerator and the denominator of the transfer function between the position of the i-th and $(i-1)$-th masses. Similarly $Num1_i(s)$ and $Den1_i(s)$ refer to the transfer function between the position of the i-th mass and the applied force F.

The linearity of the system dynamics (1) allows studying the influence of the disturbance d and the force F independently, thanks to the superposition principle. By exploring the fact that the system of Figure 2 is composed of a

series of elementary mass-spring-damper subsystems, recursive expressions for the transfer functions in (2) are derived. Such a recursive representation will play a key role in the discussion and control development below. To this end, the transfer function between i-th mass displacement y_i and the disturbance d is written as

$$\frac{y_i}{d} = \frac{y_1}{d}\frac{y_2}{y_1}\cdots\frac{y_i}{y_{i-1}} = \frac{Num_i^d}{Den_1} = \begin{cases} \dfrac{Den_{i+1}\prod_1^i (c_i s + k_i)}{Den_1} & i \le n-1 \\ \dfrac{\prod_1^i (c_i s + k_i)}{Den_1} & i = n \end{cases} \tag{3}$$

where

$$Den_i = \begin{cases} m_n s^2 + c_n s + k_n & i = n \\ [m_{n-1}s^2 + (c_{n-1}+c_n)s + k_{n-1} + k_n](m_n s^2 + c_n s + k_n) & \\ \qquad - (c_n s + k_n)^2 & i = n-1 \\ \qquad \vdots & \\ [m_i s^2 + (c_i + c_{i+1})s + k_i + k_{i+1}]Den_{i+1} & \\ \qquad - (c_{i+1}s + k_{i+1})^2 Den_{i+2} & i \le n-2 \end{cases} \tag{4}$$

A similar procedure is used to derive the transfer function between the i-th displacements y_i and the external force F, thus obtaining

$$\begin{aligned} \frac{y_i}{F} &= \frac{y_n}{F}\frac{y_{n-1}}{y_n}\cdots\frac{y_i}{y_{i+1}} = -\frac{Num_i^F}{Den1_n} \\ &= \begin{cases} -\dfrac{\prod_{i=2}^n (c_i s + k_i)}{Den1_n} & i = 1 \\ -\dfrac{Den1_{i-1}\prod_{i+1}^n (c_{i+1}s + k_{i+1})}{Den1_n} & 1 < i \le n \end{cases} \end{aligned} \tag{5}$$

where

$$Den1_i = \begin{cases} m_1 s^2 + (c_1 + c_2)s + k_1 + k_2 & i = 1 \\ [m_1 s^2 + (c_1 + c_2)s + k_1 + k_2] & \\ \quad \times [m_2 s^2 + (c_2 + c_3)s + k_2 + k_3] - (c_2 s + k_2)^2 & i = 2 \\ [m_i s^2 + (c_i + c_{i+1})s + k_i + k_{i+1}] Den1_{i-1} & \\ \quad - (c_i s + k_i)^2 Den1_{i-2} & 2 < i \leq n-1 \\ (m_n s^2 + c_n s + k_n) Den1_{n-1} - (c_n s + k_n)^2 Den1_{n-2} & i = n \end{cases} \tag{6}$$

with $Den1_n$ being equal to Den_1.

2.2. Structural decomposition with mass-split

Let us then consider the case where the structure shown in Figure 2 needs to be tested via DSS, with the substructuring interface being placed across the l-th mass. The resultant decomposed structure is shown in Figure 3 where S_1 and S_2 are the internal forces after the decomposition and the following conditions hold

$$\begin{cases} m_{l1} + m_{l2} = m_l \\ m_{l1} m_{l2} \neq 0 \end{cases} \tag{7}$$

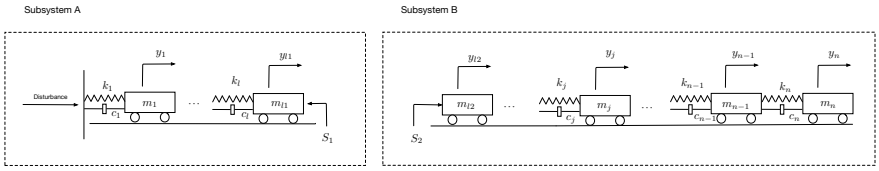


Figure 3: Schematics for DSS decomposition for a generic lumped parameter vibration problem, where the interface is placed across mass l.

Note that subsystem A has the same structure of the original system and the same approach of (2), (3) and (5) can be used to describe its dynamics. On the other hand, subsystem B is structurally different from the original system, with the main difference being that neither the left nor the right end is connected to

any fixed support via springs and dampers. The generalised equation of motion for subsystem B can be derived by imposing force balance at each mass, thus obtaining

$$\frac{y_i}{F'} = \frac{y_{l2}}{F'}\frac{y_{l+1}}{y_{l2}}\cdots\frac{y_i}{y_{i-1}} = \frac{Num_i^{F'}}{Den'_l} = \begin{cases} \dfrac{Den'_{i+1}}{Den'_l} & i = l \\ \dfrac{Den'_{i+1}\prod_{l+1}^{i}(c_i s + k_i)}{Den'_l} & l < i \leq n-1 \\ \dfrac{\prod_{l+1}^{i}(c_i s + k_i)}{Den'_l} & i = n \end{cases} \tag{8}$$

where

$$Den'_i = \begin{cases} m_n s^2 + c_n s + k_n & i = n \\ [m_{n-1}s^2 + (c_{n-1}+c_n)s + k_{n-1} + k_n] & \\ \quad \times (m_n s^2 + c_n s + k_n) - (c_n s + k_n)^2 & i = n-1 \\ \quad \vdots & \\ [m_i s^2 + (c_i + c_{i+1})s + k_i + k_{i+1}]Den'_{i+1} & \\ \quad - (c_{i+1}s + k_{i+1})^2 Den'_{i+2} & l+1 \leq i \leq n-2 \\ (m_{l2}s^2 + c_{l+1}s + k_{l+1})Den'_{l+1} & \\ \quad - (c_{l+1}s + k_{l+1})^2 Den'_{l+2} & i = l \end{cases} \tag{9}$$

and F' is the internal force applied at the interface ($F' = S_1 = S_2$ in traditional DSS). Note that the recursive equation (9) is identical to equation (4) except for the last denominator.

2.3. Stability analysis

The main goal of this section is to prove that the decomposed structure contains a marginally stable part (subsystem B) which makes DSS testing very challenging for existing control techniques. This is proved by showing that Den'_l has some roots at the origin and there are no roots with positive real parts. Similarly, a proof of asymptotic stability of subsystem A and the original system

will be derived. Let us then start from the original system. Note that equation (4) implies

$$Den_i = \prod_{q=i}^{n}(m_q s^2 + c_q s + k_q) + b_i \qquad i \leq n-1 \tag{10}$$

where the first term on the left hand side is an explicit recursion, whereas b_i collects all the remaining terms (see also equation (13) below).

An induction procedure can then be used to prove that equation (10) holds, that b_i always contains a common factor s^2 for each value of the index i, and that all coefficients in b_i are unconditionally positive as the system parameters (m_i, c_i and k_i) are always greater than zero. To this end, let us start from Den_{n-1}, which can be rearranged as

$$\begin{aligned} Den_{n-1} &= (m_{n-1}s^2 + c_{n-1}s + k_{n-1})(m_n s^2 + c_n s + k_n) + (c_n s + k_n)m_n s^2 \\ &= \prod_{q=n-1}^{n}(m_q s^2 + c_q s + k_q) + b_{n-1} \end{aligned} \tag{11}$$

thanks to equation (4). In this case $b_{n-1} = (c_n s + k_n)m_n s^2$, therefore Den_{n-1} clearly has a common factor s^2 and all coefficients are unconditionally positive. A similar procedure can be used to prove that equation (10) is satisfied when $i = n-2$ and b_{n-2} has a common factor s^2, as well as that all coefficients are unconditionally positive. Let us then assume that equation (10) holds and b_i has a common factor s^2 and unconditionally positive coefficients for generic $i = j+1$ and $i = j$. Equation (4) implies

$$
\begin{aligned}
Den_{j-1} &= [m_{j-1}s^2 + (c_{j-1} + c_j)s + k_{j-1} + k_j]Den_j - (c_j s + k_j)^2 Den_{j+1} \\
&= (m_{j-1}s^2 + c_{j-1}s + k_{j-1})Den_j + (c_j s + k_j)Den_j \\
&- (c_j s + k_j)^2 Den_{j+1} \\
&= (m_{j-1}s^2 + c_{j-1}s + k_{j-1})Den_j \\
&+ (c_j s + k_j)(m_j s^2 + c_j s + k_j) \prod_{q=j+1}^{n} (m_q s^2 + c_q s + k_q) + (c_j s + k_j)b_j \\
&- (c_j s + k_j)^2 \left[\prod_{q=j+1}^{n} (m_q s^2 + c_q s + k_q) + b_{j+1} \right] \\
&= (m_{j-1}s^2 + c_{j-1}s + k_{j-1})Den_j \\
&+ (c_j s + k_j)m_j s^2 \prod_{q=j+1}^{n} (m_q s^2 + c_q s + k_q) \\
&+ (c_j s + k_j)[b_j - (c_j s + k_j)b_{j+1}] \\
&= \prod_{q=j-1}^{n} (m_q s^2 + c_q s + k_q) + (m_{j-1}s^2 + c_{j-1}s + k_{j-1})b_j \\
&+ (c_j s + k_j)m_j s^2 \prod_{q=j+1}^{n} (m_q s^2 + c_q s + k_q) \\
&+ (c_j s + k_j)[b_j - (c_j s + k_j)b_{j+1}] \\
&= \prod_{q=j-1}^{n} (m_q s^2 + c_q s + k_q) + b_{j-1}
\end{aligned}
\tag{12}
$$

where

$$
\begin{aligned}
b_{j-1} &= (m_{j-1}s^2 + c_{j-1}s + k_{j-1})b_j + (c_j s + k_j)m_j s^2 \prod_{q=j+1}^{n} (m_q s^2 + c_q s + k_q) \\
&+ (c_j s + k_j)[b_j - (c_j s + k_j)b_{j+1}]
\end{aligned}
\tag{13}
$$

Therefore b_{j-1} also has the common factor s^2, completing the the proof that b_i always contains the term s^2 for each value of i. The proof provided in the appendix shows that a polynomial with a even maximum degree and all unconditionally positive coefficients can only have negative real roots, purely imaginary

roots or conjugated complex roots with negative real parts. Therefore, it is necessary to prove that all coefficients in $(c_j s+k_j)[b_j-(c_j s+k_j)b_{j+1}]$ are unconditionally positive. Note that this condition is already satisfied for b_{n-2} and b_{n-3}. To use induction, let us then assume that b_{j-1} and $(c_j s+k_j)[b_j-(c_j s+k_j)b_{j+1}]$ are unconditionally positive. Then b_{j-2} can be expressed as

$$\begin{aligned}
b_{j-2} &= (m_{j-2}s^2+c_{j-2}s+k_{j-2})b_{j-1} \\
&\quad + (c_{j-1}s+k_{j-1})m_{j-1}s^2\prod_{q=j}^{n}(m_q s^2+c_q s+k_q) \\
&\quad + (c_{j-1}s+k_{j-1})[b_{j-1}-(c_{j-1}s+k_{j-1})b_j] \\
&= (m_{j-2}s^2+c_{j-2}s+k_{j-2})b_{j-1} \\
&\quad + (c_{j-1}s+k_{j-1})m_{j-1}s^2\prod_{q=j}^{n}(m_q s^2+c_q s+k_q) \\
&\quad + (c_{j-1}s+k_{j-1})\{(m_{j-1}s^2+c_{j-1}s+k_{j-1})b_j \\
&\quad + (c_j s+k_j)m_j s^2\prod_{q=j+1}^{n}(m_q s^2+c_q s+k_q) \\
&\quad + (c_j s+k_j)[b_j-(c_j s+k_j)b_{j+1}]-(c_{j-1}s+k_{j-1})b_j\} \\
&= (m_{j-2}s^2+c_{j-2}s+k_{j-2})b_{j-1} \\
&\quad + (c_{j-1}s+k_{j-1})m_{j-1}s^2\prod_{q=j}^{n}(m_q s^2+c_q s+k_q) \\
&\quad + (c_{j-1}s+k_{j-1})\{m_{j-1}s^2 b_j+(c_j s+k_j)m_j s^2\prod_{q=j+1}^{n}(m_q s^2+c_q s+k_q) \\
&\quad + (c_j s+k_j)[b_j-(c_j s+k_j)b_{j+1}]\}
\end{aligned} \tag{14}$$

where

$$\begin{aligned}
&(c_{j-1}s+k_{j-1})[b_{j-1}-(c_{j-1}s+k_{j-1})b_j] \\
&= (c_{j-1}s+k_{j-1})\{m_{j-1}s^2 b_j+(c_j s+k_j)m_j s^2\prod_{q=j+1}^{n}(m_q s^2+c_q s+k_q) \\
&+ (c_j s+k_j)[b_j-(c_j s+k_j)b_{j+1}]\}
\end{aligned} \tag{15}$$

Therefore, all the coefficients of $(c_{j-1}s + k_{j-1})[b_{j-1} - (c_{j-1}s + k_{j-1})b_j]$ are unconditionally positive, and this implies that the coefficients of b_{j-2} are unconditionally positive as well. Therefore, by induction, all coefficients in b_j are unconditionally positive for each index j, as well as all coefficients in each Den_j. It can also be observed that the maximum degree for Den_j is $2j$, while the maximum degree for b_j is $2j-1$. Moreover, in Den_j the coefficients of odd degree terms are functions of the damping coefficients c_i only. Therefore, the proof provided in the appendix can be exploited to show that Den_j can only admit negative real roots, complex roots with negative real parts and purely imaginary roots. However, for Den_j to admit purely imaginary roots, at least some of the odd degree terms should be zero, but this would imply that some of the damping coefficients c_i are equal to zero, which is in conflict with the condition that all system parameters are greater than zero. Therefore, Den_j can only have negative real roots or conjugated complex roots with negative real parts, indicating that the original system and subsystem A are asymptotically stable.

A similar procedure can be applied to Den'_l and equation (9) can be rearranged as

$$Den'_l = m_{l2}s^2 \prod_{q=l+1}^{n} (m_q s^2 + c_q s + k_q) + b'_l \tag{16}$$

where b'_l also includes the common factor s^2 and its coefficient are unconditionally positive as well. Therefore, all the coefficients in Den'_l include the common factor s^2. After collecting the common factor s^2, a procedure similar to the one used for the original system can be exploited to prove that only negative real roots or conjugated complex roots with negative real parts are included, indicating that subsystem B is only marginally stable.

3. Control: challenges and proposed control architecture

In this section the control challenges introduced by the marginal stability of subsystem B are discussed and then a control architecture to address these challenges is proposed.

3.1. Time domain analysis

As proven in the previous section, the original system and subsystem A are both asymptotically stable, whereas subsystem B is only marginally stable with two poles in the origin. By using simple fraction decomposition, the transfer functions in subsystem A and subsystem B can then be written as

$$y_{l1}(s) = \left(\frac{A_{1(1)}}{s + a_{1(1)}} + \frac{A_{1(2)}}{s + a_{1(2)}} + \cdots + \frac{A_{1(2l)}}{s + a_{1(2l)}} \right) d - \left(\frac{A_{2(1)}}{s + a_{1(1)}} + \frac{A_{2(2)}}{s + a_{1(2)}} + \cdots + \frac{A_{2(2l)}}{s + a_{1(2l)}} \right) F' \tag{17}$$

$$y_{l2}(s) = \left(\frac{B_1}{s} + \frac{B_2}{s^2} + \frac{B_3}{s + a_{2(3)}} + \cdots + \frac{B_{2(n-l+1)}}{s + a_{2[2(n-1+1)]}} \right) F' \tag{18}$$

where $a_{1(i)}$ and $a_{2(i)}$ have negative real parts.

Therefore $y_{l1}(s)$ always shows a bounded response for bounded inputs, whereas the first two terms in (18) imply that unbounded responses can be exhibited by subsystem B even in presence of bounded inputs when B_1 and B_2 are not zero. In the time domain, this implies that the response of subsystem B can include an offset and a drift even in presence of constant or null inputs, i.e.

$$y_{l1} = y_1^*(t) \tag{19}$$

$$y_{l2} = y_2^*(t) + C_1 t + C_2 \tag{20}$$

where $y_1^*(t)$ and $y_2^*(t)$ are the components of the time domain response due to the external excitation, and C_1 and C_2 are obtained from terms B_1 and B_2 in equation (18).

Note that y_{l2} exhibits a linear drift and become unbounded (when F' is bounded

and stable or marginally stable (poles at the imaginary axis but not at the origin)), while $y_2^*(t)$ is always bounded. Moreover, when F' has poles at the origin or unstable, y_{l2} may also exhibit nonlinear drifts. It should also be noted that if F' is unbounded then y_{l1} will be unbounded and nonlinearities may also be included. It is also worth noting that velocities in subsystem B may drift, except when F' is stable or has poles on the imaginary axis but not at the origin. Accelerations in both subsystems are always bounded if F' is bounded, as well as displacements and velocities in subsystem A.

3.2. Challenges using traditional control framework with displacements as synchronised signal

As demonstrated in section 3.1, all displacements in subsystem B can drift and become unbounded even for bounded inputs, hence it is dangerous to physically test subsystem B in practice, although this is possible in theory. Therefore, it is beneficial to have subsystem B tested numerically, to avoid potential damage to the physical subsystem, unless when subsystem B only has single degree of freedom and a carefully tuned controller is used [21, 26, 27]. Let us then consider the case of position control with subsystem B being the numerical system. In this situation, y_{l2} and its derivatives are the numerical signals, whereas y_{l1} and its derivatives are the physical signals. The goal of the DSS synchronising controller is then to make y_{l2} and its time derivatives equal to y_{l1} and its time derivatives. A typical DSS control structure for this situation is then the one shown in Figure 4, where the controller $K(s)$ controls the internal force so that the synchronisation error $\boldsymbol{e}(t)$ is minimised.

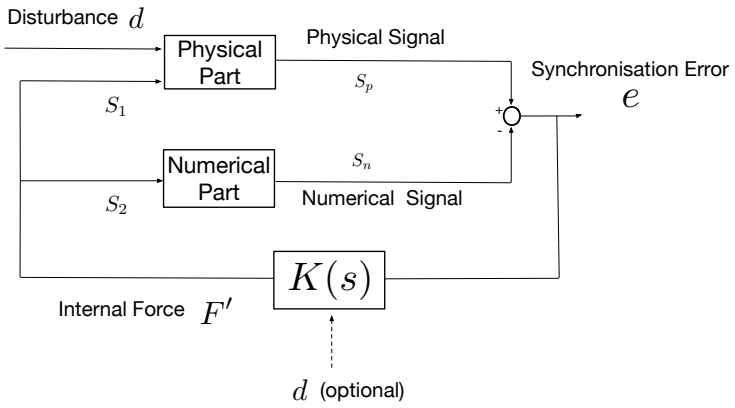


Figure 4: Traditional DSS feedback (plus optional feedforward) control scheme for mass-split models.

In traditional DSS position control design [26] the synchronisation error refers to positions at the interface, i.e. $e(t) = y_{l1}(t) - y_{l2}(t)$. However, the presence of a marginally stable system (subsystem B in this case) implies that the error e can be unbounded due to the terms C_1 and C_2 in (18) even when F' is bounded, and two cases need to be considered. In the first case, the controller $K(s)$ is unable to eliminate the divergence of error e - i.e. it does not contain a term s^2 at the numerator - and therefore the internal force F' will become unbounded as well, inducing drifts or divergence in both y_{l1} and y_{l2}. Clearly, in this case the DSS closed-loop performance is very different from the original asymptotically stable structure, which always shows bounded response to bounded disturbances d. Moreover, unbounded y_{l1} and F' will lead to damage in the experimental apparatus used in subsystem A. Another option is to design the controller $K(s)$ so that it is able to eliminate the divergence of e - i.e. it contains a term s^2 at the numerator - and therefore the internal force F' is bounded. However, due to the term C_1 in equation (20), y_{l2} can still be drifting unbounded. The same procedure can be used to show that the same issues arise whenever position-related signals are used as controller inputs. Therefore, the strategy of synchronising positions that is at the root of the most of DSS position control strategies is not suitable in presence of a marginally stable subsystem,

such as the mass-split case considered here. In this situation, no control strategy can make the DSS closed-loop response equal to the original system response for every combination of initial conditions and disturbances. The conclusion of the analysis above also applies when the optional feedforward controller is added.

3.3. Proposed control architecture

In this section a methodology to overcome these issues and enable DSS testing of marginally stable structures is presented. The basic idea can be summarised into two main points: i) use of velocity-related signals to calculate the synchronisation error $e(t)$ and ii) introduction of a local controller for the marginally stable subsystem (or, whenever possible, adequate management of initial conditions of the marginally stable subsystem).

Let us then start by considering the velocities of subsystems A and B at the interface. According to equation (20), such velocities can can be written as

$$\begin{aligned} \dot{y}_{l1} &= \dot{y}_1^*(t) \\ \dot{y}_{l2} &= \dot{y}_2^*(t) + C_1 \end{aligned} \tag{21}$$

Note that, at odd with equation (20), all the quantities in (21) are bounded when F' is stable or marginally stable (poles on the imaginary axis but not at the origin), therefore the controller only needs to eliminate the offset C_1 and ensure that the forced responses $\dot{y}_1^*$ and $\dot{y}_2^*$ converge towards each other, something that can be easily achieved by having a common factor s in the numerator of the feedback controller transfer function, for example. This is a much simpler control design task, as the controller does not have to deal with potentially unbounded drifts in neither $F'(t)$ nor $e(t)$. Note also that a similar principle can be applied to synchronise the error between accelerations given that $\ddot{y}_{l2}$ is stable, and therefore the acceleration-based synchronisation is similar to standard DSS or hybrid testing. However, using acceleration as synchronised signal means that all signals including accelerations, velocities and displacements should be monitored compared with the proposed velocity synchronisation, which only requires to monitor velocities and displacements. Therefore, the first proposed modification to the control strategy depicted in Figure 4 is the replacement

of position with velocity when calculating the synchronisation error $e(t)$, i.e. setting $e(t) = \dot{y}_1(t) - \dot{y}_2(t)$.

After having replaced positions with velocities and having designed a controller based on this synchronisation error, the DSS behaviour can still differ from the original system response if C_1 is still present due to non-zero initial conditions in subsystem B. Indeed, the control logic for generating the force F' can be written as

$$F'(s) = K(s)e(s) = K(s)\left(\dot{y}_{l1}(s) - \dot{y}_{l2}(s)\right) \tag{22}$$

with $\dot{y}_{l1}$ and $\dot{y}_{l2}$ as in (21). Standard DSS controller design strategies will then derive $K(s)$ so that the forced responses $\dot{y}_1^*$ and $\dot{y}_2^*$ converge to each other. Indeed, techniques such as Linear Substructuring Control (LSC, see [10]) are based on basically inverting the dynamics of the system to be controlled, therefore the controller will have a factor s at the numerator, indicating that the constant C_1 would be neglected. This implies that, if C_1 is not zero then a drift in y_{l2} would still occur. In fact, controller $K(s)$ is aimed at eliminating the error between velocities, and any minor offset in the cancellation of the error may translate to position drift on a long timescale.

One way to solve this issue is to use other signals such as the feedback force [31], or the combination of displacements and their derivatives, as well as the disturbance, instead of the error e as the input for the feedback controller. However, this would require a radical change in the control design approach used for these systems. Therefore, the methodology proposed here is based on relaxing the constraint $S_1 = S_2$, i.e. on allowing the numerical force to be different from the physical force generated by the actuator in the physical subsystem. Relaxing this constraint makes subsystem A and subsystem B partially decoupled, allowing the control and absence of potential drifts, as will be shown for the benchmark example considered in the next section. The resulting control architecture is shown in Figure 5, where an additional local controller $K_2(s)$ is added in the numerical subsystem to make it asymptotically stable. The action of such controller is then simply added to the output of the original velocity-based

DSS controller. The numerator of $K_2(s)$ should be set as small as possible to ensure minimal effect on subsystem B when e is equal to zero. Note that this architecture leaves complete freedom to the designer in terms of choosing the preferred control design for both the local stabilising controller and the overall DSS synchronisation controller, and does not require any accurate tuning to achieve synchronisation.

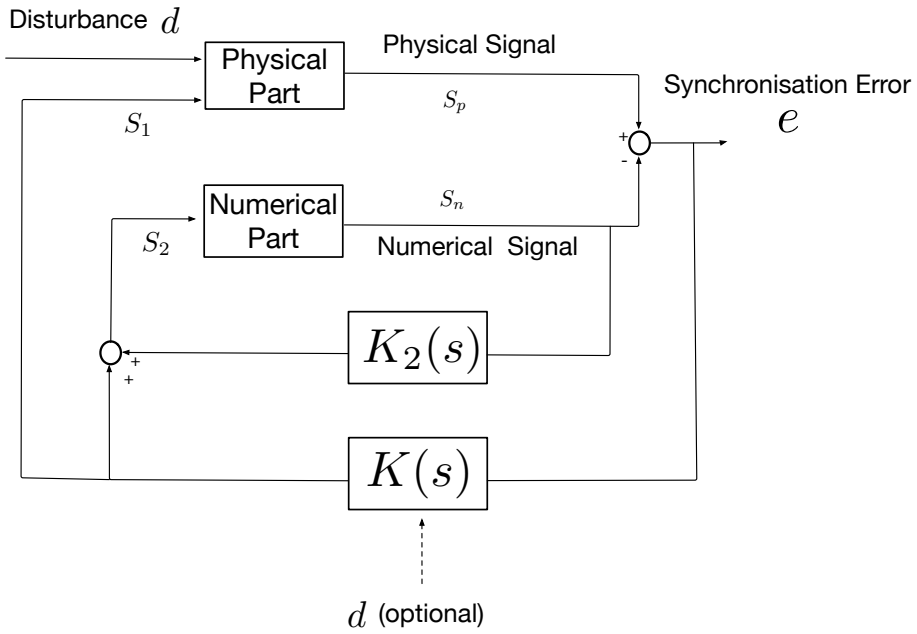


Figure 5: General DSS feedback (plus optional feedforward) control scheme with extra stabiliser for mass split models

Another, even simpler, alternative for solving the problem of potential drifts is based on the observation that the occurrence of the problematic case $(C_1, C_2) \neq (0, 0)$ is due to the initial conditions of the numerical system not being zero. However, very often, setting values for such initial conditions is under complete control of the user performing hybrid testing. Therefore, whenever possible, the initial conditions of subsystem B should be set to zero so that the additional local stabilising controller may become superfluous.

Table 1: Numerical values of parameters used for simulation examples

Index	Mass m_i	Stiffness k_i	Damping c_i
1	$500kg$	$1200N/m$	$300Ns/m$
2	$470kg$	$1100N/m$	$290Ns/m$
3	$440kg$	$1000N/m$	$280Ns/m$
4	$410kg$	$900N/m$	$270Ns/m$
5	$380kg$	$800N/m$	$260Ns/m$

4. Numerical results

In this section a five mass-spring-damper system is used to both highlight the challenges related to DSS control of marginally stable structures and to demonstrate the effectiveness of the control architecture proposed in section 3. The simple nature of the structure enables the whole system to be numerically modelled accurately, eliminating the requirement for experimental validation. Investigation of the effects of experimental uncertainties and potential model mismatches is beyond the scope of this paper. For the purpose of simulation, the numerical values of the parameters are as listed in Table 1, corresponding to natural frequencies $f_n = (0.0730, 0.1965, 0.3073, 0.3941, 0.4514)\,Hz$ and damping ratios $\zeta = (0.0618, 0.1775, 0.2795, 0.3587, 0.4006)$. The disturbance $d(t)$ applied to the structure is a chirp signal whose frequency increases from $0Hz$ to $0.5Hz$ over $45s$ and then is maintained constant for the rest of the simulation. The amplitude of the disturbance is fixed to $1mm$.

To perform hybrid testing, the third mass is split into two parts, as shown in Figure 3, with

$$\begin{cases} m_{31} = 0.6m_3 = 264kg \\ m_{32} = 0.4m_3 = 176kg \end{cases} \tag{23}$$

Finally, the initial conditions of the masses across the interface are set to non-zero values to highlight the problem associated with drifts. In particular, the

initial conditions read

$$\begin{cases} y_{31} = 1mm \\ \dot{y}_{31} = 0.7mm/s \\ y_{32} = 0.5mm \\ \dot{y}_{32} = 0.2mm/s \end{cases} \tag{24}$$

At first, a traditional DSS control design based on the linear substructuring controller (LSC) technique [30] is used to synchronise position at the interface, in accordance with the control architecture of Figure 4 with $e = \dot{y}_{31} - \dot{y}_{32}$. LSC is a feedforward plus feedback control scheme, where the control input reads $u(s) = k_d(s)d(s) + k_e(s)e(s)$. To design the feedforward gain $k_d(s)$ and the feedback gain $k_e(s)$, the error and the actuator dynamics are written as

$$e(s) = G_d(s)d(s) - G_2F'(s) = G_d(s)d(s) - G_u u(s) \tag{25}$$

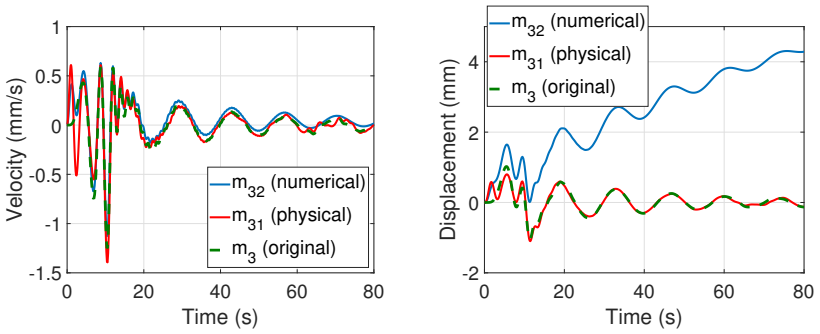
$$F'(s) = S_1(s) = \frac{b}{s+a}u(s) \tag{26}$$

For the purpose of the numerical example considered here, the actuator dynamics parameters were set to $a = b = 10$. The feedforward gain then reads [30]

$$\begin{aligned} k_d(s) &= \frac{G_d(s)}{G_u(s)} \\ &= \frac{4.1464s^2(s+10)(s+4)(s+3.793)}{(s^2+0.05669s+0.2106)(s^2+0.4385s+1.525)(s^2+1.081s+3.74)} \\ &\times \frac{(s+3.571)(s^2+0.9769s+3.052)(s^2+2.534s+8.311)}{(s^2+1.781s+6.162)(s^2+2.262s+7.969)} \end{aligned} \tag{27}$$

whereas the feedback gain $k_e = 10$ was determined using root loci methods.

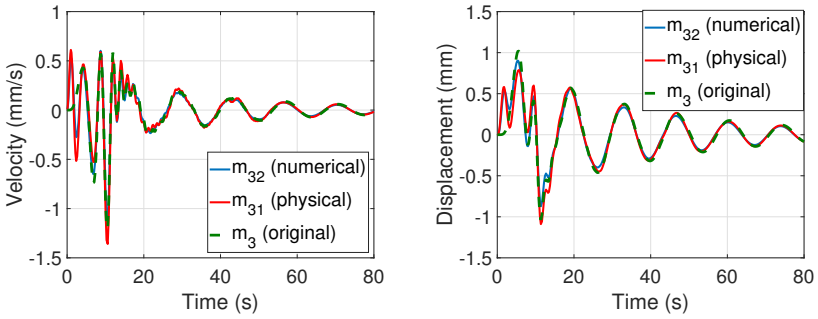
Figure 6 shows that such controller is able to almost synchronise velocities, but a constant offset is present in Figure 6a which translates to a drift in y_{32}, as shown in Figure 6b. Such control failure is in full agreement with the analysis of section 3.



(a) (b)

Figure 6: Velocities (a) and displacements (b) at the interface obtained with the traditional control architecture of Figure 4 with $e = \dot{y}_{31} - \dot{y}_{32}$ and non-zero initial conditions.

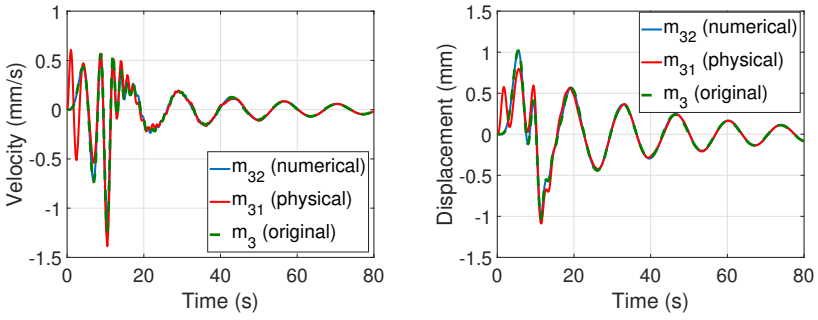
The proposed control architecture, shown in Figure 5, is then used on the same system, keeping the LSC to control S_1 and adding a simple proportional integral (PI) controller to stabilise the numerical subsystem B by adding its action to S_2. In this case, the proportional gain was set to $k_p = 50$, whereas the integral gain was set to $k_I = 25$. Note that this choice of parameters ensure that the amplitude response of the controller K_2 is much smaller than the amplitude of the frequency response of subsystem B, therefore the dynamic response of this latter is minimally influenced by the controller. The results are shown in Figure 7, indicating that now the closed-loop DSS response closely matches the original response, demonstrating that the proposed control architecture can be successfully used to perform DSS-based hybrid testing even in presence of marginally stable subsystems. In addition, tuning of the PI gains is not required and any stabilising PI controller provides very similar results. It is also advised to use small gains to minimise the effect of the PI controller on the response of subsystem B in closed-loop, as demonstrated in the simulation.



(a) (b)

Figure 7: Velocities (a) and displacements (b) at the interface obtained with the proposed control architecture of Figure 5 with $e = \dot{y}_{31} - \dot{y}_{32}$, non-zero initial conditions and a stabilising PI local controller for subsystem B.

Finally, Figure 8 shows the performance achievable with the same controller of Figure 6, when the user has the power of setting all the initial conditions in subsystem B to zero. Note that, in agreement with the analysis of section 3, also in this case the problem of drifts does not occur and the DSS closed-loop response tracks the original response as desired.



(a) (b)

Figure 8: Velocities (a) and displacements (b) at the interface obtained with the traditional control architecture of Figure 4 with $e = \dot{y}_{31} - \dot{y}_{32}$, when the user can set the initial conditions of subsystem B to zero.

These results highlight that, when a marginally stable subsystem B has non-

zero initial conditions, a traditional DSS control design approach is not capable of eliminating the drift in subsystem B. The proposed control architecture addresses such issue by adding an extra stabilising local controller in subsystem B only, which makes the overall closed-loop response close to the original system response, with a minimal extra effort due to the tuning of the additional controller. Note that, in all cases, non-zero initial conditions for subsystem A do not affect synchronisation performance, as subsystem A is asymptotically stable. On the other hand, initial conditions for subsystem B play a key role. In fact, if the users have the power of setting all such conditions to zero before starting the hybrid tests, then the additional local controller $K_2(s)$ of Figure 5 is not needed, as the response of subsystem B can track the response of the original system using the DSS controller $K(s)$ only.

5. Conclusions

In this paper a detailed analysis of issues created by the presence of marginally stable subsystems in DSS-based hybrid testing has been conducted, with the case of mass-split taken as a benchmark system. The results discussed in this paper shows that in this scenario traditional DSS control design approaches struggle to provide satisfactory performance or even fail to stabilise the marginally stable substructure. The novel control architecture shown in Figure 5 - based on velocity feedback and on the introduction of an additional local controller - has been proposed to solve this issue. The proposed analysis and control architecture is generic and can be applied to hybrid testing of any vibrating structure. Analytical and numerical results show that the proposed architecture is capable of making the DSS closed-loop response track the original system response, as desired. The analysis has also shown that the introduction of the additional local controller $K_2(s)$ can be avoided when the user has the power of setting all the initial conditions of the marginally stable subsystem to zero.

The results presented in this paper unlock the possibility of performing DSS-based hybrid testing in presence of marginally stable subsystems, an area that

has been scarcely explored so far. Potential future research directions include an analysis of robustness of the proposed technique (e.g. with respect to noise, model mismatch and experimental uncertainty), extensions to testing of nonlinear structures and applications to finite element based models with more complex boundary conditions.

Funding

This research was partially supported by the University of Liverpool/National Tsing Hua University Dual PhD scheme.

Declaration of Competing Interest

The author declares no conflict of interest.

References

[1] M. Hakuno, M. Shidawara, T. Hara, Dynamic destructive test of a cantilever beam, controlled by an analog-computer, in: Proceedings of the Japan society of civil engineers, Vol. 1969, Japan Society of Civil Engineers, 1969, pp. 1–9.

[2] M. Nakashima, H. Kato, E. Takaoka, Development of real-time pseudo dynamic testing, Earthquake Engineering & Structural Dynamics 21 (1) (1992) 79–92.

[3] R. Christenson, Y. Z. Lin, A. Emmons, B. Bass, Large-scale experimental verification of semiactive control through real-time hybrid simulation, Journal of Structural Engineering 134 (4) (2008) 522–534.

[4] P. Terwiesch, T. Keller, E. Scheiben, Rail vehicle control system integration testing using digital hardware-in-the-loop simulation, IEEE Transactions on Control Systems Technology 7 (3) (1999) 352–362.

[5] C. E. Silva, D. Gomez, A. Maghareh, S. J. Dyke, B. F. Spencer Jr, Benchmark control problem for real-time hybrid simulation, Mechanical Systems and Signal Processing 135 (2020) 106381.

[6] N. Nakata, R. Erb, M. Stehman, Mixed force and displacement control for testing base-isolated bearings in real-time hybrid simulation, Journal of Earthquake Engineering 23 (6) (2019) 1055–1071.

[7] D. D. Klerk, D. J. Rixen, S. N. Voormeeren, General framework for dynamic substructuring: History, review and classification of techniques, AIAA Journal 46 (5) (2008) 1169–1181.

[8] T. Yamaguchi, D. P. Stoten, Synthesised H_∞ / μ control design for dynamically substructured systems, Journal of Physics: Conference Series 744 (1) (2016) 012205.

[9] J. Y. Tu, Development of numerical-substructure-based and output-based substructuring controllers, Structural Control and Health Monitoring 20 (6) (2013) 918–936.

[10] D. P. Stoten, J. Y. Tu, G. Li, Synthesis and control of generalized dynamically substructured systems, Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 223 (3) (2009) 371–392.

[11] T. Kranjc, J. Slavi, M. Boltear, An interface force measurements-based substructure identification and an analysis of the uncertainty propagation, Mechanical Systems and Signal Processing 56-57 (2015) 2 – 14.

[12] J. Kullaa, Virtual sensing of structural vibrations using dynamic substructuring, Mechanical Systems and Signal Processing 79 (2016) 203 – 224.

[13] G. Ou, S. J. Dyke, A. Prakash, Real time hybrid simulation with online model updating: An analysis of accuracy, Mechanical Systems and Signal Processing 84 (2017) 223–240.

[14] Z. Mei, B. Wu, O. S. Bursi, G. Xu, Z. Wang, T. Wang, X. Ning, Y. Liu, Hybrid simulation with online model updating: Application to a reinforced concrete bridge endowed with tall piers, Mechanical Systems and Signal Processing 123 (2019) 533–553.

[15] B. M. Phillips, B. F. Spencer Jr, Model-based feedforward-feedback actuator control for real-time hybrid simulation, Journal of Structural Engineering 139 (7) (2012) 1205–1214.

[16] B. Wu, Z. Wang, O. S. Bursi, Actuator dynamics compensation based on upper bound delay for real-time hybrid simulation, Earthquake Engineering & Structural Dynamics 42 (12) (2013) 1749–1765.

[17] G. Ou, A. I. Ozdagli, S. J. Dyke, B. Wu, Robust integrated actuator control: experimental verification and real-time hybrid-simulation implementation, Earthquake Engineering & Structural Dynamics 44 (3) (2015) 441–460.

[18] X. Ning, Z. Wang, H. Zhou, B. Wu, Y. Ding, B. Xu, Robust actuator dynamics compensation method for real-time hybrid simulation, Mechanical Systems and Signal Processing 131 (2019) 49–70.

[19] G. A. Fermandois, Application of model-based compensation methods to real-time hybrid simulation benchmark, Mechanical Systems and Signal Processing 131 (2019) 394–416.

[20] A. Maghareh, S. Dyke, S. Rabieniaharatbar, A. Prakash, Predictive stability indicator: a novel approach to configuring a real-time hybrid simulation, Earthquake Engineering & Structural Dynamics 46 (1) (2017) 95–116.

[21] S. Neild, D. Stoten, D. Drury, D. Wagg, Control issues relating to real-time substructuring experiments using a shaking table, Earthquake engineering & structural dynamics 34 (9) (2005) 1171–1192.

[22] X. Shao, A. M. Reinhorn, M. V. Sivaselvan, Real-time hybrid simulation using shake tables and dynamic actuators, Journal of Structural Engineering 137 (7) (2010) 748–760.

[23] D. P. Stoten, T. Yamaguchi, Y. Yamashita, Dynamically substructured system testing for railway vehicle pantographs, Journal of Physics: Conference Series 744 (1) (2016) 012204.

[24] J. E. Carrion, B. F. Spencer Jr, Model-based strategies for real-time hybrid testing, Tech. rep., Newmark Structural Engineering Laboratory. University of Illinois at Urbana-Champaign (2007).

[25] M. Wallace, J. Sieber, S. Neild, D. Wagg, B. Krauskopf, Stability analysis of real-time dynamic substructuring using delay differential equation models, Earthquake engineering & structural dynamics 34 (15) (2005) 1817–1832.

[26] J.-Y. Tu, W.-D. Hsiao, C.-Y. Chen, Modelling and control issues of dynamically substructured systems: adaptive forward prediction taken as an example, Proc. R. Soc. A 470 (2168) (2014) 20130773.

[27] D. A. Barton, Control-based continuation: Bifurcation and stability analysis for physical experiments, Mechanical Systems and Signal Processing 84 (2017) 54–64.

[28] N. Terkovics, S. Neild, M. Lowenberg, R. Szalai, B. Krauskopf, Substructurability: the effect of interface location on a real-time dynamic substructuring test, Proceedings of the Royal Society A 472 (2192) (2016) 20160433.

[29] D. P. Stoten, A comparative study and unification of two methods for controlling dynamically substructured systems, Earthquake Engineering & Structural Dynamics 46 (2) (2017) 317–339.

[30] D. P. Stoten, R. A. Hyde, Adaptive control of dynamically substructured systems: The single-input single-output case, Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 220 (2) (2006) 63–79.

[31] G. Li, Dynamically substructured system frameworks with strict separation of numerical and physical components, Structural Control and Health Monitoring 21 (10) (2014) 1316–1333.

A. Distribution of roots for polynomials with even maximum degree and positive coefficients

The aim of this appendix is to prove that polynomials with even degrees and unconditionally positive coefficients only admit roots with negative real parts and purely imaginary roots. This proof is used in Section 2.3 to show that the original whole system and subsystem A are asymptotically stable and that subsystem B is marginally stable.

Let us then consider a polynomial $P(s)$ with a maximum even degree $2n$ and positive coefficients

$$P(s) = a_{2n}s^{2n} + a_{2n-1}s^{2n-1} + \cdots + a_i s^i + \cdots + a_0 \qquad \text{(A.1)}$$
$$a_i > 0, \quad 0 \leq i \leq 2n$$

No positive or zero real roots for $P(s)$ exist, as can be easily shown by direct substitution. Therefore, the roots of $P(s)$ can only include negative real roots, imaginary roots and complex roots. Moreover, a polynomial with a even maximum degree should have an even number of real roots, an even number of conjugated imaginary roots, an even number of conjugated complex roots, or a combination of these three possibilities.

The following proof is composed of two parts. In the first part it is demonstrated that: i) if $P(s)$ admits complex roots with positive real parts then its coefficients are not unconditionally positive, and ii) if $P(s)$ does not admit complex roots with positive real parts then its coefficients are unconditionally positive. These implications are then used in the second part of the proof to prove, exploiting the contraposition principle, that polynomials $P(s)$ with even degree and unconditionally positive coefficients only admit roots with negative real part and, potentially, purely imaginary roots.

Let us then start the proof by considering the case where $P(s)$ only admits

conjugated imaginary roots only, i.e.

$$P(s) = \prod_{q=1}^{n} (a_{2(q)} s^2 + a_{0(q)}), \quad a_{2(q)} > 0,\ a_{0(q)} > 0 \tag{A.2}$$

Therefore, it is obvious that after the expansion the coefficients of odd degree terms s^{2j-1} are equal to zero and all the others are positive. This contradicts the positivity of coefficients of $P(s)$, therefore $P(s)$ can not admit only purely imaginary roots.

Another case is when $P(s)$ only admits negative real roots and conjugated complex roots with negative real parts, i.e.

$$P(s) = \prod_{q=1}^{n} (a_{2(q)} s^2 + a_{1(q)} s + a_{0(q)}) \tag{A.3}$$

where

$$\begin{cases} a_{2(q)} > 0 \\ a_{1(q)} > 0 \\ a_{0(q)} > 0 \end{cases} \tag{A.4}$$

Induction can be used to show that all coefficients in $P(s)$ are unconditionally positive. In fact, this condition is already satisfied when $n = 1$. Let us then assume that this condition is also satisfied when $n = j$ thus

$$P(s) = a_{2j} s^{2j} + a_{2j-1} s^{2j-1} + \cdots + a_i s^i + \cdots + a_0 \tag{A.5}$$

where

$$a_i > 0, \quad 0 \le i \le 2j \tag{A.6}$$

Then when $n = j + 1$ the following condition holds

$$\begin{aligned} P(s) &= (a_{2(j+1)} s^2 + a_{1(j+1)} s + a_{0(j+1)}) \\ &\quad \times (a_{2j} s^{2j} + a_{2j-1} s^{2j-1} + \cdots + a_i s^i + \cdots + a_0) \\ &= a_{2j+2} s^{2j+2} + a_{2j+1} s^{2j+1} + a'_{2j} s^{2j} + \cdots + a'_i s^i + \cdots + a'_0 \end{aligned} \tag{A.7}$$

where

$$\begin{cases} a_{2j+2} = a_{2(j+1)}a_{2j} > 0 \\ a_{2j+1} = a_{2(j+1)}a_{2j-1} + a_{1(j+1)}a_{2j} > 0 \\ a'_i = a_{2(j+1)}a_{i-2} + a_{1(j+1)}a_{i-1} + a_{0(j+1)}a_i > 0 \quad 2 \le i \le 2j \\ a'_1 = a_{1(j+1)}a_0 + a_{0(j+1)}a_1 > 0 \\ a'_0 = a_{0(j+1)}a_0 > 0 \end{cases} \tag{A.8}$$

Therefore, if $P(s)$ only admits negative real roots and conjugated complex roots with negative real parts, then all of its coefficients are unconditionally positive.

If $P(s)$ only admits imaginary roots, conjugated complex roots with negative real parts and negative real roots, the following equation holds

$$\begin{aligned} P(s) &= \prod_{q=1}^{j}(a_{2(q)}s^2 + a_{0(q)}) \prod_{q=1}^{j_1}(a'_{2(q)}s^2 + a'_{1(q)}s + a'_{0(q)}) \\ &= a_{2j+2j_1}s^{2j+2j_1} + a_{2j+2j_1-1}s^{2j+2j_1-1} + \cdots + a_i s^i + \cdots + a_0 \end{aligned} \tag{A.9}$$

$$\begin{cases} a_{2(q)} > 0 \\ a_{0(q)} > 0 \\ a'_{2(q)} > 0 \\ a'_{1(q)} > 0 \\ a'_{0(q)} > 0 \end{cases} \tag{A.10}$$

Then a similar induction procedure can be used to show that all the coefficients a_i are unconditionally positive.

Another case is that when $P(s)$ only admits conjugated complex roots with positive real parts. In this situation the following equation holds

$$P(s) = \prod_{q=1}^{n}(a_{2(q)}s^2 - a_{1(q)}s + a_{0(q)}) \tag{A.11}$$

where

$$\begin{cases} a_{2(q)} > 0 \\ a_{1(q)} > 0 \\ a_{0(q)} > 0 \\ a_{1(q)}^2 - 4a_{2(q)}a_{0(q)} < 0 \end{cases} \tag{A.12}$$

then the induction can be used to show that in the expanded form of $P(s)$, the coefficients for s^i are always negative when i is odd, and the coefficients for s^i are always positive when i is even, thus

$$P(s) = a_{2n}s^{2n} - a_{2n-1}s^{2n-1} + \cdots + (-1)^i a_i s^i + \cdots + a_0 \tag{A.13}$$

where

$$a_i > 0, \quad 0 \le i \le 2n \tag{A.14}$$

In fact, this condition is already satisfied for $n = 1$ and $n = 2$. Let us then assume that this condition is also satisfied when $n = j$, thus

$$P(s) = a_{2j}s^{2j} - a_{2j-1}s^{2j-1} + \cdots + (-1)^i a_i s^i + \cdots + a_0 \tag{A.15}$$

where

$$a_i > 0, \quad 0 \le i \le 2j \tag{A.16}$$

Then for $n = j + 1$ the following condition holds

$$\begin{aligned} P(s) &= (a_{2(j+1)}s^2 - a_{1(j+1)}s + a_{0(j+1)}) \\ &\quad \times (a_{2j}s^{2j} - a_{2j-1}s^{2j-1} + (-1)^i a_i s^i + \cdots + a_0) \\ &= a_{2j+2}s^{2j+2} - a_{2j+1}s^{2j+1} + a'_{2j} + \cdots + a'_i s^i + \cdots + a'_0 \end{aligned} \tag{A.17}$$

where

$$\begin{cases} a_{2j+2} = a_{2(j+1)}a_{2j} \\ a_{2j+1} = a_{2(j+1)}a_{2j-1} + a_{1(j+1)}a_{2j} \\ a'_i = (-1)^{i-2}a_{2(j+1)}a_{i-2} + (-1)^i a_{1(j+1)}a_{i-1} + (-1)^i a_{0(j+1)}a_i \quad 2 \le i \le 2j \\ a'_1 = -(a_{1(j+1)}a_0 + a_{0(j+1)}a_1) \\ a'_0 = a_{0(j+1)}a_0 \end{cases} \tag{A.18}$$

Therefore, a'_i are always negative for odd values of i and a'_i are always positive for even values of i. This contradicts the positivity of coefficients of $P(s)$, therefore $P(s)$ can not admit roots with positive real parts only.

Finally if $P(s)$ admits negative real roots, conjugated complex roots with negative real parts and conjugated complex roots with positive real parts, then

$$\begin{aligned} P(s) &= \prod_{q=1}^{j}(a_{2(q)}s^2 + a_{1(q)}s + a_{0(q)}) \prod_{q=1}^{j_1}(a'_{2(q)}s^2 - a'_{1(q)}s + a'_{0(q)}) \\ &= (a_{2j}s^{2j} + a_{2j-1}s^{2j-1} + \cdots + a_i s^i + \cdots + a_0) \\ &\times (a'_{2j_1}s^{2j_1} - a'_{2j_1-1}s^{2j_1-1} + \cdots + (-1)^i a'_i s^i + \cdots + a'_0) \\ &= a''_{2j+2_j-1}s^{2(j+j_1)} + a''_{2j+2j_1-1}s^{2(j+j_1)-1} + \cdots + a''_i s^i + \cdots + a''_0 \end{aligned} \tag{A.19}$$

where

$$\begin{cases} a_i > 0, & 0 \le i \le j \\ a'_i > 0, & 0 \le i \le j_1 \\ (a'_{1(q)})^2 - 4a'_{2(q)}a'_{0(q)} < 0 & \\ a''_{2j+2j_1} = a_{2j}a'_{2j_1} & \\ a''_0 = a_0 a'_0 & \end{cases} \tag{A.20}$$

and if $j \le j_1$

$$\begin{cases} a''_i = \displaystyle\sum_{q=0}^{i-2j}(-1)^q a_{i-q}a'_q, & 2j \le i \le 2j + 2j_1 - 1 \\ a''_i = \displaystyle\sum_{q=0}^{i}(-1)^{i-q} a_q a'_{i-q}, & 0 \le i \le 2j \end{cases} \tag{A.21}$$

if $j > j_1$

$$\begin{cases} a''_i = \displaystyle\sum_{q=0}^{i-2j_1}(-1)^{i-q} a_q a'_{i-q}, & 2j_1 \le i \le 2j + 2j_1 - 1 \\ a''_i = \displaystyle\sum_{q=0}^{i}(-1)^q a_{i-q}a'_q, & 0 \le i \le 2j_1 \end{cases} \tag{A.22}$$

Therefore, apart from a''_{2j+2j_1} and a''_0, other coefficients are not always positive. The same conclusion applies when $P(s)$ includes purely imaginary roots, negative real roots and complex roots with positive real parts and when $P(s)$ only includes purely imaginary roots and complex roots with positive real parts.

In summary, two main conclusions can be drawn from the conditions above. The first one is that if $P(s)$ does not include complex roots with positive real parts then its coefficients are unconditionally positive. The second conclusion is that if $P(s)$ does include complex roots with positive real parts then its coefficients are not unconditionally positive. Then the law of contraposition - stating that a conditional statement implies that its contrapositive holds as well - can be used to show that two additional conditions hold. The first one is that if the coefficients of $P(s)$ are not unconditionally positive then $P(s)$ admits complex roots with positive real parts. The second condition is that if the coefficients of $P(s)$ are unconditionally positive then $P(s)$ does not include any complex root with positive real part. All together, these conditions imply that a polynomial $P(s)$ with even degree and unconditionally positive coefficients only admits negative real roots, complex roots with negative real parts and purely imaginary roots. The proof is therefore complete.