Dynamically substructured system frameworks with strict separation of numerical and physical components

Guang Li [*]

October 13, 2013

Abstract

A dynamically substructured system (DSS) consists of both physical and numerical components. It is used for the testings of the dynamics of some systems arising from engineering problems to overcome the drawbacks of conventional testing methods. One of the key issues influencing the DSS testing accuracy is from the synchronization of the physical and numerical components. This synchronization can be achieved by a controller (called DSS controller). To facilitate the DSS controller design, this paper develops a sophisticated DSS framework with its variations to further enhance the analysis and design of DSS. The main feature of the proposed DSS framework is that it has a strict separation of numerical and physical substructures, which can enable one to explicitly identify the relations of the substructures and signals within a DSS and thus greatly facilitate more elegant treatments of DSS problems, such as DSS establishment, conversion of different DSSs, as well as uncertainties and measurement noise incorporation in robust control. The proposed framework and its variations unify many DSS problems.

Keywords: dynamically substructured systems, hybrid dynamics testing, mechanical systems

[*] Department of Mechanical and Nuclear Engineering, Pennsylvania State University, State College, PA, 16802. (gul15@psu.edu)

1 Introduction

Dynamically substructured system (DSS) is a testing method used in the dynamics testing community. The uses of the DSS concept can be found in areas such as civil engineering [1, 2], robotics [3], automotive [4] and aerospace [5]. A DSS contains both physical components (called physical substructure) and numerical components (called numerical substructure), which are to be synchronized during a testing so that the DSS response can be as close as possible to that of the original emulated system. This feature of DSS makes it outperform the conventional pure physical or pure numerical testing, scaled size testing, and pseudo-dynamic testing [6]. DSS is also distinguishable from the hardware-in-the-loop (HiL) method, which is used traditionally to test the performance of a controller, with a hardware interface to an embedded numerical plant. For a detailed discussion about different testing methods see [7, 8, 4]. The performance of DSS testing is mainly determined by the synchronization of the physical and numerical substructure. The DSS synchronization problem has been discussed from different angles [9–14]. One way to achieve DSS synchronization is to employ a controller, called DSS controller in this paper.

To facilitate the controller design for DSS, Stoten and Hyde propose a concise and generic framework [8] as shown in Fig. 1, where the signals include the testing signal d, the control signal u and the DSS outputs z_1 and z_2 to be synchronized; G_1 contains the components on which the testing signal d is acted; G_0 and G_2 contain the components which are attached to the actuators controlled by u. The framework emphasizes the relations of the signals of a DSS, and hence facilitates the DSS controller design. This can be seen by transforming Fig. 1 into the equivalent representation of Fig. 2, where we have $G = G_0 + G_1$ and $G_d = G_1 - GK_d$ The DSS synchronization error is determined by $y = SG_d d$ with the loop transfer matrix $L = GK_y$ and the sensitivity function $S = (I + L)^{-1}$. Therefore, abundant linear control theories can be directly applied to the DSS control framework as shown in Fig. 1. Based on this framework, different DSS control strategies have been designed and implemented in real time, such as the linear control based on the root locus design, minimal control synthesis (MCS) [15], H_∞ control [16, 17], numerical-substructure-based and output-based substructuring control [18, 19], and neural network control [20, 21]. Moreover, to cope with the actuator saturation problems, which can occur under some conditions, the model predictive control [22] and anti-windup compensation [23, 17] techniques are also implemented on DSS systems. However, it is

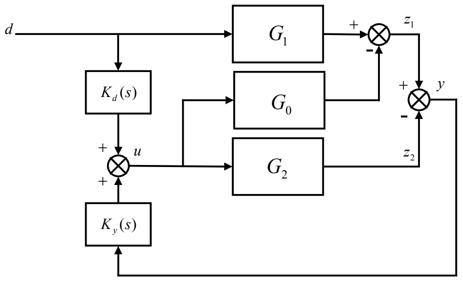


Figure 1: A generic DSS control framework proposed by [8]

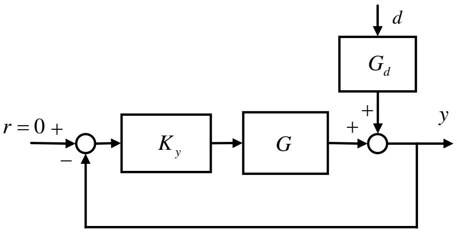


Figure 2: Equivalent representation of Fig. 1

noted that, although this framework is concise for the DSS controller design, it also has some drawbacks:

1. Each block in this framework can contain both physical and numerical components, which makes it difficult to describe the uncertainties from the physical components;
2. The transfer system is contained in the blocks G_0 and G_1, so that the dynamics of the transfer system and its saturation cannot be explicitly taken into account;
3. The interface signals are not explicitly shown, so that some potential problems associated with causality analysis and measurement noise, etc., cannot be taken into account easily in a DSS establishment.

All of the above mentioned points hinder more sophisticated DSS analysis and design. It is noticed that the physical and numerical substructures are divided strictly in many existing DSS establishments for specific problems; however, a generic DSS framework with strict separation of physical and numerical substructures is absent. The main objective of this paper is thus to further refine the framework by Stoten and Hyde [8] by proposing a unified DSS framework with complete separation of the physical and numerical substructures. This helps to gain insight into the DSS formulation and hence significantly facilitates the development of DSS systems and their transformations. Since the interface signals and the dynamics of the transfer system are explicitly signified, the causality problems in the DSS establishment can be conveniently investigated. All these features of

Table 1: Nomenclature

Name	**Description**
Emulated system	The system to which the DSS is emulated.
Physical substructure	The substructure containing all physical components.
Numerical substructure	The substructure containing all numerical components.
Physical block	A subset of physical substructure.
Numerical block	A subset of a numerical substructure.
Numerical signal	The output of a numerical block.
Physical signal	The output of a physical block.
Interface signals	The signals at the interface between the physical and numerical (or physical and physical) blocks.
Constraints signals	Interface signals; the same type of signals from different substructure satisfying some constraints.
DSS outputs	Interface signals; the same type of signals from different substructures, which are to be synchronized.
Transfer system	A set of actuators or shaking tables attached to the physical substructure.

the proposed framework pave the way for the future robust analysis and control of DSS and DSS performance validation when uncertainties in the physical substructures and the measurement noises from the output of the physical substructures are involved.

In this paper, we name this framework as a complete separation framework (CSF). This CSF represents a fairly generic class of DSS. To further enlarge the class of DSS that can be represented by the CSF, two extra frameworks are derived: one is called substructure and signal dual CSF, which is derived by swapping the physical and numerical blocks, and also the constraint and synchronization interface signals in the original CSF; the other one is called signal dual CSF, and it is derived by swapping the constraint and synchronization interface signals in the original CSF. All these CSFs represent a large class of DSS, though it is not claimed to represent all possible DSS cases. To demonstrate the derivation of the CSFs and their transformations, two examples are employed. The first one is a mass-spring-damper system; the second one is a quasi-motorcycle (QM) system built at Bristol University [15], see [15] for the DSS configuration of the QM system in a real experimental testing environment. The nomenclature of this paper is summarized in Table 1.

The structure of this paper is as follows. A CSF is firstly proposed in Section 2. Its corresponding two dual CSFs are derived in Section 3. In Sections 4 and 5, the mass-spring-damper system and the QM system are used to demonstrate the establishments of the CSFs and their transformations. The paper is concluded in Section 6.

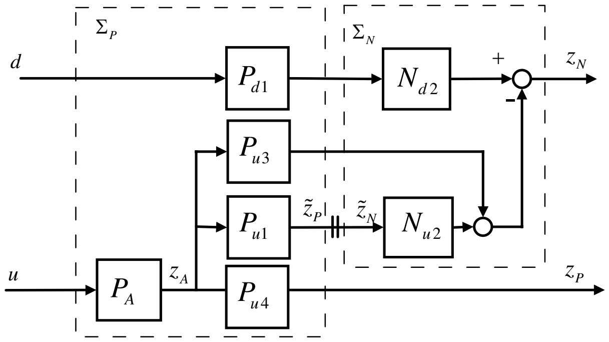


Figure 3: The block diagram of the generic DSS framework

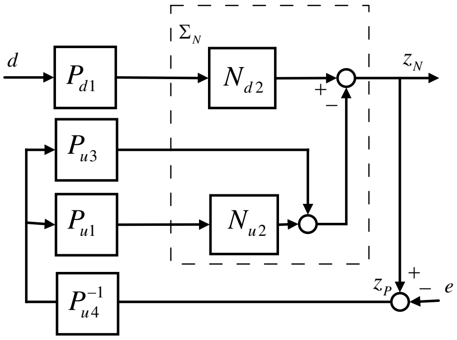


Figure 4: The block diagram of the generic emulated system framework

2 A CSF for DSS

A CSF for DSS is shown as the block diagram in Fig.3 which contains 7 blocks. For a specific system, some of the blocks may not be present, so that its DSS block diagram can be simplified. By defining a block as a group of components, we use P to represent a physical block only containing physical components and N to represent a numerical block only containing numerical components. The subindex is used to distinguish the locations of the blocks. In Fig. 3, P_A consists of the transfer functions of actuators (or transfer systems) in a diagonal form; $\{P_A, P_{d1}, P_{u1}, P_{u3}, P_{u4}\}$ constitute the physical substructure (Σ_P) and $\{N_{d2}, N_{u2}\}$ constitute the numerical substructure (Σ_N). Furthermore, $\{P_{d1}, N_{d2}\}$ represent the components, through which the testing signal d acts on the system. The input and output of P_A are the control signal u and the actuation signal z_A respectively. z_N and z_P are called DSS outputs. We define the difference between z_N and z_P, i.e. $e := z_N - z_P$ as substructuring error and define the constraint signal $\tilde{z} := \tilde{z}_P = \tilde{z}_N$ at the interface between P_{u1} and N_{u2}, where $\tilde{z}_P$ is the output of P_{u1} and $\tilde{z}_N$ is the input of N_{u2}. In Fig. 3, the sign '⊣⊢' denotes the interface. Generally, the output signals of the physical and numerical components are referred to as physical and numerical signals respectively.

For the framework shown in Fig.3, we have the following explanations:

1) The inteface signals (or variables) are defined as those residing at the interface between two adjacent blocks. They are categorized into three types:

    a) The interface signal between numerical block and physical block;

    b) The interface signal between two physical blocks;

    c) The interface signal between two numerical blocks.

    In this paper, we assume that there is no synchronization problem for case c), though sometimes it needs also to be considered (e.g., the situation when two numerical models reside separately in two computers at different locations.). With this assumption, the two numerical blocks linked to each other can be merged into one numerical block.

2) We classify all the available interface signals in cases a) and b) into two groups according to their physical senses: one group contains the constraint signals, denoted by $\{\tilde{z}_N, \tilde{z}_p\}$, while the other group contains DSS output signals, denoted by $\{z_N, z_P\}$. For example, in a mechanical system the interface signals can be forces and displacements, we can choose the forces as the interaction constraint to be satisfied (i.e. the output force of one block is equivalent to the input force of the other block), while the displacements from the two blocks are to be minimized. We call this as DSS force control (see e.g. the case in [24]). In [25], DSS force control and displacement control are referred to as effort actuation and flow actuation respectively. The appropriate choice of force control and displacement control is an essential factor when considering the DSS causality problem.

3) We assume the components in the physical and numerical blocks P_{d1} and N_{d2} in Fig. 3 are not directly connected, although the blocks appear to have a cascade connection. That is, the disturbance in each channel can only go though either a physical component in P_{d1} or a numerical component in N_{d2}, while not being able to go through both numerical and physical components in series. Many DSS problems satisfy this assumption.

Now we explore this CSF. From Fig. 3, the following relations hold:

$$z_N = N_{d2}P_{d1}d - P_{u3}z_A - N_{u2}\tilde{z}_N \tag{1}$$

$$\tilde{z}_P = P_{u1}z_A \tag{2}$$

$$z_A = P_A u \tag{3}$$

$$z_P = P_{u4}z_A \tag{4}$$

If we set the constraint variables equivalent i.e. $\tilde{z}_N = \tilde{z}_P$, then the equations (1)-(3) lead to

$$z_N = N_{d2}P_{d1}d - (P_{u3} + N_{u2}P_{u1})z_A \tag{5}$$

Hence the DSS can be expressed by

$$z_N = N_{d2}P_{d1}d - (P_{u3} + N_{u2}P_{u1})P_A u \tag{6a}$$

$$z_P = P_{u4}P_A u \tag{6b}$$

which take the same form with Fig. 1, with $G_0 = (P_{u3} + N_{u2}P_{u1})P_A$, $G_1 = N_{d2}P_{d1}$ and $G_2 = P_{u4}P_A$.

Define the DSS error as

$$e = z_N - z_P. \tag{7}$$

From (4), (5) and (7), we have

$$z_N = P_{u4}[P_{u3} + P_{u4} + N_{u2}P_{u1}]^{-1}[N_{d2}P_{d1}d + (P_{u3} + N_{u2}P_{u1})P_{u4}^{-1}e] \tag{8a}$$

$$z_P = P_{u4}[P_{u3} + P_{u4} + N_{u2}P_{u1}]^{-1}(N_{d2}P_{d1}d - e) \tag{8b}$$

which reflect the influence from DSS error e on DSS outputs z_N and z_P. Setting $e = 0$ in (8) gives

$$z_E := [I + (P_{u3} + N_{u2}P_{u1})P_{u4}^{-1}]^{-1}N_{d2}P_{d1}d \tag{9}$$

If N_{u2} is replaced by P_{u2} in (9), then the system (9) exactly represents the original system to which the DSS of Fig. 3 is emulated. Thus we call the system (9) as an *emulated* system, as shown in Fig. 4. Note that the causality requires P_{u4} and $P_{u3} + N_{u2}P_{d1} + P_{u4}$ to be invertible and proper. In this paper we do not investigate the solution of the causal problem, but refer to the method developed in [25] as a possible solution. The framework and the ones introduced later in this paper provide a convenient way to investigate this problem.

If we further define the errors between the outputs of the DSS and the emulated system as

$$e_N = z_N - z_E = [I + (P_{u3} + N_{u2}P_{u1})P_{u4}^{-1}]^{-1}(P_{u3} + N_{u2}P_{u1})P_{u4}^{-1}e \tag{10}$$

$$e_P = z_E - z_P = [I + (P_{u3} + N_{u2}P_{u1})P_{u4}^{-1}]^{-1}e \tag{11}$$

then the following relation hold:

$$e = e_N + e_P = N_{d2}P_{d1}d - (P_{u3} + N_{u2}P_{u1} + P_{u4})P_A u \tag{12}$$

which can be alternatively derived by substituting (6a) and (6b) into $e = z_N - z_P$.

From (6) and (9), we have the following proposition:

Proposition 1. *For the DSS framework shown in Fig. 3 and its emulated system shown in Fig. 4, if the DSS error $e = 0$, then $z_N = z_P = z_E$.*

Remark 1. *This proposition explicitly justifies the DSS control objective, that is, the regulation of the DSS error e guarantees that the DSS outputs, z_N and z_P, converge to the output of the emulated system, z_E.*

3 The dual DSS systems

Based on the strict separation framework as shown in Fig. 3, we introduce two dual systems: substructure & signal dual DSS system (Sub&Sig-DSS) and signal dual DSS system (Sig-DSS). These dual systems not only generalize the applicability of the proposed CSF shown in Fig. 3, but also reveal insightful information for DSS establishment. Note that the concept and motivation of the dual DSS systems are different from the dual substructured system introduced in [19, 26], which mainly focus on the relationship between the assembled and disassembled substructured systems, though some similarities are shared.

The Sub&Sig-DSS is obtained by swapping physical and numerical blocks of the original DSS, as well as (some or all of) the constraint signals and DSS outputs. Note that by swapping the physical and numerical blocks, the types (i.e., physical or numerical) of the output signals from the blocks are also changed correspondingly due to the definition of the numerical signal and physical signal (cf. Table 1).

To derive the Sig-DSS, only (some or all of) the constraint signals and DSS outputs are swapped, while the block properties (i.e., physical or numerical) remain the same. This type of dual system helps to identify which interface signals are more suitable to be DSS outputs while the others are more suitable for the constraint signals, in terms of physical realization (e.g. causality) and control issues. In [15], it is noted that: *"In many DSSs there is a degree of arbitrariness over the selection of these synchronized variables and the interaction constraints. This will be a topic for future research."* The investigation of Sig-DSS addresses this problem.

Similarly, we can also define a substructure dual DSS (Sub-DSS) by only swapping the physical and numerical blocks. We do not give more details here, and just illustrate its relation with other DSS transformations by examples in the next section.

3.1 Substructure & signal dual

Corresponding to the original DSS shown in Fig. 3, we consider its dual DSS by interchanging both the block properties and interface variables. Specifically, the physical blocks are changed to numerical blocks (i.e. $\{P_{d1}, P_{u1}, P_{u3}\}$ are replaced by $\{N_{d1}, N_{u1}, N_{u3}\}$), while the numerical blocks are changed to physical blocks, (i.e., $\{N_{d2}, N_{u2}\}$ are replaced by $\{P_{d2}, P_{u2}\}$); the constraint signals $\tilde{z}_N$ and $\tilde{z}_P$ are used as DSS outputs; the DSS outputs z_N and z_P are used as the constraint signals. Following this rule, the equations (1) - (4) are converted to:

$$z_P = P_{d2}N_{d1}d - N_{u3}z_A - P_{u2}\tilde{z}_P \tag{13}$$

$$\tilde{z}_N = N_{u1}z_A \tag{14}$$

$$\tilde{z}_P = P_{\tilde{A}}u \tag{15}$$

$$z_N = N_{u4}z_A \tag{16}$$

where $\tilde{z}_P$ is generated by another actuator $P_{\tilde{A}}$ (in many cases, this actuator can be the same physical actuator as the original one, but with different variable to be measured as the examples to be shown later); alternatively, the dynamics of $P_{\tilde{A}}$ can be estimated by $N_{u1}P_A$.

If we eliminate z_A from (13) and (16), and set the constraint variables equivalent in

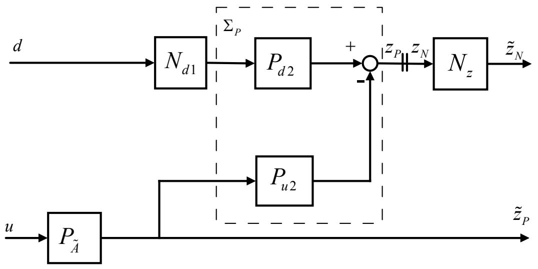


Figure 5: The block diagram of the substructure & signal dual generic DSS framework.

(13), i.e. $z := z_N = z_P$, then we have

$$z = N_{u4}(N_{u3} + N_{u4})^{-1}(P_{d2}N_{d1}d - P_{u2}\tilde{z}_P) \tag{17}$$

and

$$\tilde{z}_N = N_{u1}(N_{u3} + N_{u4})^{-1}(P_{d2}N_{d1}d - P_{u2}\tilde{z}_P) \tag{18}$$

Hence the dual DSS system can be represented as

$$\tilde{z}_N = N_{u1}(N_{u3} + N_{u4})^{-1}(P_{d2}N_{d1}d - P_{u2}P_{\tilde{A}}u) \tag{19a}$$

$$\tilde{z}_P = P_{\tilde{A}}u \tag{19b}$$

and the substructuring error is

$$\tilde{e} := \tilde{z}_N - \tilde{z}_P = N_{u1}(N_{u3} + N_{u4})^{-1}(P_{d2}N_{d1}d - P_{u2}P_{\tilde{A}}u) - P_{\tilde{A}}u$$

This system is illustrated in Fig. 5, where $N_z := N_{u1}(N_{u3} + N_{u4})^{-1}$.

Substituting $\tilde{e} = \tilde{z}_N - \tilde{z}_P$ into (18) leads to

$$\tilde{z}_N = (I + N_zP_{u2})^{-1}N_z(P_{d2}N_{d1}d + P_{u2}\tilde{e})$$

$$\tilde{z}_P = (I + N_zP_{u2})^{-1}(N_zP_{d2}N_{d1}d - \tilde{e})$$

This system is illustrated in Fig. 6, which is a counterpart to the Sub&Sig-DSS. When $\tilde{e} = 0$, this system becomes the emulated system of the Sub&Sig-DSS. The output of the emulated system is

$$\tilde{z}_E := N_{u1}(N_{u3} + N_{u4} + P_{u2}N_{u1})^{-1}P_{d2}N_{d1}d \tag{20}$$

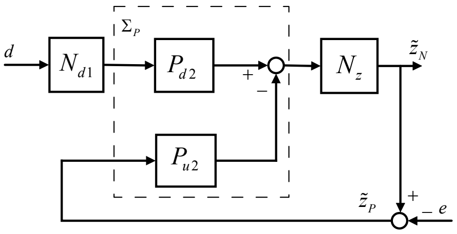


Figure 6: The block diagram of the dual generic emulated system framework.

3.2 Signal dual DSS

To derive the Sig-DSS, we need to first determine which variables can be exchanged. This requires a strict separation of the variables from the numerical and physical components. To achieve this, further partition of the signals and the blocks are necessary.

In the DSS system Fig. 3, suppose $d \in \mathbb{R}^l$, $u, z_A, z_N, z_P \in \mathbb{R}^n$, $\tilde{z}_N, \tilde{z}_P \in \mathbb{R}^m$ with $m \leq n$, and $P_{d1} \in \mathbf{RH}_\infty^{l\times l}$, $N_{d2} \in \mathbf{RH}_\infty^{n\times l}$, $P_A \in \mathbf{RH}_\infty^{n\times n}$, $P_{u1} \in \mathbf{RH}_\infty^{m\times n}$, $N_{u2} \in \mathbf{RH}_\infty^{n\times m}$, $P_{u3}, P_{u4} \in \mathbf{RH}_\infty^{n\times n}$. Here $\mathbf{RH}_\infty^{n\times m}$ denotes the space consisting of proper real rational $n \times m$ transfer function matrices with no poles on the right half plane. We can arrange P_{u3} and N_{u2} in such a form

$$N_{u2} = \begin{bmatrix} 0 \\ \bar{N}_{u2} \end{bmatrix} \qquad\qquad P_{u3} = \begin{bmatrix} \bar{P}_{u3} \\ 0 \end{bmatrix} \tag{21}$$

with $\bar{N}_{u2} \in \mathbf{RH}_\infty^{m\times m}$ and $\bar{P}_{u3} \in \mathbf{RH}_\infty^{(n-m)\times n}$. In this way, the sum of the outputs from P_{u3} and N_{u2} is a stacked vector, whose first $n-m$ elements are physical signals from P_{u3} and last m elements are numerical signals from N_{u2}. Furthermore, P_{d1} and N_{d2} can also be arranged in a similar way such that the output of the disturbance channel contains the outputs from physical components in its first $n-m$ entries and the outputs from numerical components in its last m entries. We give an example to show how to do this.

Example 1. *Consider Fig. 7, which has $l = 3$ input testing signals d_i with $i = 1, \ldots, 3$ going through 3 components G_{di} with $i = 1, \ldots, 3$. Assume G_{d1} and G_{d3} are physical components and G_{d2} is a numerical component, in which case the dimension of the constraint signal is $m = 1$. We further assume the dimension of the DSS output is $n = 4$ and the testing signals going through channels $2, 3, 4$ as $\tilde{d}_1 = 0$, $\tilde{d}_2 = G_{d1}d_1$, $\tilde{d}_3 = G_{d3}d_3$ and*

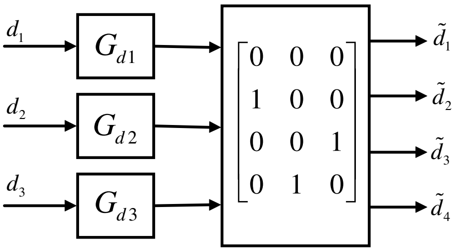


Figure 7: An example to show how testing signals to enter DSS

$\tilde{d}_4 = G_{d2}d_2$. *Then we can derive the numerical and physical blocks as*

$$N_{d2} = \begin{bmatrix}0&0&0\\1&0&0\\0&0&1\\0&1&0\end{bmatrix}\begin{bmatrix}1&0&0\\0&G_{d2}&0\\0&0&1\end{bmatrix} = \begin{bmatrix}0&0&0\\1&0&0\\0&0&1\\0&G_{d2}&0\end{bmatrix}, \qquad P_{d1} = \begin{bmatrix}G_{d1}&0&0\\0&1&0\\0&0&G_{d3}\end{bmatrix}$$

so that

$$G_d = N_{d2}P_{d1} = \begin{bmatrix}0&0&0\\G_{d1}&0&0\\0&0&G_{d3}\\0&G_{d2}&0\end{bmatrix}$$

□

Corresponding to the above arrangement of the blocks, z_N and z_P can be partitioned into two parts

$$z_N = \begin{bmatrix}z_N^{(P)}\\z_N^{(N)}\end{bmatrix} \qquad z_P = \begin{bmatrix}z_P^{(P)}\\z_P^{(N)}\end{bmatrix} \tag{22}$$

so that $z_N^{(P)} \in \mathbb{R}^{(n-m)\times 1}$ is the difference between the outputs of the physical components P_{d1} and P_{u3}, and $z_N^{(N)} \in \mathbb{R}^{m\times m}$ is the difference between the outputs of the numerical components N_{d2} and N_{u2}. The output of the actuators z_P is accordingly partitioned into

$$z_P^{(P)} = P_{u4}^{(P)}P_Au \tag{23}$$

$$z_P^{(N)} = P_{u4}^{(N)}P_Au \tag{24}$$

where

$$P_{u4}^{(P)} = \begin{bmatrix} I_{n-m} & 0_{(n-m)\times m} \end{bmatrix} P_{u4} \tag{25}$$

$$P_{u4}^{(N)} = \begin{bmatrix} 0_{m\times(n-m)} & I_m \end{bmatrix} P_{u4} \tag{26}$$

Now we want to use $z_N^{(N)}$ and $z_P^{(N)}$ as the constraint variable, and the DSS outputs are constructed as

$$\hat{z}_N = \begin{bmatrix} z_N^{(P)} \\ \tilde{z}_N \end{bmatrix} \qquad\qquad \hat{z}_P = \begin{bmatrix} z_P^{(P)} \\ \tilde{z}_P \end{bmatrix} \tag{27}$$

respectively. From (3) and (2), we have the synchronization signal $\tilde{z}_P$

$$\tilde{z}_P = P_{u1} P_A u = P_{\tilde{A}} u \tag{28}$$

where

$$P_{\tilde{A}} := P_{u1} P_A \tag{29}$$

Thus, $\hat{z}_P$ is determined by

$$\hat{z}_P := \begin{bmatrix} z_P^{(P)} \\ \tilde{z}_P \end{bmatrix} = \begin{bmatrix} P_{u4}^{(P)} \\ P_{u1} \end{bmatrix} P_A u \tag{30}$$

Suppose N_{d2} is partitioned in accordance with the partition of z_N, such that

$$N_{d2} = \begin{bmatrix} N_{d2}^{(P)} \\ N_{d2}^{(N)} \end{bmatrix}$$

with $N_{d2}^{(P)} \in \mathbf{RH}_\infty^{(n-m)\times l}$ and $N_{d2}^{(N)} \in \mathbf{RH}_\infty^{m\times l}$. Then from the partitions of P_{u3} and N_{u2} in (21), (1) can be written as

$$z_N = \begin{bmatrix} z_N^{(P)} \\ z_N^{(N)} \end{bmatrix} = \begin{bmatrix} N_{d2}^{(P)} \\ N_{d2}^{(N)} \end{bmatrix} P_{d1} d - \begin{bmatrix} \bar{P}_{u3} \\ 0 \end{bmatrix} z_A - \begin{bmatrix} 0 \\ \bar{N}_{u2} \end{bmatrix} \tilde{z}_N \tag{31}$$

Suppose the inverse of $\bar{N}_{u2}$ exists. Then pre-multiplying (31) by

$$\begin{bmatrix} 0_{m\times(n-m)} & \bar{N}_{u2}^{-1} \end{bmatrix}$$

leads to

$$\tilde{z}_N = \bar{N}_{u2}^{-1}(N_{d2}^{(N)} P_{d1} d - z_N^{(N)}) \tag{32}$$

Combining the above equation with

$$z_N^{(P)} = N_{d2}^{(P)} P_{d1} d - \bar{P}_{u3} z_A \tag{33}$$

leads to

$$\begin{bmatrix} z_N^{(P)} \\ \tilde{z}_N \end{bmatrix} = \begin{bmatrix} I_{n-m} & 0 \\ 0 & \bar{N}_{u2}^{-1} \end{bmatrix} N_{d2} P_{d1} d - P_{u3} z_A - \begin{bmatrix} 0_{(n-m)} & 0 \\ 0 & \bar{N}_{u2}^{-1} \end{bmatrix} z_N \tag{34}$$

Since the constraint variables satisfy $z_N^{(N)} = z_P^{(N)} = P_{u4}^{(N)} P_A u$, the DSS output can be represented as

$$\hat{z}_N := \begin{bmatrix} z_N^{(P)} \\ \tilde{z}_N \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & \bar{N}_{u2}^{-1} \end{bmatrix} N_{d2} P_{d1} d - \left(P_{u3} + \begin{bmatrix} 0_{(n-m)\times(n-m)} & 0 \\ 0 & \bar{N}_{u2}^{-1} \end{bmatrix} P_{u4} \right) P_A u \tag{35}$$

By simplifying (35) and combining it with (30), we have the Sig-DSS

$$\hat{z}_N = \hat{N}_z \left(N_{d2} P_{d1} d - \begin{bmatrix} \bar{P}_{u3} \\ P_{u4}^{(N)} \end{bmatrix} P_A u \right) \tag{36a}$$

$$\hat{z}_P = \begin{bmatrix} z_P^{(P)} \\ \tilde{z}_P \end{bmatrix} = \begin{bmatrix} P_{u4}^{(P)} \\ P_{u1} \end{bmatrix} P_A u \tag{36b}$$

with $\hat{N}_z := \begin{bmatrix} I & 0 \\ 0 & \bar{N}_{u2}^{-1} \end{bmatrix}$. The resulting Sig-DSS is illustrated in Fig. 8.

When $n = m$, i.e. P_{u3} does not exist, $\bar{N}_{u2} = N_{u2}$, $N_{d2}^{(N)} = N_{d2}$, $\hat{N}_z = N_{u2}^{-1}$ and $P_A^{(N)} = P_A$, the Sig-DSS framework (36) is reduced to

$$\begin{aligned} \hat{z}_N &= \tilde{z}_N = N_{u2}^{-1}(N_{d2} P_{d1} d - P_{u4} P_A u) \\ \hat{z}_P &= \tilde{z}_P = P_{\tilde{A}} u = P_{u1} P_A u \end{aligned} \tag{37}$$

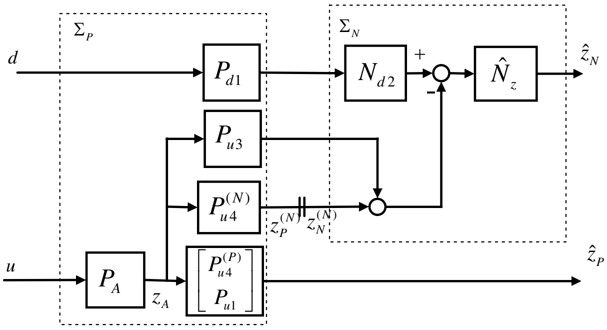


Figure 8: The block diagram of the signal dual DSS framework

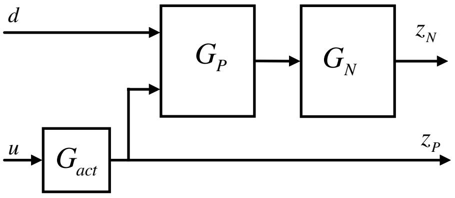


Figure 9: The block diagram of the generic DSS framework without considering substructure internal relations.

3.3 The generic DSS framework with a strict separation of components

In summary, the original DSS system and its two dual systems can be represented by one generic form, as shown in Fig. 9, without considering internal relations. (Note that in the Sub&Sig-DSS, N_{d1} and P_{d2} are interchangeable, but with different representations.) In this generic framework the actuators are separated out from the physical substructure. This generic framework without considering the internal relations may be convenient for designing nominal DSS controllers.

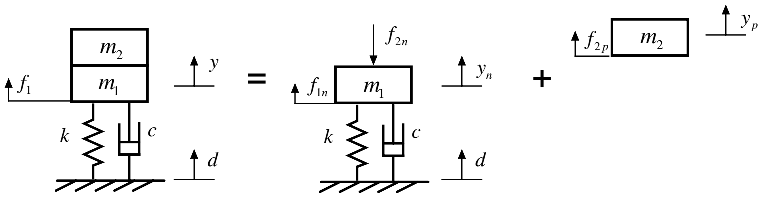


Figure 10: Substructuring of split mass - schematic plot

4 Mass split example

In this section, we consider a mass(m)-damper(c)-spring(k) system as shown in Fig. 10. This example is extensively studied in the DSS literature (e.g. [24, 25]). We show how to establish DSS using the frameworks proposed in this paper.

In Fig. 10, d is the testing signal (displacement); y is the displacement of the mass; f_1 is the force acting on the mass from the spring and the damper. The dynamics equation for the emulated system in Fig. 10 is

$$m\ddot{y} = k(d - y) + c(\dot{d} - \dot{y}) \tag{38}$$

and its Laplace transform representation is

$$y(s) = \frac{cs + k}{ms^2 + cs + k} d(s) \tag{39}$$

Suppose the mass m is split into two parts – the top mass is m_2 and the bottom mass m_1. f_2 is the interaction force between m_1 and m_2. The system are supposed to have two substructures: the top substructure consisting of mass m_2 and the bottom substructure consisting of m_1, k and c. We can consider each substructure either as physical or numerical, and also consider the interface signal either as force or displacement. Thus we can derive 4 different DSS formulations. We adopt new notations to represent them concisely. For example, we use '$P - N : y$' to denote the DSS when the top mass is physical, the bottom remaining parts are numerical, and the DSS outputs are displacement signals.

4.1 Case $P - N : y$

Suppose the numerical substructure contains the bottom mass m_1, spring k and damper c; the physical substructure contains the top mass m_2. Then we have the following relations:

$$m_1\ddot{y}_n = f_{1n} - f_{2n} \tag{40}$$

$$f_{1n} = k(d - y_n) + c(\dot{d} - \dot{y}_n) \tag{41}$$

$$m_2\ddot{y}_p = f_{2p} \tag{42}$$

from which we have

$$y_n = \frac{cs+k}{m_1 s^2 + cs + k} d - \frac{1}{m_1 s^2 + cs + k} f_{2n} \tag{43}$$

$$y_p = \frac{1}{m_2 s^2} f_{2p} \tag{44}$$

Suppose the constraint signals are $f_{2n} = f_{2p}$ and the force f_{2p} is generated by a force actuator such that $f_{2p} = P_{Af} u$. Then the DSS can be represented by equations

$$y_n = N_{d2} d - N_{u2} P_{Af} u \tag{45}$$

$$y_p = P_{u4} P_{Af} u \tag{46}$$

with

$$N_{d2} = \frac{cs+k}{m_1 s^2 + cs + k} \qquad N_{u2} = \frac{1}{m_1 s^2 + cs + k} \qquad P_{u4} = \frac{1}{m_2 s^2}$$

This is the DSS framework shown in Fig. 3, with $P_{d1} = I$, $P_{u3} = 0$, $P_{u1} = I$. Using (9), we can derive the dynamics of the emulated system as follows:

$$y_E = P_{u4}(P_{u4} + N_{u2})^{-1} N_{d2} d = \frac{cs+k}{(m_1 + m_2)s^2 + cs + k} \tag{47}$$

which is consistent with (39).

In the following, we can directly derive the other three cases by employing the DSS frameworks and transformations proposed in this paper.

4.2 Case $P - N : f$

Consider the case that the top mass is physical, the remaining parts are numerical, while the constraint signals are $y_p = y_n$ and the DSS output signals are f_{2n} and f_{2p}. This is the Sig-DSS of the case $P - N : y$. Using the relations in (37), we have

$$f_{2n} = N_{u2}^{-1}(N_{d2} d - y_n) = N_{u2}^{-1}(N_{d2} d - y_p) = N_{u2}^{-1}(N_{d2} d - P_{u4} P_{Af} u) \tag{48}$$

$$f_{2p} = P_{Af} u \tag{49}$$

In this case, the block N_{u2}^{-1} is not proper and noncausal.

4.3 Case $N - P : f$

Consider the case that the top mass is numerical, the remaining parts are physical, while the constraint signals are $y_p = y_n$ and the DSS output signals are f_{2n} and f_{2p}. This is the Sub&Sig-DSS of the case $P - N : y$. From the relations in (19a) with $N_z = N_{u4}^{-1}$ and $P_{\tilde{A}} = P_{Af}$, we have

$$f_{2n} = N_{u4}^{-1}(P_{d2}d - P_{u2}P_{Af}u) \tag{50}$$

$$f_{2p} = P_{Af}u \tag{51}$$

Moreover, from (20) we can derive f of the emulated system as

$$f_E = (N_{u4} + P_{u2})^{-1}P_{d2}d = \frac{m_2 s^2(cs+k)}{ms^2 + cs + k} \tag{52}$$

4.4 Case $N - P : y$

Consider the case that the top mass is numerical, the remaining parts are physical, while the constraint signals are $f_{2p} = f_{2n}$ and the DSS output signals y_n and y_p are swapped due to the change of blocks.

In this case, the DSS is derived by changing N_{d2}, N_{u2} and P_{u4} to P_{d2}, P_{u2} and N_{u4} respectively and swap y_n and y_p in equations (45) and (46). f_{2p} is generated by an actuator $f_{2p} = P_{Af}u$, so that the DSS is described by

$$y_p = P_{d2}d - P_{u2}P_{Af}u \tag{53}$$

$$y_n = N_{u4}P_{Af}u \tag{54}$$

This DSS is a Sub&Sig-DSS of the case $P - N : f$ and it can also be viewed as a Sub-DSS of the case $P - N : y$.

Remark 2. *The relations of these four DSS are shown in Fig. 11. From the above four cases, we can see that the two cases with force as constraint signal ($P - N : y$ and $N - P : y$) are causal DSSs, while the other two cases with displacement as constraint signal ($P - N : f$ and $N - P : f$) are noncausal DSSs. For the noncausal DSSs, some methods are available to accommodate this noncausal problem (cf. [25]). Moreover, in all the four DSSs, the physical components and numerical components are strictly separated, that is, there are no blocks containing mixed numerical and physical components. This feature*

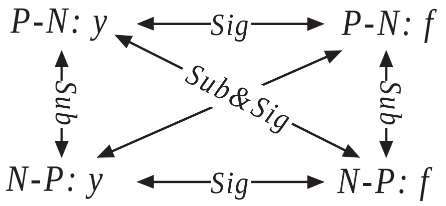
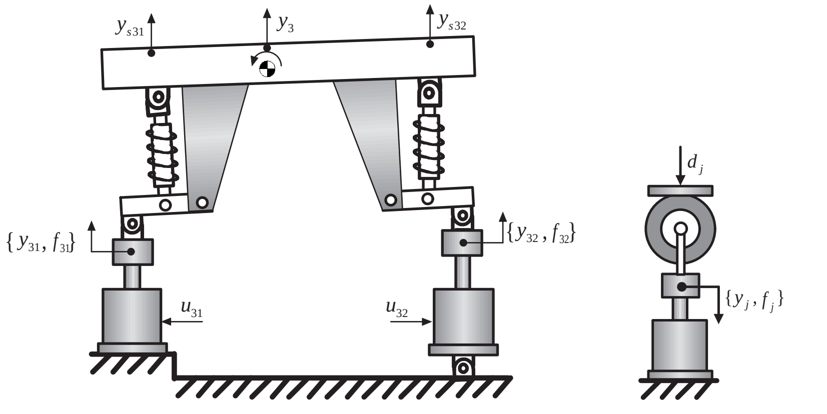


Figure 11: The relations of the four DSS formulations of the mass split example



(a) Vehicle body with two suspension struts

(b) Front (j=1) and rear wheels (j=2)

Figure 12: Schematic representation of the QM rig.

makes it easy to identify and quantify the uncertainties and disturbances associated with the physical components only, and thus significantly facilitates the robust stability analysis and robust controller design for DSS testings.

5 Examples from the quasi-motorcycle (QM) DSS

In this section, we use more complicated examples to show that a variety of DSS problems can be cast into the generic framework and its dual DSS framework proposed in Sections 2 and 3. These examples are extracted from the QM DSS testing rig developed at the University of Bristol as shown schematically in Fig. 12, which consists of three mechanically separated substructures $\{\Sigma_1, \Sigma_2, \Sigma_3\}$. Σ_1 has one vertically mounted 2 DOF rigid body with two suspension struts substructure and two swing arms; Σ_2 and Σ_3 are two horizontally mounted 1 DOF wheel/tyre substructures. 4 hydraulic actuators are connected with the two swing arms and the two wheel hubs respectively. Each of the substructure can serve as either numerical or physical substructure. For a detailed introduction of this system, see [15, 27, 17, 21], etc.; here, we only present the main dynamics equations of the system:

$$y_1 = G_{yd1}d_1 - G_{yf1}f_1 \tag{55}$$

$$y_2 = G_{yd2}d_2 - G_{yf2}f_2 \tag{56}$$

$$f_{31} = P_2 s^2 y_{s31} + P_3 s^2 y_{s32} \tag{57}$$

$$f_{32} = P_3 s^2 y_{s31} + P_1 s^2 y_{s32} \tag{58}$$

$$y_{s31} = G_{31}y_{31} \tag{59}$$

$$y_{s32} = G_{32}y_{32} \tag{60}$$

with P_1, P_2 and P_3 as some fixed parameters and

$$\begin{aligned} G_{yd1} &= \frac{c_1 s + k_1}{m_1 s^2 + c_1 s + k_1} & G_{yd2} &= \frac{c_2 s + k_2}{m_2 s^2 + c_2 s + k_2} \\ G_{yf1} &= \frac{1}{m_1 s^2 + c_1 s + k_1} & G_{yf2} &= \frac{1}{m_2 s^2 + c_2 s + k_2} \\ G_{31} &= \frac{c_{31} s + k_{31}}{m_{31} s^2 + c_{31} s + k_{31}} & G_{32} &= \frac{c_{32} s + k_{32}}{m_{32} s^2 + c_{32} s + k_{32}} \end{aligned}$$

5.1 The QM DSS

In this section, we consider the DSS examples generated from the whole QM system. These DSS examples are all multivariable systems. Recall that the whole QM system contains three substructures denoted by $\{\Sigma_1, \Sigma_2, \Sigma_3\}$, corresponding to front wheel, rear wheel and vehicle body together with two suspension struts plus the two swing arms. Different combinations of the numerical and physical substructures lead to 5 types of DSSs. By following the terminologies adopted in [15], we use $\{\Sigma_{N1}, \Sigma_{N2}, \Sigma_{P3}\}$ to denote the case when Σ_1 and Σ_2 are numerical substructures, and Σ_3 is a physical substructure. Since there is only one physical mode in this DSS, we call it as a single interaction mode DSS, concisely denoted by SiM. Another single mode DSS is represented by $\{\Sigma_{N1}, \Sigma_{P2}, \Sigma_{N3}\}$. The subindex N (or P) denotes the associated substructure is numerical (or physical). To distinguish these two single mode DSS, we denote the first one as SiM 1 and the second one as SiM 2. In a similar way, we can derive multi-mode (MuM) $\{\Sigma_{P1}, \Sigma_{P2}, \Sigma_{N3}\}$ and mixed mode (MiM) $\{\Sigma_{N1}, \Sigma_{P2}, \Sigma_{P3}\}$ and all physical modes (PhM) $\{\Sigma_{P1}, \Sigma_{P2}, \Sigma_{P3}\}$. Among these, SiM 1 and MuM are Sub& Sig-DSS to each other; SiM 2 and MiM are Sig&Sub-DSS to each other. We do not consider the DSS consisting of all numerical substructures.

The interface signals are the displacements and the forces at the attachment points of the wheel hubs. The ends of the swing arms are represented by $\{y_{31}, y_{32}\}$, $\{y_1, y_2\}$, $\{f_{31}, f_{32}\}$ and $\{f_1, f_2\}$.

It will be shown that SiM1 and MuM are Sub&Sig-DSS to each other, and SiM2 and MiM are Sub&Sig-DSS to each other. Using these relations between the dual systems, we can straightforwardly convert one DSS to its dual system.

5.1.1 SiM 1

SiM 1 has the DSS structure of $\{\Sigma_{N1}, \Sigma_{N2}, \Sigma_{P3}\}$. The interface signals are $\{y_{n1}, y_{p31}\}$, $\{f_{n1}, f_{p31}\}$, $\{y_{n2}, y_{p32}\}$ and $\{f_{n2}, f_{p32}\}$. Substituting these notations into (55) - (60) leads to

$$y_{n1} = G_{yd1}d_1 - G_{yf1}f_{n1} \tag{61}$$

$$y_{n2} = G_{yd2}d_2 - G_{yf2}f_{n2} \tag{62}$$

$$f_{p31} = P_2 s^2 G_{31} y_{p31} + P_3 s^2 G_{32} y_{p32} \tag{63}$$

$$f_{p32} = P_3 s^2 G_{31} y_{p31} + P_1 s^2 G_{32} y_{p32} \tag{64}$$

$$y_{p31} = G_{a31}u_{31} \tag{65}$$

$$y_{p32} = G_{a32}u_{32} \tag{66}$$

where G_{a31} and G_{a32} are actuators. By choosing the displacements as the synchronizing signals while the forces as the constraint signals, i.e. $f_{p31} = f_{n1}$ and $f_{p32} = f_{n2}$, we can derive

$$y_{n1} = G_{yd1}d_1 - G_{yf1}(P_2 s^2 G_{31} G_{a31} u_{31} + P_3 s^2 G_{32} G_{a31} u_{31}) \tag{67}$$

$$y_{n2} = G_{yd2}d_2 - G_{yf2}(P_3 s^2 G_{31} G_{a31} u_{31} + P_1 s^2 G_{32} G_{a32} u_{32}) \tag{68}$$

Note that the physical parameters are only contained in G_{31} and G_{32}. From (65) - (68), the DSS framework of SiM 1 is

$$z_N = N_{d2}d - N_{u2}P_{u1}P_A u \tag{69}$$

$$z_P = P_A u \tag{70}$$

where $z_N = [y_{n1}, y_{n2}]^T$, $z_P = [y_{p31}, y_{p32}]^T$, $d = [d_1, d_2]^T$, $u = [u_{31}, u_{32}]$, $N_{d2} = \mathrm{diag}(G_{yd1}, G_{yd2})$, $N_{u2} = \mathrm{diag}(G_{yf1}, G_{yf2})$, $P_A = \mathrm{diag}(G_{act1}, G_{act2})$, and

$$P_{u1} = \begin{bmatrix} P_2 s^2 G_{31} & P_3 s^2 G_{32} \\ P_3 s^2 G_{31} & P_1 s^2 G_{32} \end{bmatrix}.$$

The interface constraint signals are $\tilde{z}_p = [f_{p31}, f_{p32}]$ and $\tilde{z}_N = [f_{n1}, f_{n2}]$. The block diagram of SiM1 can be represented by Fig. 3 with $P_{d1} = I$, $P_{u3} = 0$ and $P_{u4} = I$.

Remark 3. *For SiM 1, although the numerical model of the physical block P_{u1} is non-causal, the numerical model for $P_{u1}P_A$ is causal. This indicates that when designing a robust controller or conducting robustness analysis, the uncertainties associated with P_A or $P_{u1}P_A$ can be taken into account directly; however for the uncertainties associated with P_{u1} alone, it is necessary to make the model of P_{u1} causal first, e.g. by adding a filter or by using system identification to derive this model directly.*

5.1.2 MuM – Substructure & Signal Dual DSS of SiM1

The Substructure & Signal Dual DSS of SiM 1 takes the form of $\{\Sigma_{P1}, \Sigma_{P2}, \Sigma_{N3}\}$, which is actually MuM. The interface signals are $\{y_{p1}, y_{n31}\}$, $\{f_{p1}, f_{n31}\}$, $\{y_{p2}, y_{n32}\}$ and $\{f_{p2}, f_{n32}\}$. A straightforward application of the substructure & signal dual relationship leads to the DSS framework for MuM

$$\tilde{z}_N = N_{u1}(P_{d2}d - P_{u2}P_{\tilde{A}}u) \tag{71}$$

$$\tilde{z}_P = P_{\tilde{A}}u \tag{72}$$

with $\tilde{z}_N = [f_{n31}, f_{n32}]^T$, $\tilde{z}_P = [f_{p1}, f_{p2}]^T$. The expressions of N_{u1}, P_{u2} and P_{d2} are exactly equivalent to those transfer functions in SiM1 with the same subindices; $\tilde{z}_P$ are generated by two actuators: $f_{p1} = G_{Af1}u_1$, $f_{p2} = G_{Af2}u_2$ so that $P_{\tilde{A}} = \mathrm{diag}(G_{Af1}, G_{Af2})$. The DSS constraint signals are $z_P = [y_{p1}, y_{p2}]$, $z_N = [y_{n31}, y_{n32}]$. The block diagram of MuM can be represented by Fig. 5 with $N_{d1} = I$, $N_{u3} = 0$ and $N_{u4} = I$.

Remark 4. *For MuM, all the blocks are causal; this means the uncertainties associated with P_{d2}, P_u2 and $P_{\tilde{A}}$ can be taken into account directly for robustness analysis and robust controller design.*

5.1.3 MiM

The MiM takes the DSS substructure of $\{\Sigma_{P1}, \Sigma_{N2}, \Sigma_{P3}\}$. The interface signal between the physical body plus suspension struts Σ_3 and the front physical wheel Σ_1 are the forces $\{f_{p31}, f_{p1}\}$ and displacements $\{y_{p31}, y_{p1}\}$; and the interface signal between Σ_3 and Σ_2 are the forces $\{f_{p32}, f_{n2}\}$ and displacements $\{y_{p32}, y_{n2}\}$. We choose $\{f_{p32}, f_{n2}\}$ as the constraint signal so that $f_{p32} = f_{n2}$, and synchronize the signals $\{f_{p31}, f_{p1}\}$, $\{y_{p31}, y_{p1}\}$ and $\{y_{p32}, y_{n2}\}$. Note that in the physical interface between Σ_{P3} and Σ_{P1}, we can synchronize both pairs of displacements and forces because two actuators are employed to provide displacement and force signal respectively. Hence the DSS outputs are $z_N = \begin{bmatrix} f_{p31} & y_{p1} & y_{n2} \end{bmatrix}^T$ and $z_P = \begin{bmatrix} f_{p1} & y_{p31} & y_{p32} \end{bmatrix}^T$. From (55) - (60), z_N can be derived by

$$\begin{aligned} f_{p31} &= P_2 s^2 G_{31} G_{a31} u_{31} + P_3 s^2 G_{32} G_{a32} u_{32} \\ y_{p1} &= G_{yd1} d_1 - G_{yf1} G_{a1} u_1 \\ y_{n2} &= G_{yd2} d_2 - P_3 s^2 G_{yf2} G_{31} G_{a31} u_{31} - P_1 s^2 G_{yf2} G_{32} G_{a32} u_{32} \end{aligned}$$

where the physical signals contained in z_2 are generated by three actuators $f_{p1} = G_{a1} u_1$, $y_{p31} = G_{a31} u_{31}$ and $y_{p32} = G_{a32} u_{32}$.

Hence the DSS framework of MiM takes the form of (6) with $P_{u4} = I$ and

$$P_{u1} = \begin{bmatrix} 0 & P_3 G_{31} s^2 & P_1 G_{32} s^2 \end{bmatrix} \qquad P_{u3} = \begin{bmatrix} 0 & -P_2 G_{31} s^2 & -P_3 G_{32} s^2 \\ G_{yf1} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{73}$$

$$N_{u2} = \begin{bmatrix} 0 \\ 0 \\ G_{yf2} \end{bmatrix} \qquad N_{d2} = \begin{bmatrix} 0 & 0 \\ I & 0 \\ 0 & G_{yd2} \end{bmatrix} \tag{74}$$

$$P_{d1} = \text{diag}(G_{yd1}, I) \qquad P_A = \text{diag}(G_{a1}, G_{a31}, G_{a32}) \tag{75}$$

The block diagram of MiM can be represented by Fig. 3 with $P_{u4} = I$, and the equations are

$$z_N = N_{d2} P_{d1} d - (P_{u3} + N_{u2} P_{u1}) P_A u \tag{76}$$

$$z_P = P_A u \tag{77}$$

Remark 5. *For MiM, both P_{u1} and P_{u3} contains noncausal elements; however $P_{u1}P_A$ and $P_{u3}P_A$; the same remark hold with Remark 3.*

5.1.4 SiM2

SiM2 has the structure of $\{\Sigma_{N1}, \Sigma_{P2}, \Sigma_{N3}\}$. The interface signals between Σ_{P2} and Σ_{N3} are $\{y_{p2}, y_{n32}\}$ and $\{f_{p2}, f_{n32}\}$; the interface signals between Σ_{N1} and Σ_{N3} are $\{y_{n1}, y_{n31}\}$ and $\{f_{n1}, f_{n31}\}$, with the constraints $y_{n1} = y_{n31}$ and $f_{n1} = f_{n31}$. We choose $\{y_{p2}, y_{n32}\}$ as the constraint signal so that $\{y_{p2} = y_{n32}\}$ and synchronize $\{f_{p2}, f_{n32}\}$. It is obvious that SiM2 is a Sub&Sig-DSS of MiM. Hence SiM 2 can be derived directly from MiM using equations (19a) and (19b).

We first notice that SiM 2 only contains one numerical substructure, i.e., Σ_{P2}. Further considering that displacement signal is used as the synchronization variable in MiM, an actuator is needed in SiM 2 to provide the force signal $f_{p2} = \tilde{G}_{Af2}u$.

Since the dynamics expressions of N_{u1}, P_{u2}, N_{u3}, N_{d1} and P_{d2} equal to those of MiM with the same subindices respectively, we can derive

$$N_z = N_{u1}(I + N_{u3})^{-1} = \begin{bmatrix} N_{z1} & N_{z2} & N_{z3} \end{bmatrix}$$

with

$$\begin{aligned}
N_{z1} &= -P_3 G_{31} s^2 G_{yf1}(1 + P_{22} G_{31} s^2 G_{yf1})^{-1} \\
N_{z2} &= P_3 G_{31} s^2 (1 + P_{22} G_{31} s^2 G_{yf1})^{-1} \\
N_{z3} &= -(P_3 G_{31} s^2)(P_3 G_{32} s^2) G_{yf1}(1 + P_{22} G_{31} s^2 G_{yf1})^{-1} + P_1 G_{31} s^2
\end{aligned}$$

Hence, SiM 2 can be expressed as

$$\begin{aligned}
\tilde{z}_N &= N_z(P_{d2} N_{d1} d - P_{u2} P_{\tilde{A}} u) \\
\tilde{z}_P &= P_{\tilde{A}} u
\end{aligned}$$

with $P_{\tilde{A}} = \tilde{G}_{Af2}$. Its block diagram can be represented by Fig. 5.

Remark 6. *Form MiM, the only noncausal block is the numerical block N_z, whose noncausal problem can be easily coped with by adding an extra filter.*

5.1.5 PhM

PhM has three physical substructures as $\{\Sigma_{P1}, \Sigma_{P2}, \Sigma_{P3}\}$. Since there are no numerical components, PhM is a special case of the DSS shown in Fig. 3. This type of DSS has practical benefits when the physical components under testing are remote to each other and need to be syncronized.

The DSS outputs and the synchronization variables generated by the actuators are: $z_{p1} = \begin{bmatrix} f_{p31} & f_{p32} & y_{p1} & y_{n2} \end{bmatrix}^T$, $z_{p2} = \begin{bmatrix} f_{p1} & f_{p1} & y_{p31} & y_{p32} \end{bmatrix}^T$. We have the relations

$$\begin{aligned} f_{p31} &= P_2 s^2 G_{31} G_{a31} u_{31} + P_3 s^2 G_{32} G_{a32} u_{32} \\ f_{p32} &= P_3 s^2 G_{31} G_{a31} u_{31} + P_1 s^2 G_{32} G_{a32} u_{32} \\ y_{p1} &= y_{p1d} - y_{p1f} = G_{yd1} d_1 - G_{yf1} G_{a1} u_1 \\ y_{p2} &= y_{p2d} - y_{p2f} = G_{yd2} d_2 - G_{yf2} G_{a2} u_2 \end{aligned}$$

and $f_{p1} = G_{a1} u_1$, $y_{p31} = G_{a31} u_{31}$, $f_{p2} = G_{a2} u_2$ and $y_{p32} = G_{a32} u_{32}$. Hence, the PhM can be expressed as

$$\begin{aligned} z_{p1} &= P_{d1} d - P_{u3} P_A u \\ z_{p2} &= P_A u \end{aligned}$$

with $P_{d2} = I$, $P_{u1} = 0$, $P_{u2} = 0$, $P_A = \text{diag}(G_{a1}, G_{a2}, G_{a31}, G_{a32})$ and

$$P_{d1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ G_{yd1} & 0 \\ 0 & G_{yd2} \end{bmatrix} \qquad P_{u3} = \begin{bmatrix} 0 & 0 & -P_2 s^2 G_{31} & -P_3 s^2 G_{32} \\ 0 & 0 & -P_3 s^2 G_{31} & -P_1 s^2 G_{32} \\ G_{yf1} & 0 & 0 & 0 \\ 0 & G_{yf2} & 0 & 0 \end{bmatrix}$$

The block diagram of PhM can be represented by Fig. 3 with $N_{d2} = I$, $P_{u1} = I$, $n_{u2} = I$ and $P_{u4} = I$.

Remark 7. *Form PhM, only the physical block P_{u3} contains some noncausal entries. The same remark with Remark 3 holds for PhM regarding the causality and the uncertainty issues.*

6 Conclusion

We have proposed a DSS framework with a complete separation of physical and numerical substructures. This framework can be transformed into other forms by using the internal relations of the substructures and signals. This framework and its transformed ones unify most exiting DSS formulations. The spring-mass-damper with mass split system and the quasi-motorcycle system are used as concrete examples to demonstrate the DSS establishment and the transformations using these frameworks. These frameworks help gain insights of the DSS, and provide a convenient way to investigate many essential problems associated with DSS, e.g. the causality problem. Based on these frameworks, further researches such as robust stability and DSS performance validation can be conducted.

References

[1] T. Nakashima, H. Kato, and E. Takaoka. Development of real-time pseudo dynamic testing. *Earthquake Engineering and Structural Dynamics*, 21:79–92, 1992.

[2] Sung-Kyung Lee, Eun Churn Park, Kyung-Won Min, and Ji-Hun Park. Real-time substructuring technique for the shaking table test of upper substructures. *Engineering Structures*, 29:2219–2232, 2007.

[3] D. P. Stoten, J. Tu, G. Li, and Koichi Koganezawa. Robotic subsystem testing using an adaptively controlled dynamically substructured framework. In *9th IFAC Symposium on Robot Control*, Gifu, Japan, 2009.

[4] A R Plummer. Model-in-the-loop testing. *Proc. IMechE Part I: Journal of Systems and Control Engineering*, 220:183–199, 2006.

[5] Daniel D. Nana and Stephen Huzar. Synthesis shuttle vehicle damping using substructure test results. *Journal of Spacecraft and Rockets*, 10(12):790–797, 1973.

[6] Pui-Shum B. Shing and Stephen A. Mahin. Cumulative experimental errors in pseudodynamic tests. *Earthquake Engineering & Structural Dynamics*, 15(4):409–424, 1987.

[7] M. S. Williams and A. Blakeborough. Laboratory testing of substurctures under

dynamic loads: an introductory review. *Phil. Trans. R. Soc. Lond. A,*, 359:1651–1669, 2001.

[8] D. P. Stoten and R. A. Hyde. Adaptive control of dynamically substructured systems: the single-input single-output case. *Proc. IMechE Part I: Systems and Control Engineering*, 220:63–79, 2006.

[9] C. P. Lamarche, A. Bonelli, O. S. Bursi, and R. Tremblay. A rosenbrock-w method for real-time dynamic substructuring and pseudo-dynamic testing. *Earthquake Engineering & Structural Dynamics*, 38(9):1071–1092, 2009.

[10] O. S. Bursi, L. He, A. Bonelli, and P. Pegon. Novel generalized-α methods for interfield parallel integration of heterogeneous structural dynamic systems. *J. Comput. Appl. Math.*, 234(7):2250–2258, August 2010.

[11] Toshihiko Horiuchi and Takao Konno. A new method for compensating actuator delay in real time hybrid experiments. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 359(1786):1893–1909, 2001.

[12] A. P. Darby, M. S. Williams, and A. Blakeborough. Stability and delay compensation for real-time substructure testing. *Journal of Engineering Mechanics*, 128(12):1276–1284, 2002.

[13] M. I. Wallace, D.J Wagg, and S.A Neild. An adaptive polynomial based forward prediction algorithm for multi-actuator real-time dynamic substructuring. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, 461(2064):3807–3826, 2005.

[14] P. A. Bonnet, A. Blakeborough, M. S. Williams, and C. A. Taylor. *Hybrid Simulation: Theory, Implementation and Applications*, chapter Real time substructuring in the UK, pages 145–155. Taylor & Francis Group, London, UK, 2008.

[15] D. P. Stoten, J. Tu, and G. Li. Synthesis and control of generalised dynamically substructured systems. *Proc. IMechE Part I: Systems and Control Engineering*, 223:371–392, 2009.

[16] J. Y. Tu, D. P. Stoten, G. Li, and R. A. Hyde. A state-space approach for the control of dynamically substructured systems. In *3rd IEEE Multi-conference on Systems and Control*, St Petersburg, Russia, 2009.

[17] G. Li, G. Herrmann, D. P. Stoten, J. Tu, and M. C. Turner. Application of robust antiwindup techniques to dynamically substructured systems. *IEEE/ASME Transactions on Mechatronics*, 18(1):263–272, 2013.

[18] Jia-Ying Tu and Jheng-Ying Jiang. Substructurability and exact synchronisation analysis. *Structural Control and Health Monitoring*. published online.

[19] J. Y. Tu. Development of numerical-substructure-based and output-based substructuring controllers. *Structural Control and Health Monitoring*, 20(6).

[20] G. Li, J. Na, D. P. Stoten, and X. Ren. Adaptive feedforward control for dynamically substructured systems based on neural network compensation. In *the 18th IFAC World Congress*, pages 944–949, Milan, Italy, 2011.

[21] G. Li, J. Na, D.P. Stoten, and X. Ren. Adaptive neural network feedforward control for dynamically substructured systems. *IEEE Transactions on Control Systems Technology*, 2013. published online.

[22] G. Li, D. P. Stoten, and J. Tu. Model predictive control of dynamically substructured systems with application to a servohydraulically-actuated mechanical plant. *IET Control Theory Appl.*, 4:253–264, 2010.

[23] G. Li, G. Herrmann, D. P. Stoten, J. Tu, and M. C. Turner. A novel disturbance rejection anti-windup framework. *International Journal of Control*, 84(1):123–137, 2011.

[24] S. A. Neild, D. P. Stoten, D. Drury, and D. J. Wagg. Control issues relating to real-time substructuring experiments using a shaking table. *Earthquake Engineering & Structural Dynamics*, 34(9):1171–1192, 2005.

[25] P. J. Gawthrop, S. A. Neild, A. Gonzalez-Buelga, and D. J. Wagg. Causality in real-time dynamic substructure testing. *Mechatronics*, 19(7):1105 – 1115, 2009.

[26] Rixen D.J. Voormeeren, S.N. A dual approach to substructure decoupling techniques. In *28th IMAC, A Conference on Structural Dynamics*, Jacksonville, FL, US.

[27] David P Stoten, Jia-Ying Tu, and Guang Li. Adaptive control of generalised dynamically substructured systems. *17th IFAC World Congress, Seoul, Korea*, pages 14090–14095, 2008.