Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

A generic dynamically substructured system framework and its dual counterparts

Guang Li *

* *The School of Engineering and Materials Science, Queen Mary, University of London, Mile End Road, London, E1 4NS.*
(guangli78@gmail.com)

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 sophisticated treatments of DSS problems, such as DSS establishment, causality, as well as uncertainties and measurement noise incorporation in robust control. The proposed framework and its variations unify many DSS problems.

1. INTRODUCTION

Dynamically substructured system (DSS) is a testing method used in the dynamics testing community. The applications of the DSS concept can be found in areas such as civil engineering [Nakashima et al., 1992], robotics [Stoten et al., 2009b], automotive [Plummer, 2006] and aerospace [Nana and Huzar, 1973]. 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 [Shing and Mahin, 1987]. 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; however, DSS is also defined within the HiL category in some literature. For a detailed discussion about different testing methods see Williams and Blakeborough [2001], Stoten and Hyde [2006].

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 [Lamarche et al., 2009, Bursi et al., 2010, Horiuchi and Konno, 2001, Wallace et al., 2005]. 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 [2006] propose a concise and generic framework 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 $S = (I + L)^{-1}$. Based on this framework, the DSS synchronization problem can be resolved by directly applying abundant control strategies, such as the linear control based on the root locus design, minimal control synthesis (MCS) [Stoten et al., 2009a], H_∞ control [Tu et al., 2009, Li et al., 2013a], neural network control [Li et al., 2011b, 2013b], model predictive control [Li et al., 2010] and anti-windup compensation [Li et al., 2011a, 2013a] techniques. However, 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 controller design.

All these 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

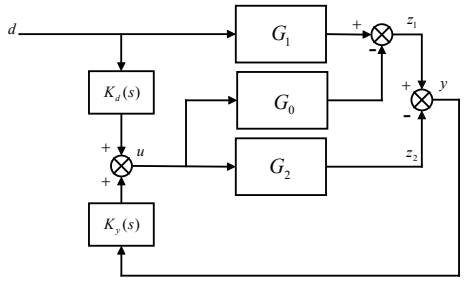

Fig. 1. A generic DSS control framework proposed by Stoten and Hyde [2006]

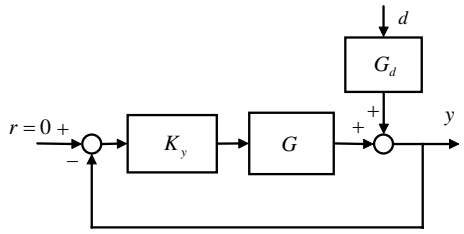

Fig. 2. Equivalent representation of Fig. 1

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 [2006] 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 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.

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 Section 4, the mass-spring-damper system is used to demonstrate the establishments of the CSFs and their transformations. The paper is concluded in Section 5.

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

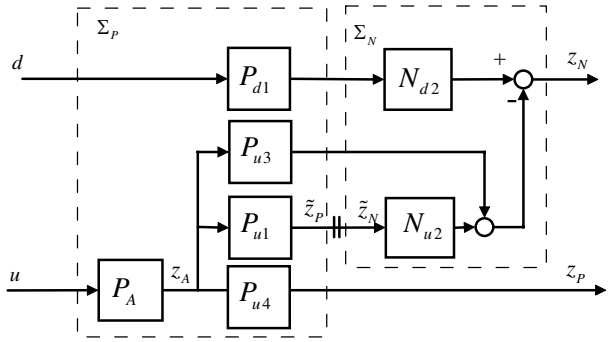

Fig. 3. The block diagram of the generic DSS framework

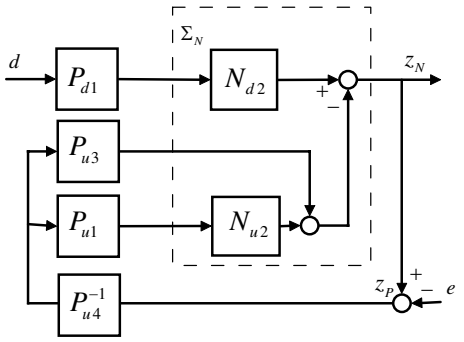

Fig. 4. The block diagram of the generic emulated system framework

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.).

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 Neild et al. [2005]). In Gawthrop et al. [2009], 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. 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.

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

$$z_N = N_{d2}P_{d1}d - P_{u3}z_A - N_{u2}\tilde{z}_N \quad (1)$$
$$\tilde{z}_P = P_{u1}z_A \quad (2)$$
$$z_A = P_A u \quad (3)$$
$$z_P = P_{u4}z_A \quad (4)$$

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

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

Hence the DSS can be expressed by

$$z_N = N_{d2}P_{d1}d - (P_{u3} + N_{u2}P_{u1})P_A u \quad (6a)$$
$$z_P = P_{u4}P_A u \quad (6b)$$

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

$$e = z_N - z_P. \quad (7)$$

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

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

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

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

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 $P_{u3}+P_{u4}+N_{u2}P_{u1}$ 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 Gawthrop et al. [2009] 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

$$e_N = z_N - z_E = P_{u4}[P_{u3} + P_{u4} + N_{u2}P_{u1}]^{-1}(P_{u3} + N_{u2}P_{u1})P_{u4}^{-1}e$$
$$e_P = z_E - z_P = P_{u4}[P_{u3} + P_{u4} + N_{u2}P_{u1}]^{-1}e$$

then the following relation hold:

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

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. 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:

$$z_P = P_{d2}N_{d1}d - N_{u3}z_A - P_{u2}\tilde{z}_P \quad (10)$$
$$\tilde{z}_N = N_{u1}z_A \quad (11)$$
$$\tilde{z}_P = P_{\tilde{A}}u \quad (12)$$
$$z_N = N_{u4}z_A \quad (13)$$

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 (10) and (13), and set the constraint variables equivalent in (10), i.e. $z := z_N = z_P$, then we have

$$z = N_{u4}(N_{u3} + N_{u4})^{-1}(P_{d2}N_{d1}d - P_{u2}\tilde{z}_P) \quad (14)$$

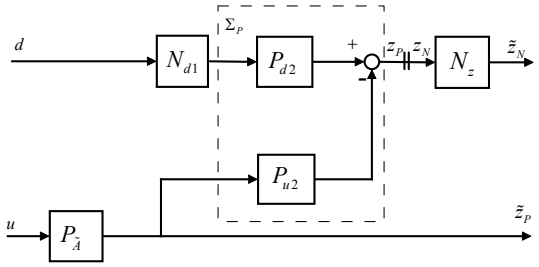

Fig. 5. The block diagram of the substructure & signal dual generic DSS framework.

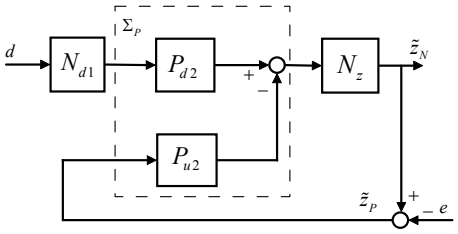

Fig. 6. The block diagram of the dual generic emulated system framework.

and

$$\tilde{z}_N = N_{u1}(N_{u3} + N_{u4})^{-1}(P_{d2}N_{d1}d - P_{u2}\tilde{z}_P) \tag{15}$$

Hence the dual DSS system can be represented as

$$\tilde{z}_N = N_{u1}(N_{u3} + N_{u4})^{-1}(P_{d2}N_{d1}d - P_{u2}P_{\tilde{A}}u) \tag{16a}$$
$$\tilde{z}_P = P_{\tilde{A}}u \tag{16b}$$

and the substructuring error is

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

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 (15) leads to

$$\begin{aligned}\tilde{z}_N &= (I + N_z P_{u2})^{-1} N_z (P_{d2}N_{d1}d + P_{u2}\tilde{e}) \\ \tilde{z}_P &= (I + N_z P_{u2})^{-1}(N_z P_{d2}N_{d1}d - \tilde{e})\end{aligned}$$

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

$$\tilde{z}_E := N_{u1}(N_{u3} + N_{u4} + P_{u2}N_{u1})^{-1}P_{d2}N_{d1}d \tag{17}$$

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

$$N_{u2} = \begin{bmatrix} 0 \\ \bar{N}_{u2} \end{bmatrix} \qquad P_{u3} = \begin{bmatrix} \bar{P}_{u3} \\ 0 \end{bmatrix} \tag{18}$$

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

$$z_N = \begin{bmatrix} z_N^{(P)} \\ z_N^{(N)} \end{bmatrix} \qquad z_P = \begin{bmatrix} z_P^{(P)} \\ z_P^{(N)} \end{bmatrix} \tag{19}$$

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

$$z_P^{(P)} = P_{u4}^{(P)} P_A u \tag{20}$$
$$z_P^{(N)} = P_{u4}^{(N)} P_A u \tag{21}$$

where

$$P_{u4}^{(P)} = \begin{bmatrix} I_{n-m} & 0_{(n-m)\times m} \end{bmatrix} P_{u4} \tag{22}$$
$$P_{u4}^{(N)} = \begin{bmatrix} 0_{m\times(n-m)} & I_m \end{bmatrix} P_{u4} \tag{23}$$

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

$$\hat{z}_N = \begin{bmatrix} z_N^{(P)} \\ \tilde{z}_N \end{bmatrix} \qquad \hat{z}_P = \begin{bmatrix} z_P^{(P)} \\ \tilde{z}_P \end{bmatrix} \tag{24}$$

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

$$\tilde{z}_P = P_{u1}P_A u = P_{\tilde{A}}u \tag{25}$$

where

$$P_{\tilde{A}} := P_{u1}P_A \tag{26}$$

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

$$\hat{z}_P := \begin{bmatrix} z_P^{(P)} \\ \tilde{z}_P \end{bmatrix} = \begin{bmatrix} P_{u4}^{(P)} \\ P_{u1} \end{bmatrix} P_A u \tag{27}$$

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 (18), (1) can be written as

$$z_N = \begin{bmatrix} z_N^{(P)} \\ z_N^{(N)} \end{bmatrix} = \begin{bmatrix} N_{d2}^{(P)} \\ N_{d2}^{(N)} \end{bmatrix} P_{d1}d - \begin{bmatrix} \bar{P}_{u3} \\ 0 \end{bmatrix} z_A - \begin{bmatrix} 0 \\ \bar{N}_{u2} \end{bmatrix} \tilde{z}_N \tag{28}$$

Suppose the inverse of $\bar{N}_{u2}$ exists. Then pre-multiplying (28) by

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

leads to

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

Combining the above equation with

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

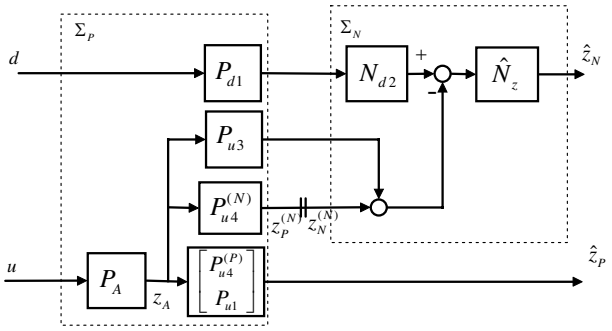

Fig. 7. The block diagram of the signal dual DSS framework

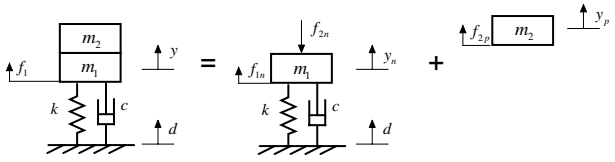

Fig. 8. Substructuring of split mass - schematic plot

leads to

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

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

$$\begin{aligned} \hat{z}_N := \begin{bmatrix} z_N^{(P)} \\ \tilde{z}_N \end{bmatrix} &= \begin{bmatrix} I & 0 \\ 0 & \bar{N}_{u2}^{-1} \end{bmatrix} N_{d2} P_{d1} d \\ &- \left(P_{u3} + \begin{bmatrix} 0_{(n-m)\times(n-m)} & 0 \\ 0 & \bar{N}_{u2}^{-1} \end{bmatrix} P_{u4} \right) P_A u \end{aligned} \tag{29}$$

By simplifying (29) and combining it with (27), we have the Sig-DSS

$$\hat{z}_N = \hat{N}_z \left(N_{d2} P_{d1} d - \begin{bmatrix} \bar{P}_{u3} \\ P_{u4}^{(N)} \end{bmatrix} P_A u \right) \tag{30a}$$

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

with $\hat{N}_z := \begin{bmatrix} I & 0 \\ 0 & \bar{N}_{u2}^{-1} \end{bmatrix}$. The resulting Sig-DSS is illustrated in Fig. 7.

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 (30) is reduced to

$$\begin{aligned} \hat{z}_N &= \tilde{z}_N = N_{u2}^{-1}(N_{d2} P_{d1} d - P_{u4} P_A u) \\ \hat{z}_P &= \tilde{z}_P = P_{\bar{A}} u = P_{u1} P_A u \end{aligned} \tag{31}$$

4. MASS SPLIT EXAMPLE

In this section, we consider a mass(m)-damper(c)-spring(k) system as shown in Fig. 8. This example is extensively studied in the DSS literature (e.g. Neild et al. [2005], Gawthrop et al. [2009]). We show how to establish DSS using the frameworks proposed in this paper.

In Fig. 8, 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. 8 is

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

and its Laplace transform representation is

$$y(s) = \frac{cs + k}{ms^2 + cs + k} d(s) \tag{32}$$

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:

$$\begin{aligned} m_1 \ddot{y}_n &= f_{1n} - f_{2n} \\ f_{1n} &= k(d - y_n) + c(\dot{d} - \dot{y}_n) \\ m_2 \ddot{y}_p &= f_{2p} \end{aligned}$$

from which we have

$$\begin{aligned} y_n &= \frac{cs + k}{m_1 s^2 + cs + k} d - \frac{1}{m_1 s^2 + cs + k} f_{2n} \\ y_p &= \frac{1}{m_2 s^2} f_{2p} \end{aligned}$$

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

$$y_n = N_{d2} d - N_{u2} P_{Af} u \tag{33}$$

$$y_p = P_{u4} P_{Af} u \tag{34}$$

with

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

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:

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

which is consistent with (32).

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 (31), we have

$$\begin{aligned} f_{2n} &= N_{u2}^{-1}(N_{d2} d - y_p) = N_{u2}^{-1}(N_{d2} d - P_{u4} P_{Af} u) \\ f_{2p} &= P_{Af} u \end{aligned}$$

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 (16a) with $N_z = N_{u4}^{-1}$ and $P_{\tilde{A}} = P_{Af}$, we have

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

Moreover, from (17) we can derive f of the emulated system as

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

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 (33) and (34). f_{2p} is generated by an actuator $f_{2p} = P_{Af}u$, so that the DSS is described by

$$y_p = P_{d2}d - P_{u2}P_{Af}u \quad (35)$$
$$y_n = N_{u4}P_{Af}u \quad (36)$$

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 $P-N:y$.

5. 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 is 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

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, 2010.

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.

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.

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.

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.

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, 2011a.

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, 2011b.

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, 2013a.

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*, 2013b. published online.

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

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

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.

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

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

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.

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, 2009a.

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, 2009b.

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.

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.

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.