A Hybrid Testing Method Based on Adaptive Feed-Forward Filters

Andreas Bartl*, Morteza Karamooz Mahdiabadi, Christina Insam, Johannes Mayet and Daniel J. Rixen*

Chair of Applied Mechanics, Technical University of Munich, Boltzmannstr. 15, 85748 Garching, Germany

Abstract

Hybrid testing is a technique which allows tests on components of complex systems with realistic boundary conditions. This can be achieved by coupling test rigs with simulation models of the overall system. Applications of hybrid testing to problems involving complex flexible structures demand careful controller design which may be prone to changes in system dynamics or model inaccuracies. Adaptive feed-forward filters offer a robust approach to coupling experimental and virtual subcomponents. In this paper, a framework based on adaptive feed-forward filters with harmonic basis functions is presented. Within the framework, a Least-Means-Squares adaption law and a Recursive-Least-Square adaption law using QR-factorization are proposed. The methods require no prior system knowledge and allow the coupling of structures with multiple degree-of-freedom interfaces. Questions related the measurement and actuation of interfaces are discussed. The methods are successfully applied to a double-clamped beam with an interface composed of two degrees of freedom. The results show the agreement of coupled test and reference system in the frequency domain as well as the ability of the methods to couple systems with arbitrary harmonic excitations. The experiments, furthermore, include a comparison of the different adaption laws in terms of the convergence behavior.

Keywords: real-time hybrid testing, adaptive feed-forward filter, recursive least-squares, transmission simulator

1. Introduction

Vibrational tests are an essential part of the development process in many engineering fields as e.g. in aerospace, biomedical, or automotive engineering. As physical testing is often more costly and time consuming than simulations, it is desirable to replace tests with simulations. In contrast, a simulation cannot always capture the complexity inherent to the physical system. One way to keep costs and complexity down is to test subcomponents of complex systems separately. The behavior of a single mechanical component, however, depends on the coupled dynamics of the overall system. To overcome this problem, methods which combine simulations and experiments have been developed. The idea is to split a complex system into a physically tested part—referred to as experimental component—and a simulated part—referred to as virtual component in this paper. An actuation system is controlled such that both components are coupled and the overall system behavior is imitated. This paradigm is called hybrid testing. The advantages of such approaches are numerous: Frequent changes in the virtual component during the development process can be flexibly integrated in tests. Boundary conditions which cannot be reproduced otherwise such as aerodynamic or fluid interaction forces can be replicated by actuators. Components that do not yet exist physically can be part of the test. This fact makes it possible to test components early in the design process. Unseen hardware defects that do not originate from the design process but from the manufacturing process can be detected during the test. In contrast to full system tests, hybrid tests can be executed under controlled conditions; they are repeatable and can be fully automated. The paradigm of coupling simulations and physical systems has been applied to a number of method classes which differ in their applications and basic assumptions. Hardware-in-The-Loop (HiL) is a well-known approach, which couples simulation and hardware tests. An overview is given e.g. in [1]. HiL simulations are widely used in automotive and aerospace industry. Mostly, controller hardware and software are tested while environment, hardware components such as mechanical components, hydraulic systems, sensors and actuators are simulated in real-time. In classical HiL simulations, actuators—if they are present at all—are part of the component under test. The product's inherent input and output channels act as interface between experimental and virtual component. Hence in contrast to real-time hybrid testing approaches, the dynamics of the transfer or actuation system do not deteriorate the test performance.

*Corresponding author
Email address: andreas.bartl@tum.de (Andreas Bartl)

Pseudo-Dynamic testing is a method which couples simulations and experiments within an extended time-scale. The method was first reported in [2]. The first applications of the Pseudo-Dynamic tests were proposed in civil engineering where structures were exposed to earthquake loads. The extended time scale implies that all time dependent effects, as inertia or damping, have to be retrieved from a model. The methods allow effects such as crack propagation in structures to be studied. The extended time scale also means that no stability issues caused by actuator delays occur. [3] introduces the use of substructures in Pseudo-Dynamic testing: Instead of coupling the rate dependent effects from a simulation to a test of the overall structure, only a subcomponent is tested while the overall structure and the rate dependent effects are simulated. [4] describes an application of Pseudo-Dynamic testing in the aerospace industry where an air-to-air refilling process is simulated with a hybrid test. The probe–drogue impacts and contact are replicated using an industrial robot. One drawback to the Pseudo-Dynamic Testing approach lies in the fact that the rate dependent effects still have to be modeled for the simulation. This is sometimes impossible and conflicts with the objective of hybrid tests.

Real-time hybrid testing overcomes the limitations of Pseudo-Dynamic testing by applying all forces in real-time and hence capturing also rate-dependent effects—as damping and inertia in the experimental component. The first uses of the method have been reported by [5], [6] and [7]. In contrast to classical HiL tests, the actuator dynamics in real-time Hybrid Tests add delays and frequency-dependent time lags. Those delays and lags can cause instabilities and deterioration in the accuracy of the test. Delays are caused by numerical time integration, actuator dynamics and the dynamics of the test specimen, i.e. the dynamics of the actuation system are changed through the coupling to the test specimen. A model of the test specimen's dynamics is usually not available and the dynamics may be subject to changes during the test. Those problems are addressed by different interface-synchronization methods. [6] proposes an algorithm for compensating constant actuator delay, which uses polynomial forward prediction. [8] and [9] discuss a delay estimation during the test which allows the prediction algorithm to adapt to the system dynamics, or in other words, to adapt to different delays at different frequencies. [10] makes use of Minimal Control Synthesis, which is a model reference control strategy. The advantage of this approach is that, even with unknown actuator or test specimen dynamics, a stable synchronization control can be achieved. [11] introduces a framework for the linear analysis of real-time hybrid testing systems, referred to as Model-In-The-Loop testing. [12] denotes real-time hybrid testing as Dynamic Substructure Systems and suggests a framework which allows the use of numerous control strategies. In this work, linear control, as well as Minimal Control Synthesis are applied. [13], [14] and [15] present work within this framework. [13] proposed a Model-Predictive Control for the synchronization of virtual and experimental component. The approach allows actuator saturation and limits to be accounted for. Model-Predictive Control is also applied by [16]. [15] makes use of neural networks as an adaptive feed-forward controller. To do so, the control problem is formulated as disturbance rejection, which extends the existing framework. [14] uses linear state space control to synchronize the interface. [17] addresses the problem of testing a structure consisting of many equivalent components by updating a model of one component during the test and then using this component model in the simulation of the virtual subsystem. The interaction control in robotics states a similar problem. The objective of this type of robot control is not to follow defined force or motions, but rather, to provide a certain impedance or admittance at the tool-center point. This makes it necessary to control the robot such that it imitates a certain dynamic model while in contact with the environment. Impedance determines a force output as a result of velocity input. An approach to impedance control was introduced by [18]. Admittance determines velocity output as a result of force input. Admittance control stability issues were investigated by [19]. See for example [20] for an overview of different interaction control laws. The application for this type of control can be found in force-feedback haptic displays, in physiotherapy or in compliant manipulation in contact with an unknown environment. The topic of compliant robots has lately become important in the context of human-robot collaboration. Usually, the applications in robotics allow the use of very simple impedance or admittance models as lumped inertias or springs. Applications of real-time hybrid testing occur in various fields of engineering such as aerospace, marine, biomedical, civil, rail or automotive engineering. Many methods are highly application dependent. As mentioned above, testing in earthquake engineering was one of the first applications of real-time hybrid testing. [5], [6] and [7] were among the first to report results in this area. [21] and [22] present an application in testing of off-shore structures. The described procedure allows tests on scaled structures in water basins. Models of aerodynamic forces are coupled to structures such as oil-drilling platforms and wind turbines. The wind forces from the model are applied using a system of wires and winches. The motion of the structures is measured using a visual tracking system. [23] describes satellite testing scenarios. [24] propose control strategies for contact simulation of spacecraft docking procedures. Automotive applications are proposed in [11]: The aerodynamics of a racing car are simulated and coupled to a chassis dynamic test rig. A second use case suggests improving hub-coupled road excitation test rigs by combing them with tire models. Similar problems as in mechanical systems occur in electrical power systems. In this area, hardware components like power electronic drives and motors are coupled to simulation models of power grids, loads or sources. [25] proposes testing inverter hardware with virtual motors and associated mechanical loads. Megawatt scale motor drives are tested using virtual gas turbine generator systems in [26]. [16] couples experimental components such as combustion engines and models of elastic drive-train dynamics. The

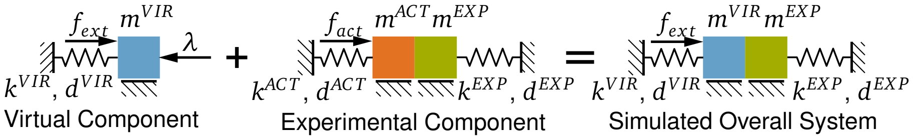


Figure 1: Lumped-mass system used for numerical investigations

test allows optimization of the coupled dynamics to reduce unwanted oscilations. Friction forces, hydraulic valve dynamics and unmodeled flexibilities make testing railway vehicle pantographs necessary. [27] and [28] propose real-time hybrid testing techniques as a replacement for expensive test drives: The pantograph is tested in the laboratory while a contact wire, messenger wire, droppers and suspensions are simulated.

So far, the real-time hybrid testing technique has rarely been applied to Noise, Vibration and Harshness (NVH) engineering. The objective of work in this field is often to reduce unwanted vibration. Especially when coupling lightly damped structures, the poles of the coupled overall systems often are close to instability. Complex and unknown dynamics make the design of hybrid testing controllers a difficult task. An approach that can overcome this problem is the use of adaptive feed-forward filters. The test dynamics are inherently more stable since the feed-forward filters do not shift the poles of the system. The adaption allows the filters to be used without prior system knowledge. Feed-forward filters have been used for testing applications in the well-established Time Waveform Replication (TWR) method. As described in [29] and [30], the technique allows replication of reference signals in vibration tests, e.g. road simulations. Usually, the actuators work under real-time control while the feed-forward filters generate the input signal. In an outer loop, the feed-forward filters are adapted using off-line iterations. This process is a type of Iterative Learning Control (ILC). The technique is commercialized and widely used in industry applications. [31] proposes the use of adaptive feed-forward for the coupling of simulation models and experiments. A Least-Mean-Squares (LMS) filter is used to test piezo actuators with realistic boundary conditions. The use of LMS filters in durability tests is described in [32]. The interface synchronization problem can be formulated as a disturbance rejection problem. This structure is used by [15] with Neural Networks as feed-forward filters. The framework defined in this paper makes use of harmonic basis functions and can be seen as extension to the above-mentioned publications for multiple DoF interfaces and a broader range of adaption laws. We propose two algorithms—one based on an LMS-type adaption law and one based on a Recursive-Least-Squares (RLS) adaption law. Additionally, problems related to calculating interface forces and moments in actuation systems are addressed. In Sec. 2 we formulate the coupling problem and rearrange it in a form, which can be used with adaptive feed-forward filters. A simple numerical example is introduced. Two types of coupling algorithms are presented in Sec. 3. The measurement of interface forces and states is discussed in Sec. 4. Finally, the technique is applied to an experimental test case in Sec. 5.

2. The coupling problem

Three subsystems are part of any hybrid test: the virtual component, the experimental component and the actuation system. In the following section, the dynamics of the three subsystems are defined. In the next step, equilibrium and compatibility constraints, which are necessary for coupling virtual and experimental component, are defined. The equilibrium constraint is met by choosing the interface forces as an input to the virtual component. Exploiting the assumptions of harmonic excitation and steady state, the equations can be rewritten such that the system responses are combinations of harmonic basis functions. This form can be used to derive a feed-forward filter, which enforces the compatibility constraint. An adaption law is chosen such that the filter coefficients converge to the correct values.

2.1. A simple numerical example

The idea of the hybrid test is illustrated in Fig. 1 for a simple lumped-mass system. Both, the virtual component (drawn in blue in Fig. 1) and the experimental component (drawn in green in Fig. 1) are mass-spring-damper systems. The excitation force $\boldsymbol{f}_{ext}$ and the interface force $\boldsymbol{\lambda}^{VIR}$ act on the virtual component. The actuator system (drawn in orange in Fig. 1) is represented by a mass-spring-damper system and the actuation force $\boldsymbol{f}_{act}$. The aim of a hybrid test is to control $\boldsymbol{f}_{act}$ and $\boldsymbol{\lambda}^{VIR}$ such that the motion of the virtual component and of the experimental system simulates the behavior of the coupled overall system. All properties, which were used for the numerical experiments below, are listed in Tab. 1. Bode plots of the dynamics of the coupled reference system, the virtual component and the test rig are shown in Fig. 5.

Virtual Component (VIR)	
m^{VIR}	$0.1\,\mathrm{kg}$
d^{VIR}	$0.05\,\frac{\mathrm{N\cdot s}}{\mathrm{m}}$
k^{VIR}	$1000\,\frac{\mathrm{N}}{\mathrm{m}}$

Experimental Component (EXP)	
m^{EXP}	$0.01\,\mathrm{kg}$
d^{EXP}	$0.05\,\frac{\mathrm{N\cdot s}}{\mathrm{m}}$
k^{EXP}	$1000\,\frac{\mathrm{N}}{\mathrm{m}}$

Actuator (ACT)	
m^{ACT}	$0.1\,\mathrm{kg}$
d^{ACT}	$1\,\frac{\mathrm{N\cdot s}}{\mathrm{m}}$
k^{ACT}	$100\,\frac{\mathrm{N}}{\mathrm{m}}$

Table 1: System parameters used in the numerical case study

2.2. Component model representation

Because we want to couple two mechanical systems, namely the virtual and the experimental components, the objective of the hybrid testing method will be to satisfy equilibrium and compatibility constraints. The dynamics of the virtual component with the system coordinates $\boldsymbol{q}^{VIR}$ are defined by Eqs. (1)

$$\begin{aligned} &\boldsymbol{M}^{VIR}\ddot{\boldsymbol{q}}^{VIR} + \boldsymbol{D}^{VIR}\dot{\boldsymbol{q}}^{VIR} + \boldsymbol{K}^{VIR}\boldsymbol{q}^{VIR} = \boldsymbol{G}^{VIR^T}\boldsymbol{\lambda}^{VIR} + \boldsymbol{f}_{ext}^{VIR} \\ &\boldsymbol{y}^{VIR} = \boldsymbol{G}^{VIR}\boldsymbol{q}^{VIR} \end{aligned} \tag{1}$$

where the mass matrix is denoted by $\boldsymbol{M}^{VIR}$, the stiffness matrix by $\boldsymbol{K}^{VIR}$, the damping matrix by $\boldsymbol{D}^{VIR}$, the external forces by $\boldsymbol{f}_{ext}^{VIR}$, and $\boldsymbol{G}^{VIR}$ restricts the system coordinates to the interface displacements $\boldsymbol{y}^{VIR}$. Accordingly, the transposed $\boldsymbol{G}^{VIR^T}$ projects the interface forces $\boldsymbol{\lambda}^{VIR}$ on the system coordinates. For the calculation of the system responses during the test, a Newmark time integration scheme can be applied. Actuation system and experimental component form a coupled system during the test. On the interface between the two, one can measure interface forces and interface displacements. A set of dynamic equations and constraints defines their coupled dynamics. Eq. (2) defines the dynamics of the experimental component with the system coordinates $\boldsymbol{q}^{EXP}$. Eq. (3) defines the dynamics of the actuators with the system coordinates $\boldsymbol{q}^{ACT}$.

$$\boldsymbol{M}^{EXP}\ddot{\boldsymbol{q}}^{EXP} + \boldsymbol{D}^{EXP}\dot{\boldsymbol{q}}^{EXP} + \boldsymbol{K}^{EXP}\boldsymbol{q}^{EXP} = \boldsymbol{f}_{ext}^{EXP} + \boldsymbol{G}^{EXP^T}\boldsymbol{\lambda}^{EXP} \tag{2}$$

$$\boldsymbol{M}^{ACT}\ddot{\boldsymbol{q}}^{ACT} + \boldsymbol{D}^{ACT}\dot{\boldsymbol{q}}^{ACT} + \boldsymbol{K}^{ACT}\boldsymbol{q}^{ACT} = \boldsymbol{B}^{ACT}\boldsymbol{u}^{ACT} + \boldsymbol{G}^{ACT^T}\boldsymbol{\lambda}^{EXP} \tag{3}$$

In these equations, the system matrices are defined correspondingly to the virtual component. $\boldsymbol{\lambda}^{EXP}$ is the vector of the interface forces between actuation system and experimental component. Accordingly, $\boldsymbol{G}^{EXP^T}$ and $\boldsymbol{G}^{ACT^T}$ project $\boldsymbol{\lambda}^{EXP}$ on the systems' coordinates. $\boldsymbol{\lambda}^{EXP}$ is measured or–if direct measurement is not possible–estimated during the experiment as described in Sec. 4. The compatibility constraint between actuation system and experimental component is defined by Eq. (4).

$$\boldsymbol{G}^{ACT}\boldsymbol{q}^{ACT} - \boldsymbol{G}^{EXP}\boldsymbol{q}^{EXP} = \boldsymbol{0} \tag{4}$$

The interface displacements $\boldsymbol{y}^{EXP}$ are defined as

$$\boldsymbol{y}^{EXP} = \boldsymbol{G}^{EXP}\boldsymbol{q}^{EXP}.$$

The equations for the experimental component and the actuators can be solved e.g. with a HHT-α method for time integration with constraints for simulation purposes. However, the system, consisting of actuation system and experimental component, does not have to be solved when using the proposed methods. It is the task of the control law to synchronize the interface of virtual and experimental component and to enforce the compatibility constraint. The interface gap $\boldsymbol{g}(t)$ defines the synchronization error in the test:

$$\boldsymbol{g} = \boldsymbol{y}^{EXP} - \boldsymbol{y}^{VIR}$$

The compatibility constraint between the virtual and the experimental component is defined by $\boldsymbol{g} = \boldsymbol{0}$. The equilibrium constraint is inherently met during the test by applying the measured interface forces $\boldsymbol{\lambda}^{EXP}$ with opposite sign to the virtual system:

$$\boldsymbol{\lambda}^{VIR} = \boldsymbol{\lambda} = -\boldsymbol{\lambda}^{EXP}$$

We refer to $\boldsymbol{\lambda}^{VIR}$ as $\boldsymbol{\lambda}$ in the following sections. Fig. 2 shows a block diagram of the described structure, where the interface compatibility is enforced by an arbitrary control law. This parallel structure makes applying standard control theory straightforward. In this contribution, the controller will be implemented as an adaptive feed-forward compensator.

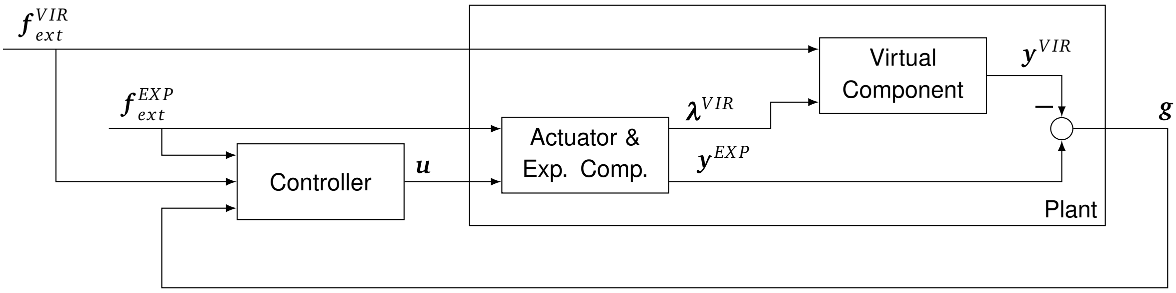


Figure 2: A parallel control structure for a hybrid test: Interface equilibrium is enforced by imposing the measured interface forces on the virtual component. A feedback controller minimizes the interface gap g such that the system meets the compatibility constraints.

2.3. Harmonic responses

The objective of this section is to reformulate the compatibility constraint such that a feed-forward filter, which couples the virtual and the experimental component, can be found. Basic assumptions are steady-state and harmonic excitation. Those assumptions are valid for many applications in NVH engineering, such as e.g. tests regarding effects of drive-train vibrations. The harmonic approach allows for efficient filters and adaption laws with low computational costs and memory consumption to be built. With both assumptions, the actuator input signal is a sum of harmonic basis functions, which can be characterized by a single vector $\boldsymbol{\theta}$. The vector has to be chosen such that both subcomponents are coupled. To do so, the interface gap $\boldsymbol{g}(t)$ is written in the frequency domain—depending on the Fourier transforms $\boldsymbol{U}(\omega)$ and $\boldsymbol{F}_{ext}(\omega)$ of the actuator inputs and the excitation forces—and the expression is set to $\mathbf{0}$. $\boldsymbol{H}_{g,u}(\omega)$ is the transfer function between actuator input and interface gap and $\boldsymbol{H}_{g,ext}(\omega)$ is the transfer function between external excitation and interface gap.

$$\boldsymbol{G}(\omega) = \boldsymbol{H}_{g,u}(\omega)\boldsymbol{U}(\omega) + \boldsymbol{H}_{g,ext}(\omega)\boldsymbol{F}_{ext}(\omega) \overset{!}{=} \mathbf{0} \tag{5}$$

An example of an excitation force vector with n_ω harmonics and n_{int} interface DoFs is given in Eq. (6).

$$\boldsymbol{f}_{ext}(t) = \boldsymbol{I}\sum_{i=1}^{n_\omega}\cos(\omega_i t) \tag{6}$$

This expression uses the excitation frequencies ω_i with $i \in [1\dots n_\omega]$ and the vector of ones $\boldsymbol{I} \in \mathbb{R}^{n_{int}}$. In the frequency domain using the Dirac δ-function, it writes

$$\boldsymbol{F}_{ext}(\omega) = \boldsymbol{I}\sqrt{2\pi}\sum_{i=1}^{n_\omega}\left(\frac{\delta(\omega-\omega_i)}{2} + \frac{\delta(\omega+\omega_i)}{2}\right).$$

For an excitation force with n_ω harmonics, the actuator input signal $\boldsymbol{u}(t)$ has to be a sum of n_ω harmonics in order to be able to couple the system. The actuator input in frequency domain is given in Eq. (7).

$$\boldsymbol{U}(\omega) = -\boldsymbol{H}_{g,u}(\omega)^{-1}\boldsymbol{H}_{g,ext}(\omega)\boldsymbol{F}_{ext}(\omega) = \sqrt{2\pi}\sum_{i=1}^{n_\omega}\boldsymbol{\psi}(\omega_i)\left(\frac{\delta(\omega-\omega_i)}{2} + \frac{\delta(\omega+\omega_i)}{2}\right) \tag{7}$$

All phase shifts and amplitudes caused by $\boldsymbol{F}_{ext}(\omega)$, $\boldsymbol{H}_{g,u}(\omega)$ and $\boldsymbol{H}_{g,ext}(\omega)$ are defined by the complex vectors $\boldsymbol{\psi}(\omega_i)$. Note that $\boldsymbol{H}_{g,u}(\omega)$ is assumed to be invertible, implying that the number of interface DoF is equal to the number of actuator DoF. Transforming $\boldsymbol{U}(\omega)$ back into the time domain and some elementary complex number operations—using the complex conjugate $\bar{\boldsymbol{\psi}}(\omega_i)$ of $\boldsymbol{\psi}(\omega_i)$—yields the matrix-vector form, Eq. (8).

$$
\begin{aligned}
\boldsymbol{u}(t) &= \mathfrak{F}^{-1}(\boldsymbol{U}(\omega)) = \frac{1}{2\pi}\int_{-\infty}^{\infty}\boldsymbol{U}(\omega)\mathrm{e}^{j\omega t}\mathrm{d}\omega = \frac{1}{2}\sum_{i=1}^{n_\omega}\left(\bar{\boldsymbol{\psi}}(\omega_i)\mathrm{e}^{-j\omega_i t} + \boldsymbol{\psi}(\omega_i)\mathrm{e}^{j\omega_i t}\right) = \sum_{i=1}^{n_\omega}\mathrm{Re}\left(\boldsymbol{\psi}(\omega_i)\mathrm{e}^{j\omega_i t}\right) \\
&= \sum_{i=1}^{n_\omega}\mathrm{Re}\,\mathrm{e}^{j\omega_i t}\mathrm{Re}\,\boldsymbol{\psi}(\omega_i) - \mathrm{Im}\,\mathrm{e}^{j\omega_i t}\mathrm{Im}\,\boldsymbol{\psi}(\omega_i) = \underbrace{\begin{bmatrix} \boldsymbol{I}\cos(\omega_1 t) \\ -\boldsymbol{I}\sin(\omega_1 t) \\ \vdots \\ \boldsymbol{I}\cos\left(\omega_{n_\omega} t\right) \\ -\boldsymbol{I}\sin\left(\omega_{n_\omega} t\right) \end{bmatrix}^T}_{\boldsymbol{W}(t)} \underbrace{\begin{bmatrix} \mathrm{Re}\,(\boldsymbol{\psi}(\omega_1)) \\ \mathrm{Im}\,(\boldsymbol{\psi}(\omega_1)) \\ \vdots \\ \mathrm{Re}\left(\boldsymbol{\psi}(\omega_{n_\omega})\right) \\ \mathrm{Im}\left(\boldsymbol{\psi}(\omega_{n_\omega})\right) \end{bmatrix}}_{\boldsymbol{\theta}}
\end{aligned}
\tag{8}
$$

The time-dependent harmonic basis functions can be written in a compact way using a matrix $\boldsymbol{W}(t) \in \mathbb{R}^{n_{int}\times 2n_{int}n_\omega}$. The characteristics of the input signal are defined by the parameter vector $\boldsymbol{\theta} \in \mathbb{R}^{2n_{int}n_\omega}$. This form can be used to generate the actuator input $\boldsymbol{u}(t)$ from $\boldsymbol{\theta}$. Using the same complex number operations, the interface gap $\boldsymbol{g}(t)$ is rearranged in matrix-vector form as can be seen in Eq. (9).

$$
\begin{aligned}
\boldsymbol{g}(t) =& \frac{1}{2\pi}\int_{-\infty}^{\infty}\boldsymbol{H}_{g,u}(\omega)\boldsymbol{U}(\omega)\mathrm{e}^{j\omega t}\mathrm{d}\omega + \frac{1}{2\pi}\int_{-\infty}^{\infty}\boldsymbol{H}_{g,ext}(\omega)\boldsymbol{F}_{ext}(\omega)\mathrm{e}^{j\omega t}\mathrm{d}\omega \\
=& \boldsymbol{W}(t)\begin{bmatrix} \mathrm{Re}\left(\boldsymbol{H}_{g,u}(\omega_1)\boldsymbol{\psi}(\omega_1)\right) \\ \mathrm{Im}\left(\boldsymbol{H}_{g,u}(\omega_1)\boldsymbol{\psi}(\omega_1)\right) \\ \vdots \\ \mathrm{Re}\left(\boldsymbol{H}_{g,u}(\omega_{n_\omega})\boldsymbol{\psi}(\omega_{n_\omega})\right) \\ \mathrm{Im}\left(\boldsymbol{H}_{g,u}(\omega_{n_\omega})\boldsymbol{\psi}(\omega_{n_\omega})\right) \end{bmatrix} + \boldsymbol{g}_{ext} \\
=& \boldsymbol{W}(t)\underbrace{\begin{bmatrix} \mathrm{Re}\left(\boldsymbol{H}_{g,u}(\omega_1)\right) & -\mathrm{Im}\left(\boldsymbol{H}_{g,u}(\omega_1)\right) & & & \\ \mathrm{Im}\left(\boldsymbol{H}_{g,u}(\omega_1)\right) & \mathrm{Re}\left(\boldsymbol{H}_{g,u}(\omega_1)\right) & & & \\ & & \ddots & & \\ & & & \mathrm{Re}\left(\boldsymbol{H}_{g,u}(\omega_{n_\omega})\right) & -\mathrm{Im}\left(\boldsymbol{H}_{g,u}(\omega_{n_\omega})\right) \\ & & & \mathrm{Im}\left(\boldsymbol{H}_{g,u}(\omega_{n_\omega})\right) & \mathrm{Re}\left(\boldsymbol{H}_{g,u}(\omega_{n_\omega})\right) \end{bmatrix}}_{\boldsymbol{P}_{g,u}}\boldsymbol{\theta} + \boldsymbol{g}_{ext}
\end{aligned}
\tag{9}
$$

$\boldsymbol{g}(t)$ is built from $\boldsymbol{W}(t)$, $\boldsymbol{\theta}$ and the interface transfer matrix $\boldsymbol{P}_{g,u}$. The interface transfer matrix $\boldsymbol{P}_{g,u} \in \mathbb{R}^{2n_{int}n_\omega\times 2n_{int}n_\omega}$ contains the transfer behavior between the actuator inputs and the interface gap in the form of real numbers. We also introduce the symbol $\boldsymbol{g}_{ext}(t)$ for the contribution of the excitation forces to the interface gap. This form can be used to derive the adaption law for $\boldsymbol{\theta}$. A perfect $\boldsymbol{\theta}$ would yield a interface gap $\boldsymbol{g}$ that is zero.

3. Adaption Laws

The matrix-vector formulation of Sec. 2.3 serves as a framework and allows numerous adaption laws to be applied. We apply two adaption laws which proof to work in the context of coupling structures: a Least-Mean-Squares adaption law, which is based on a stochastic gradient descent and a Recursive-Least-Squares adaption law, which is based on a deterministic recursive formulation. Using time step size Δt and time step number k into Eq. (9) yields the time discrete form given in Eq. (10).

$$
\boldsymbol{g}[k] = \boldsymbol{W}[k]\boldsymbol{P}_{g,u}\boldsymbol{\theta} + \boldsymbol{g}_{ext}[k]
\tag{10}
$$

In this expression, we use brackets to indicate a specific time instance. For clarity, we distinguish the current parameter vector $\boldsymbol{\theta}[k]$ and the optimal parameter vector $\boldsymbol{\theta}^o$, which couples virtual and experimental component. In the coupled state ($\boldsymbol{g}[k] = \boldsymbol{0}$) the parameter vector takes on the optimal value $\boldsymbol{\theta}[k] = \boldsymbol{\theta}^o$. Accordingly, we distinguish the true interface transfer matrix $\boldsymbol{P}_{g,u}$ and the estimated matrix $\boldsymbol{P}_{g,u}^{est}$. Note that during the adaption and the identification process, $\boldsymbol{\theta}$ and $\boldsymbol{P}_{g,u}^{est}$ change depending on time step k. Hence, they write $\boldsymbol{\theta}[k]$ and $\boldsymbol{P}_{g,u}^{est}[k]$ in the following section.

3.1. Least-Mean-Squares-based adaption law

Least-Means-Squares-type algorithms have been successfully applied to Active-Noise-Cancellation and disturbance rejection (see e.g. [33]). The properties of this algorithms include simple implementation and low computational costs. The coupling problem can be formulated in the form of a disturbance rejection problem (Fig. 3). Hence, it is possible to apply LMS algorithms to hybrid testing. The adaption law proposed in the following can be seen as a narrow-band version of the fxLMS algorithm (see e.g. [34]).

3.1.1. Adaption

The objective of the adaption process is to find the optimal parameter vector $\boldsymbol{\theta}^o$. The parameter vector $\boldsymbol{\theta}^o$ defines the actuator input, which is necessary to couple the subcomponents. The LMS-type adaption law makes use of a cost function of Eq. (11), which is the expected value of the squared interface gap $\boldsymbol{g}$.

$$J[k] = \mathrm{E}\left\{\boldsymbol{g}^T[k]\boldsymbol{g}[k]\right\} = \mathrm{E}\left\{\left(\boldsymbol{W}[k]\boldsymbol{P}_{g,u}[k]\boldsymbol{\theta}[k] + \boldsymbol{g}_{ext}[k]\right)^T \left(\boldsymbol{W}[k]\boldsymbol{P}_{g,u}[k]\boldsymbol{\theta}[k] + \boldsymbol{g}_{ext}[k]\right)\right\} \tag{11}$$

The basic idea of the LMS algorithm is a steepest descent on the cost function. The direction of the descent is defined by the negative gradient of the cost function. Eq. (12) contains the gradient of the given cost function with respect to parameter vector $\boldsymbol{\theta}$.

$$\nabla J[k] = \frac{\partial J[k]}{\partial \boldsymbol{\theta}[k]} = 2\mathrm{E}\left\{\boldsymbol{P}_{g,u}[k]^T \boldsymbol{W}[k]^T \boldsymbol{g}[k]\right\} \tag{12}$$

This expression for the gradient still depends on the expectation function $\mathrm{E}\left\{\boldsymbol{P}_{g,u}[k]^T \boldsymbol{W}[k]^T \boldsymbol{g}[k]\right\}$, which cannot be calculated directly. For the on-line adaption, the expectation function is approximated using the last sample:

$$\mathrm{E}\left\{\boldsymbol{P}_{g,u}[k]^T \boldsymbol{W}[k]^T \boldsymbol{g}[k]\right\} = \boldsymbol{P}^{est}_{g,u}[k]^T \boldsymbol{W}[k]^T \boldsymbol{g}[k]$$

The adaption law (13) ultimately results from the gradient of the cost function.

$$\boldsymbol{\theta}[k+1] = \boldsymbol{\theta}[k] - \mu_{LMS}\boldsymbol{P}^{est}_{g,u}[k]^T \boldsymbol{W}[k]^T \boldsymbol{g}[k] \tag{13}$$

The adaption gain μ_{LMS} defines the step size of the gradient descent. Note that, instead of the true interface transfer matrix $\boldsymbol{P}_{g,u}$, we have to use the estimated matrix $\boldsymbol{P}^{est}_{g,u}$. Sec. 3.1.2 shows an identification procedure to build $\boldsymbol{P}^{est}_{g,u}$. Fig. 3 shows the resulting block diagram for the adaption process: The filter uses information from the external excitation signal as an input, namely the excitation frequencies of the external forces $\boldsymbol{f}_{ext}$. They define the basis functions that are contained in $\boldsymbol{W}[k]$. The filter coefficients $\boldsymbol{\theta}$ are adapted solely from using the interface gap signal $\boldsymbol{g}$.

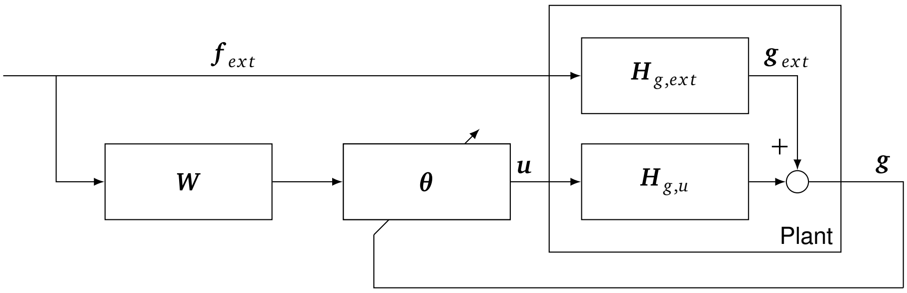


Figure 3: Block diagram for adapting $\boldsymbol{\theta}$: The system dynamics seen in Fig. 2 can be rearranged as a superposition of interface force contribution and external force contribution. Interface synchronization is realized via an LMS-type adaptive feed-forward filter.

3.1.2. Identification

As shown in the previous section, it is necessary to estimate the interface transfer matrix $\boldsymbol{P}_{g,u}$. An LMS-based approach—similar to the one used for the adaption of $\boldsymbol{\theta}$—will be used to identify $\boldsymbol{P}_{g,u}$. The basis for deriving an identification algorithm is the system identification error $\boldsymbol{e}[k]$.

$$\boldsymbol{e}[k] = \boldsymbol{W}[k]\boldsymbol{P}^{est}_{g,u}[k]\boldsymbol{\theta}[k] + \boldsymbol{g}_{ext}[k] - \boldsymbol{g}[k] \tag{14}$$

Since $\boldsymbol{g}_{ext}[k]$ may be unknown, it is removed during the identification process by setting $\boldsymbol{f}_{ext}^{VIR}(t) = \mathbf{0}$ and $\boldsymbol{f}_{ext}^{EXP}(t) = \mathbf{0}$. To make the same derivations as in the previous section, the values in the matrix $\boldsymbol{P}_{g,u}^{est}$ have to be rearranged in a vector according to the example given in Appendix A. We refer to this vector as $\tilde{\boldsymbol{P}}_{g,u}^{est}$. The resulting rearranged form in Eq. (15) contains the matrix $\tilde{\boldsymbol{\theta}}[k]$, which is a result of rearranging the expressions from Eq. (9).

$$\boldsymbol{e}[k] = \boldsymbol{W}[k]\tilde{\boldsymbol{\theta}}[k]\tilde{\boldsymbol{P}}_{g,u}^{est}[k] - \boldsymbol{g}[k] \quad \text{with} \quad \tilde{\boldsymbol{\theta}} \in \mathbb{R}^{2n_{int}^2 n_\omega \times 2n_{int}^2 n_\omega} \quad \text{and} \quad \tilde{\boldsymbol{P}}_{g,u} \in \mathbb{R}^{2n_{int}^2 n_\omega} \tag{15}$$

The cost function and the resulting adaption law are given in Eqs. (16) and (17). The identification process can be initialized with $\tilde{\boldsymbol{P}}_{g,u} = \mathbf{0}$.

$$J[k] = \mathrm{E}\left\{\boldsymbol{e}^T[k]\boldsymbol{e}[k]\right\} \tag{16}$$

$$\tilde{\boldsymbol{P}}_{g,u}^{est}[k+1] = \tilde{\boldsymbol{P}}_{g,u}^{est}[k] + \mu_{id,LMS}\tilde{\boldsymbol{\theta}}[k]^T\boldsymbol{W}[k]^T\boldsymbol{g}[k] \tag{17}$$

It is noteworthy that in order to be able to identify $\boldsymbol{P}_{g,u}$, it is necessary to use an input signal $\boldsymbol{u}(t) = \boldsymbol{W}(t)\boldsymbol{\theta}$, which excites the dynamics sufficiently. To get this input signal, the parameter vector $\boldsymbol{\theta}$ is filled with periodically changing random values. Those random values are changed in time intervals of length t_{rand}. The block diagram for the identification system is shown in Fig. 4.

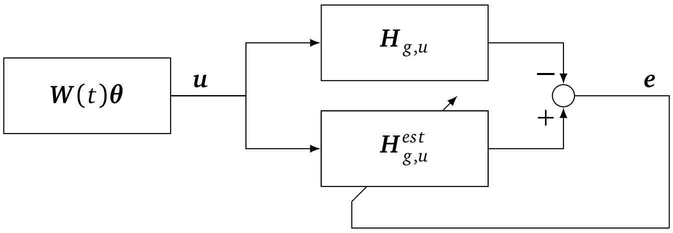

$\boldsymbol{W}(t)\boldsymbol{\theta}$
$\boldsymbol{u}$
$\boldsymbol{H}_{g,u}$
$\boldsymbol{H}_{g,u}^{est}$
−
+
$\boldsymbol{e}$

Figure 4: Block diagram for identifying $\boldsymbol{P}_{g,u}$: During the identification process, excitation forces $\boldsymbol{f}_{ext}$ seen in Fig. 3 are set to zero. The dynamics are exposed through excitations with a random vector $\boldsymbol{\theta}_i d$ at the actuator inputs.

3.1.3. Testing Procedure

Alg. 1 shows a pseudo-code for the testing procedure using the LMS-type algorithm. The procedure starts with an identification phase. The time length of the identification phase is defined by t_{id}. The identification period t_{id} is split into smaller periods t_{rand} in which $\boldsymbol{\theta}$ changes. The value of t_{rand} has to be chosen sufficiently large to allow the transients to disappear. In general, transients can be present for rapid changes of $\boldsymbol{\theta}$. For identifying $\boldsymbol{P}_{g,u}^{est}$, the vector $\boldsymbol{\theta}$ is rearranged into the matrix $\tilde{\boldsymbol{\theta}}$ according to Appendix A after each change in $\boldsymbol{\theta}$. The other way round, the matrix $\boldsymbol{P}_{g,u}^{est}$, which is used during the adaption process, is retrieved by rearrangement of the vector $\tilde{\boldsymbol{P}}_{g,u}^{est}$. The parameter vector can be initialized with $\boldsymbol{\theta} = \mathbf{0}$. It can also be initialized with an estimate from a numerical model using the transfer functions $\boldsymbol{H}_{g,u}(\omega)$ and $\boldsymbol{H}_{g,ext}(\omega)$ in order to accelerate the adaption process. The time length of the adaption phase is defined by t_{ad}.

3.1.4. Numerical experiment: Dynamics of the coupled system

In the following section, the algorithm described above is applied to the lumped-mass system of Sec. 2.1. The system is harmonically excited with the frequencies $f_k \in [5\ \mathrm{Hz}, 10\ \mathrm{Hz}, \ldots, 50\ \mathrm{Hz}]$. Fig. 5 shows the transfer functions of the open-loop and closed-loop system between excitation forces and displacements. For the closed loop system, the transfer function between excitation forces and the interface displacements of the virtual subcomponent are shown. Note that we make use of the fact that the dynamics of the proposed LMS controller can be expressed as a linear system. See [35] for the proof and details. The plots indicate that the closed-loop dynamics match with reference dynamics in phases and amplitudes exactly at the excitation frequencies. On all other points, the closed-loop transfer function stays on the curve of the virtual component. Fig. 6 shows a comparison of the closed-loop transfer functions for the described algorithm with automatically generated controllers. The controllers were created using the Matlab Control System Toolbox, but similar results can be obtained using any other tuning algorithm. The PID tuning resulted in a pure integral gain. The result is that a correct coupling is achieved only for frequencies below $1\ \mathrm{Hz}$. The system coupled with the Linear Quadratic Regulator (LQR) is far closer to the reference transfer function in a broad frequency range. However, the LQR is complex and highly dependent on the coupled system. As the following section shows, changes in the system dynamics can deteriorate the test performance or cause instabilities.

Algorithm 1 Hybrid testing with LMS-based interface synchronization

```
Initialize $\tilde{\boldsymbol{P}}^{est}_{g,u}[0] := \mathbf{0}$ and $k := 0$
while $t < t_{id}$ do
    if $\mod(t, t_{rand}) = 0$ then
        $\boldsymbol{\theta}[k]$ :=vector of random numbers
        Rearrange $\boldsymbol{\theta}[k]$ in $\tilde{\boldsymbol{\theta}}[k]$
    end if
    Set actuator input to $\boldsymbol{u}[k] = \boldsymbol{W}[k]\boldsymbol{\theta}[k]$
    Measure $\boldsymbol{\lambda}[k]$ and $\boldsymbol{y}^{EXP}[k]$
    Calculate $\boldsymbol{y}^{VIR}[k]$ through time integration (external forces $\boldsymbol{f}_{ext}[k] = \mathbf{0}$ and $\boldsymbol{\lambda}[k]$)
    $\tilde{\boldsymbol{P}}^{est}_{g,u}[k+1] = \tilde{\boldsymbol{P}}^{est}_{g,u}[k] + \mu_{LMS}\tilde{\boldsymbol{\theta}}[k]^T\boldsymbol{W}[k]^T(\boldsymbol{y}^{EXP}[k] - \boldsymbol{y}^{VIR}[k])$
    $k := k+1$
end while
Initialize $\boldsymbol{\theta}[0] := \mathbf{0}$, $k := 0$
Rearrange $\tilde{\boldsymbol{P}}^{est}_{g,u}$ in $\boldsymbol{P}^{est}_{g,u}$
while $t < t_{id} + t_{ad}$ do
    Set actuator input to $\boldsymbol{u}[k] = \boldsymbol{W}[k]\boldsymbol{\theta}[k]$
    Measure $\boldsymbol{\lambda}[k]$ and $\boldsymbol{y}^{EXP}[k]$
    Calculate $\boldsymbol{y}^{VIR}[k]$ through time integration (external forces $\boldsymbol{f}_{ext}[k]$ and $\boldsymbol{\lambda}[k]$)
    $\boldsymbol{\theta}[k+1] = \boldsymbol{\theta}[k] + \mu_{id,LMS}\boldsymbol{P}^{est}_{g,u}[k]^T\boldsymbol{W}[k]^T(\boldsymbol{y}^{EXP}[k] - \boldsymbol{y}^{VIR}[k])$
    $k := k+1$
end while
```

3.1.5. Numerical experiment: Effects of identification errors on the stability of the coupled system

The stability of the closed-loop system can deteriorate due to modifications in the system dynamics during the test or with respect to the assumed system dynamics. Those modifications can be caused e.g. by temperature-dependent effects, wear, poor system identification, or poor controller tuning. To simulate the effect of system modifications to the control performance, we modified the mass m_{EXP} in the numerical example. The LQR in Fig. 7 is tuned to the original value of m_{EXP}, while the value of m_{EXP} in the system under control is varied. More specifically, the mass of the system under test is set to values of $m^*_{EXP} \in [0.001 \text{ kg}, 0.002 \text{ kg}, \dots, 0.03 \text{ kg}]$. The poles in the right half plane show that the performance and the stability of the closed-loop system is highly sensitive to changes in the system dynamics.

In Fig. 8, the same procedure is repeated and applied to the adaptive feed-forward filter. Again, the controller is tuned to the original value of m_{EXP}, while the system under control is varied. This means, in the case of the adaptive feed-forward filter, that identifying $\boldsymbol{P}^{est}_{g,u}$ is performed on the system with the original mass.

Another perspective on the robustness of the approach using the adaptive feed-forward filter is given in Fig. 9. The stability of the closed-loop system is mapped over the estimated values of $\boldsymbol{H}^{est}_{g,u}$, which are usually obtained from the identification process. Remember that $\boldsymbol{H}^{est}_{g,u}$ is a complex scalar for $n_\omega = 1$ and $n_{int} = 1$ and the real and imaginary part of $\boldsymbol{H}^{est}_{g,u}$ are contained in the matrix $\boldsymbol{P}^{est}_{g,u}$. All maps were created using the adaption gain $\mu_{LMS} = 1$. The maps are shown for the excitation frequencies $\omega_0 = 8$ Hz, $\omega_0 = 16$ Hz, which is close to the virtual component's resonance and $\omega_0 = 21$ Hz, which is close to the coupled system's reference. According to Fig. 9, the phase error of the estimated $\boldsymbol{H}^{est}_{g,u}(\omega_0)$ with respect to the true $\boldsymbol{H}_{g,u}(\omega_0)$ can amount up to 90° for sufficiently small adaption gain. The map for the excitation frequency $\omega_0 = 16$ Hz, however, indicates that the stable region can shrink depending on the system dynamics. Even though this seems robust, for complex interfaces with multiple DoF and excitations with multiple harmonics, the performance of LMS-type filters may decline. This is the reason for the application of RLS-type algorithms to the coupling problem as explained in the next section.

3.2. Recursive-Least-Squares-based adaption law

The LMS-based algorithm from the previous section may face problems where systems with multiple DoF interfaces are coupled. The algorithm proposed in this section makes use of a Recursive-Least-Squares adaption law. RLS-based adaption laws have been successfully applied to Active-Noise-Cancellation as shown e.g. in [36] or [34]. The cost function for this type of algorithm contains the actual sum of squared interface gaps, while the LMS-based algorithm is based on the expected value of the squared interface gap. This fact makes convergence of the RLS-based algorithm faster since in each time step the exact

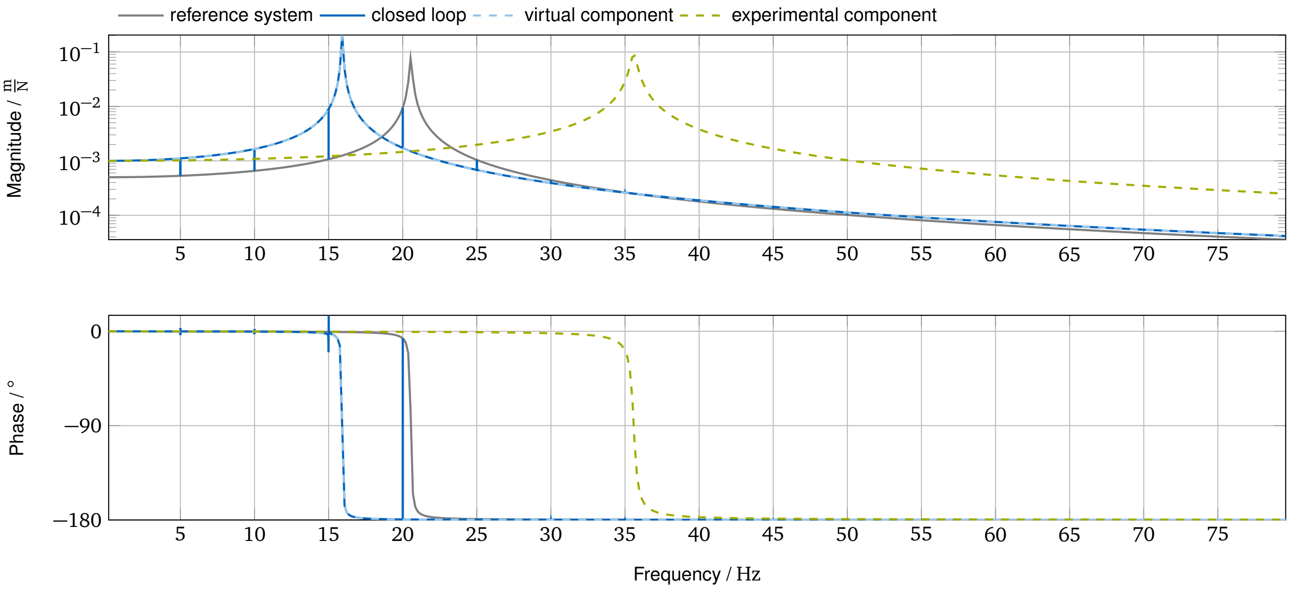


Figure 5: Coupling of two mass-spring-damper systems with an adaptive feed-forward filter. The transfer functions shows that the steady-state closed-loop dynamics match with reference dynamics at the excitation frequencies $f_k \in [5\,\mathrm{Hz}, 10\,\mathrm{Hz}, \ldots, 50\,\mathrm{Hz}]$.

solution to the sum of squares problem is obtained. Furthermore, the error of the RLS algorithm approaches zero while—due its stochastic nature–the LMS algorithm exhibits a remaining small error.

3.2.1. Adaption

The cost function (18) consists of the sum over the squared gap and a regularization term. Since the excitations or the system properties may be subject to changes during the test, the resulting solution for the parameter vector $\boldsymbol{\theta}$ should depend mainly on new data. A so-called forgetting factor μ_{RLS} $(0 < \mu_{RLS} \leq 1)$ with an exponent increasing with the age of the data ensures this property. The exponentially decreasing weight of the squared interface gap causes the contribution of old data to the solution to decrease as new data is available. The weight for the newest sample is 1. Choosing a low value of μ_{RLS} produces a fast adaption while choosing a high value yields a slow adaption. $\boldsymbol{\Xi}$ is a positive-definite matrix, which is a measure of confidence in the starting value $\boldsymbol{\theta}[0] = \boldsymbol{0}$. It can be set to $\boldsymbol{\Xi} = \beta \boldsymbol{I}$, with scalar β and unity matrix $\boldsymbol{I}$. High values of β will help to prevent jumps in $\boldsymbol{\theta}$ at the start of the adaption phase, while low values will allow a faster adaption to the optimal $\boldsymbol{\theta}^o$.

$$\begin{aligned} J[k] &= \sum_{i=0}^{k} \mu_{RLS}^{k-i} \boldsymbol{g}^T[i]\boldsymbol{g}[i] + \mu_{RLS}^{k+1} \boldsymbol{\theta}^T[k]\boldsymbol{\Xi}\boldsymbol{\theta}[k] \\ &= \sum_{i=0}^{k} \mu_{RLS}^{k-i} (\boldsymbol{W}[i]\boldsymbol{P}_{g,u}[i]\boldsymbol{\theta}[k] + \boldsymbol{g}_{ext}[i])^T (\boldsymbol{W}[i]\boldsymbol{P}_{g,u}[i]\boldsymbol{\theta}[k] + \boldsymbol{g}_{ext}[i]) + \mu_{RLS}^{k+1} \boldsymbol{\theta}^T[k]\boldsymbol{\Xi}\boldsymbol{\theta}[k] \end{aligned} \tag{18}$$

Setting the gradient of the cost function (Eq. (19)) to $\nabla J[k] = \boldsymbol{0}$ yields an equation which can be solved for $\boldsymbol{\theta}[k]$. The computational cost and memory consumption of solving the problem in each time step, however, would be high and would increase with every additional data sample. This fact is the reason for introducing an update scheme, which adds new data samples to the solution as they become available.

$$\nabla J[k] = \frac{\partial J[k]}{\partial \boldsymbol{\theta}[k]} = \sum_{i=0}^{k} 2\mu_{RLS}^{k-i} \left(\boldsymbol{P}_{g,u}^T[i]\boldsymbol{W}^T[i]\boldsymbol{W}[i]\boldsymbol{P}_{g,u}[i]\boldsymbol{\theta}[k] + \boldsymbol{P}_{g,u}^T[i]\boldsymbol{W}^T[i]\boldsymbol{g}_{ext}[i] \right) + \mu_{RLS}^{k+1}\boldsymbol{\Xi}\boldsymbol{\theta}[k] \tag{19}$$

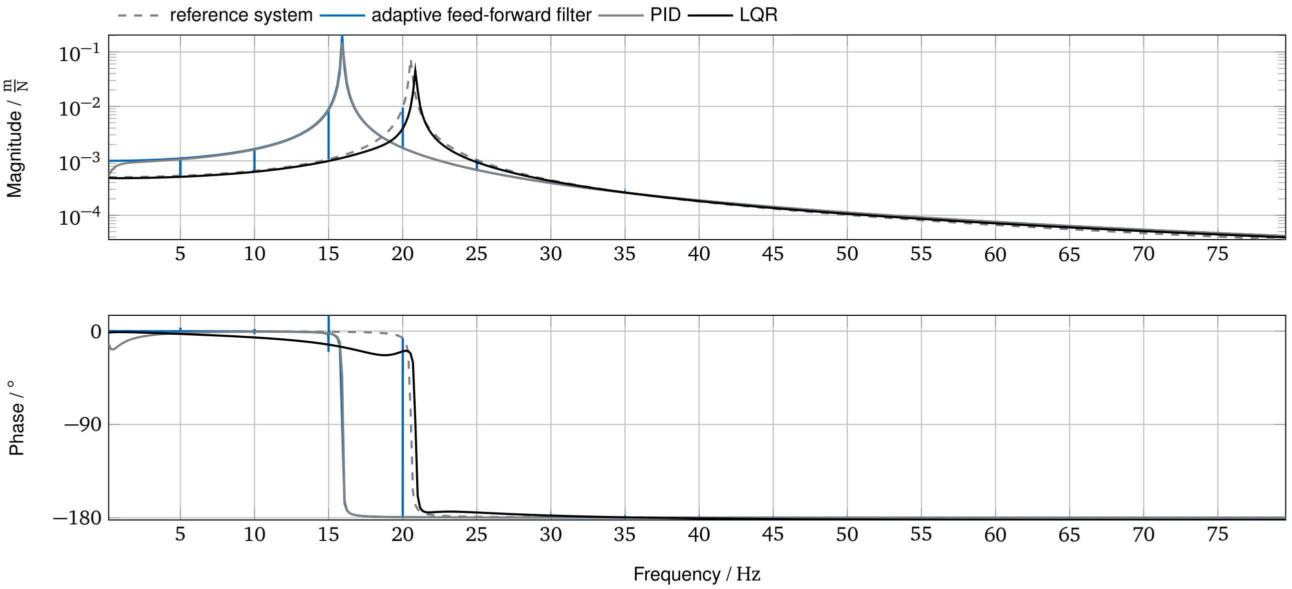


Figure 6: Comparing the closed-loop transfer functions for LMS filter-based algorithms with automatically generated controllers.

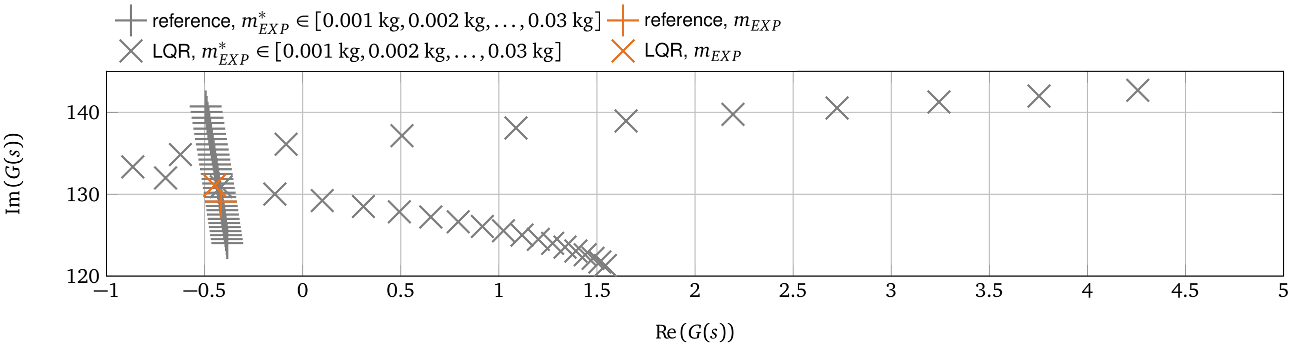


Figure 7: Detail from pole plot for closed-loop transfer functions $G(s)$: The LQR is tuned to the original value of m_{EXP}, whereas the value of $m^*_{EXP} \in [0.001\ \mathrm{kg}, 0.002\ \mathrm{kg}, \ldots, 0.03\ \mathrm{kg}]$ in the system under test is varied.

Eq. (20) is a rearranged form of the above-mentioned condition $\nabla J[k] = \mathbf{0}$. For convenience, we introduce the symbol $\boldsymbol{\Phi}$ for the terms depending on $\boldsymbol{\theta}$ and the symbol $\boldsymbol{s}$ for the terms depending on $\boldsymbol{g}_{ext}$:

$$\boldsymbol{\Phi}[k]\boldsymbol{\theta}[k] = \boldsymbol{s}[k]$$
$$\text{with}\quad \boldsymbol{\Phi}[k] = \sum_{i=0}^{k} \mu_{RLS}^{k-i} \boldsymbol{P}_{g,u}^{T}[i]\boldsymbol{W}^{T}[i]\boldsymbol{W}[i]\boldsymbol{P}_{g,u}[i] + \mu_{RLS}^{k+1}\boldsymbol{\Xi} \quad \text{and} \quad \boldsymbol{s}[k] = \sum_{i=0}^{k} -\mu_{RLS}^{k-i}\boldsymbol{P}_{g,u}^{T}[i]\boldsymbol{W}^{T}[i]\boldsymbol{g}_{ext}[i] \tag{20}$$

The update scheme is obtained by reformulating $\boldsymbol{\Phi}[k+1]$ and $\boldsymbol{s}[k+1]$ in a recursive form. To do so, the sums from the definition in Eq. (20) are split into the summands containing the previous values ($\boldsymbol{\Phi}[k]$ and $\boldsymbol{s}[k]$) and the summands containing the newly added values. Accordingly, the initialization needs to be $\boldsymbol{\Phi}[0] = \boldsymbol{\Xi}$ to agree with the definitions in Eq. (20).

$$\begin{aligned} \boldsymbol{\Phi}[k+1] &= \mu_{RLS}\boldsymbol{\Phi}[k] + \boldsymbol{P}_{g,u}^{T}[k+1]\boldsymbol{W}^{T}[k+1]\boldsymbol{W}[k+1]\boldsymbol{P}_{g,u}[k+1] \\ \boldsymbol{s}[k+1] &= \mu_{RLS}\boldsymbol{s}[k] - \boldsymbol{P}_{g,u}^{T}[k+1]\boldsymbol{W}^{T}[k+1]\boldsymbol{g}_{ext}[k+1] \end{aligned} \tag{21}$$

A numerically stable and efficient way to perform the update of $\boldsymbol{\theta}[k]$ is the so-called QR-RLS algorithm. Alg. 2 describes the full update cycle. The derivation of the algorithm is described in Appendix B. $\boldsymbol{\Phi}^{\frac{1}{2}}[0]$ is initialized as $\sqrt{\beta}\boldsymbol{I}$. The parameter

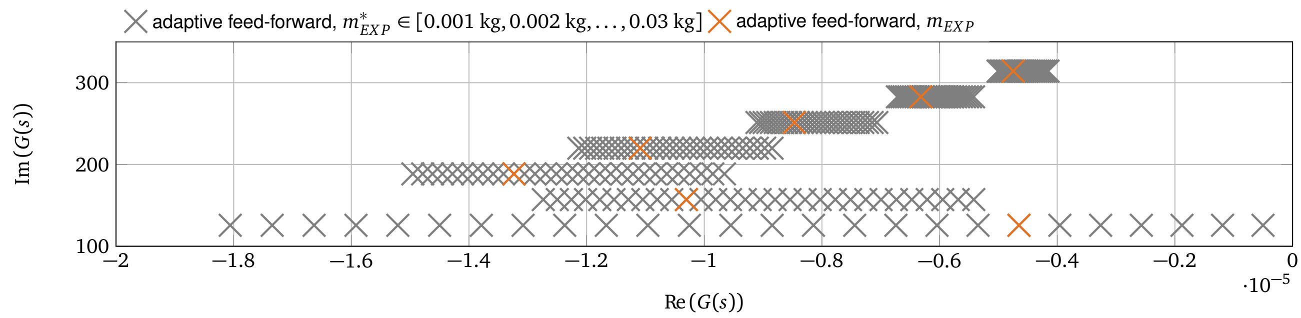


Figure 8: Detail from pole plot for closed-loop transfer functions $G(s)$: The interface transfer matrix $\boldsymbol{P}_{g,u}$ is identified with the original value of m_{EXP}, whereas the value of $m^*_{EXP} \in [0.001\,\mathrm{kg}, 0.002\,\mathrm{kg}, \dots, 0.03\,\mathrm{kg}]$ in the system under test is varied.

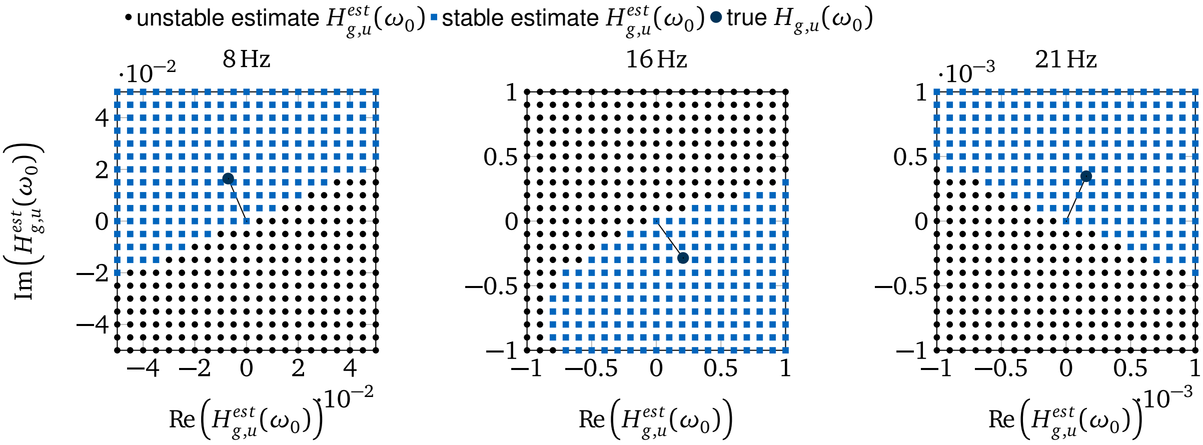


Figure 9: Stability map for a range of identified values $H^{est}_{g,u}(\omega_0)$: The three plots show stable regions for excitation frequencies of $\omega_0 = 8$ Hz, $\omega_0 = 16$ Hz and $\omega_0 = 21$ Hz. The true value $H_{g,u}(\omega_0)$ lies within the stable region.

vector is initialized with $\boldsymbol{\theta}[0] = \boldsymbol{0}$. For the correct adaption of $\boldsymbol{\theta}$, an estimate of the interface transfer matrix $\boldsymbol{P}^{est}_{g,u}$ is necessary. The identification procedure for this estimate is described in the following section. Note that, in contrast to the LMS-based algorithm, we make use of the contribution of the external forces $\boldsymbol{g}_{ext}$ to the interface gap $\boldsymbol{g}$. The interface gap itself is not explicitly calculated during the adaption process. A block diagram of the described system is shown in Fig. 10.

3.2.2. Identification

The matrix $\boldsymbol{P}_{u,g}$, which represents the system dynamics at the interface, was used to obtain $\boldsymbol{\theta}$ in the previous section. A RLS-based approach can be applied to identify $\boldsymbol{P}_{u,g}$. To do so, we use a cost function that contains the squared identification error $\boldsymbol{e}[i]$. For the purpose of identification the values of $\boldsymbol{P}_{u,g}$ have to be rearranged in the vector $\tilde{\boldsymbol{P}}_{u,g}$ according to the example given in Appendix A. Correspondingly, vector $\boldsymbol{\theta}$ is rearranged in the matrix $\tilde{\boldsymbol{\theta}}$ according to Appendix A.

$$\begin{aligned} J[k] &= \sum_{i=0}^{k} \mu^{k-i}_{id,RLS} \left(\boldsymbol{e}^T[i]\boldsymbol{e}[i]\right) + \mu^{k+1}_{id,RLS} \tilde{\boldsymbol{P}}^{est,T}_{u,g}[k] \boldsymbol{\Xi} \tilde{\boldsymbol{P}}^{est}_{u,g}[k] \\ &= \sum_{i=0}^{k} \mu^{k-i}_{id,RLS} \left(\boldsymbol{W}[i]\tilde{\boldsymbol{\theta}}[i]\tilde{\boldsymbol{P}}^{est}_{u,g}[k] - \boldsymbol{g}[i] + \boldsymbol{g}_{ext}[i]\right)^T \left(\boldsymbol{W}[i]\tilde{\boldsymbol{\theta}}[i]\tilde{\boldsymbol{P}}^{est}_{u,g}[k] - \boldsymbol{g}[i] + \boldsymbol{g}_{ext}[i]\right) \\ &\quad + \mu^{k+1}_{id,RLS} \tilde{\boldsymbol{P}}^{est,T}_{u,g}[k] \boldsymbol{\Xi} \tilde{\boldsymbol{P}}^{est}_{u,g}[k] \end{aligned} \tag{22}$$

Alg. 3 results from the recursive formulation based on the cost function in Eq. (22). The derivation is analogous to the one in the previous section: $\boldsymbol{P}^{est}_{g,u}$ is replaced by $\tilde{\boldsymbol{\theta}}$ and $\boldsymbol{\theta}$ is replaced by $\tilde{\boldsymbol{P}}^{est}_{g,u}$. The input signal $\boldsymbol{u}$ is produced by setting $\boldsymbol{\theta}$ to random values in time intervals of length t_{rand}.

Algorithm 2 Hybrid testing with RLS-based interface synchronization: adaption

```
Initialize $\boldsymbol{\Phi}^{\frac{1}{2}}[0] := \Xi^{\frac{1}{2}}$, $\boldsymbol{\theta}[0] := \mathbf{0}$ and $k := 0$
while $t < t_{ad}$ do
    Set actuator input to $\boldsymbol{u}[k] = \boldsymbol{W}[k]\boldsymbol{\theta}[k]$
    Calculate $\boldsymbol{g}_{ext}[k+1]$ through time integration (external forces $\boldsymbol{f}_{ext}[k+1]$)
    Build up matrix $\mathcal{A} = \begin{bmatrix} \sqrt{\mu_{RLS}}\boldsymbol{\Phi}^{\frac{1}{2}}[k] & \boldsymbol{W}[k+1]\boldsymbol{P}^{est}_{g,u} \\ \sqrt{\mu_{RLS}}\boldsymbol{\theta}^T[k]\boldsymbol{\Phi}^{\frac{1}{2}}[k] & -\boldsymbol{g}^T_{ext}[k+1] \end{bmatrix}$
    QR-factorization $\mathcal{A} = \mathcal{B}\mathcal{C}$
    Extract values $\begin{bmatrix} \mathcal{B}_{1,1} & \mathcal{B}_{1,2} \\ \mathcal{B}_{2,1} & \mathcal{B}_{2,2} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}^{\frac{1}{2}}[k+1] & \mathbf{0} \\ \boldsymbol{\theta}^T[k+1]\boldsymbol{\Phi}^{\frac{1}{2}}[k+1] & \mathcal{B}_{2,2} \end{bmatrix}$
    Solve $\mathcal{B}_{2,1} = \boldsymbol{\theta}^T[k+1]\mathcal{B}_{1,1}$ for $\boldsymbol{\theta}[k+1]$
    $k := k+1$
end while
```

Algorithm 3 Hybrid testing with RLS-based interface synchronization: identification

```
Initialize $\boldsymbol{\Phi}^{id,\frac{1}{2}}[0] := \Xi^{id,\frac{1}{2}}$, $\tilde{\boldsymbol{P}}^{est}_{g,u}[0] := \mathbf{0}$ and $k := 0$
while $t < t_{id}$ do
    if $\mathrm{mod}\,(t, t_{rand}) = 0$ then
        $\boldsymbol{\theta}[k+1]$ :=vector of random numbers
        Rearrange $\boldsymbol{\theta}[k+1]$ in $\tilde{\boldsymbol{\theta}}[k+1]$
    end if
    Set actuator input to $\boldsymbol{u}[k] = \boldsymbol{W}[k]\boldsymbol{\theta}[k]$
    Measure $\boldsymbol{\lambda}[k+1]$ and $\boldsymbol{y}^{EXP}[k+1]$
    Calculate $\boldsymbol{y}^{VIR}[k+1]$ through time integration (external forces $\boldsymbol{f}^{EXP}_{ext}[k+1]$ and interface forces $\boldsymbol{\lambda}[k+1]$)
    Build up matrix $\mathcal{A}^{id} = \begin{bmatrix} \sqrt{\mu_{id,RLS}}\boldsymbol{\Phi}^{id,\frac{1}{2}}[k] & \boldsymbol{W}[k+1]\tilde{\boldsymbol{\theta}}[k+1] \\ \sqrt{\mu_{id,RLS}}\tilde{\boldsymbol{P}}^{est}_{g,u}[k]\boldsymbol{\Phi}^{id,\frac{1}{2}}[k] & (\boldsymbol{y}^{EXP}[k+1]\boldsymbol{y}^{VIR}[k+1] - \boldsymbol{g}_{ext}[k+1])^T \end{bmatrix}$
    QR-factorization $\mathcal{A}^{id} = \mathcal{B}^{id}\mathcal{C}^{id}$
    Extract values $\begin{bmatrix} \mathcal{B}^{id}_{1,1} & \mathcal{B}^{id}_{1,2} \\ \mathcal{B}^{id}_{2,1} & \mathcal{B}^{id}_{2,2} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}^{id,\frac{1}{2}}[k+1] & \mathbf{0} \\ \tilde{\boldsymbol{P}}^{est,T}_{g,u}[k+1]\tilde{\boldsymbol{\Phi}}^{id,\frac{1}{2}}[k+1] & \mathcal{B}^{id}_{2,2} \end{bmatrix}$
    Solve $\mathcal{B}^{id}_{2,1} = \tilde{\boldsymbol{P}}^{est,T}_{g,u}[k+1]\mathcal{B}^{id}_{1,1}$ for $\tilde{\boldsymbol{P}}^{est}_{g,u}[k+1]$
    $k := k+1$
end while
```

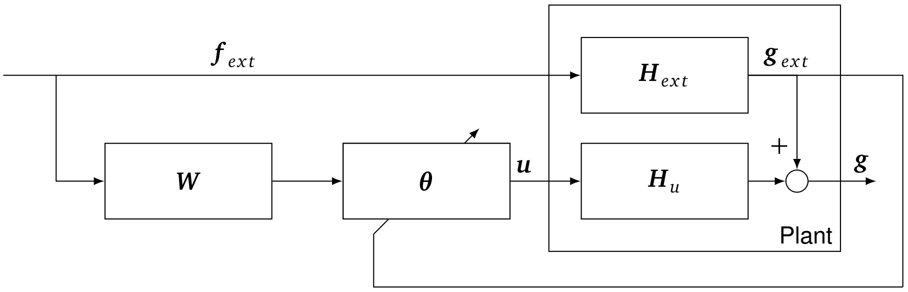


Figure 10: Block diagram for the RLS-based adaption process.

3.3. Simultaneous identification and adaption

In the sequential adaption and identification procedures described in the previous sections, changes in the systems' dynamics make a new identification step necessary. The solution is a simultaneous adaption and identification procedure. The outline of the algorithm is given in Alg. 4. An identification phase precedes the simultaneous identification and adaption process. This procedure makes it possible to initialize the interface transfer vector $\tilde{\boldsymbol{P}}^{est}_{g,u}[0]$ with the identified value, and prevents unwanted jumps in $\boldsymbol{\theta}$. Note that in this simultaneous approach, knowledge of $\boldsymbol{g}_{ext}[k]$ is necessary. Thus, applying the algorithm is possible where the external excitations or an additional identification of $\boldsymbol{g}_{ext}$ is performed.

Algorithm 4 Hybrid testing with RLS-based interface synchronization: simultaneous identification and adaption

```
Initialize $\boldsymbol{\Phi}^{id,\frac{1}{2}}[0] = \boldsymbol{\Xi}^{id,\frac{1}{2}}$, $\tilde{\boldsymbol{P}}^{est}_{g,u}[0] = \tilde{\boldsymbol{P}}^{est}_{g,u}$, $\boldsymbol{\Phi}^{\frac{1}{2}}[0] = \boldsymbol{\Xi}^{\frac{1}{2}}$, $\boldsymbol{\theta}[0] = \boldsymbol{0}$ and $k := 0$
while $t < t_{ad}$ do
    Set actuator input to $\boldsymbol{u}[k] = \boldsymbol{W}[k]\boldsymbol{\theta}[k]$
    Measure $\boldsymbol{\lambda}[k+1]$ and $\boldsymbol{y}^{EXP}[k+1]$
    Calculate $\boldsymbol{y}^{VIR}[k+1]$ through time integration (external forces $\boldsymbol{f}_{ext}[k+1]$ and $\boldsymbol{\lambda}[k+1]$)
    Calculate $\boldsymbol{g}^{VIR}_{ext}[k+1]$ through time integration (external forces $\boldsymbol{f}_{ext}[k+1]$ and $\boldsymbol{\lambda}[k+1]$)
    Rearrange $\tilde{\boldsymbol{P}}^{est}_{g,u}[k]$ in $\boldsymbol{P}^{est}_{g,u}[k]$
    Build up matrix $\mathcal{A} = \begin{bmatrix} \sqrt{\mu_{RLS}}\boldsymbol{\Phi}^{\frac{1}{2}}[k] & \boldsymbol{W}[k+1]\boldsymbol{P}^{est}_{g,u}[k] \\ \sqrt{\mu_{RLS}}\boldsymbol{\theta}^T[k]\boldsymbol{\Phi}^{\frac{1}{2}}[k] & -\boldsymbol{g}^T_{ext}[k+1] \end{bmatrix}$
    QR-factorization $\mathcal{A} = \mathcal{B}\mathcal{C}$
    Extract values $\begin{bmatrix} \mathcal{B}_{1,1} & \mathcal{B}_{1,2} \\ \mathcal{B}_{2,1} & \mathcal{B}_{2,2} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}^{\frac{1}{2}}[k+1] & \boldsymbol{0} \\ \boldsymbol{\theta}^T[k+1]\boldsymbol{\Phi}^{\frac{1}{2}}[k+1] & \mathcal{B}_{2,2} \end{bmatrix}$
    Solve $\mathcal{B}_{2,1} = \boldsymbol{\theta}^T[k+1]\mathcal{B}_{1,1}$ for $\boldsymbol{\theta}[k+1]$
    Rearrange $\boldsymbol{\theta}[k+1]$ in $\tilde{\boldsymbol{\theta}}[k+1]$
    Build up matrix $\mathcal{A}^{id} = \begin{bmatrix} \sqrt{\mu_{id,RLS}}\boldsymbol{\Phi}^{id,\frac{1}{2}}[k] & \boldsymbol{W}[k+1]\tilde{\boldsymbol{\theta}}[k+1] \\ \sqrt{\mu_{id,RLS}}\tilde{\boldsymbol{P}}^{est}_{g,u}[k]\boldsymbol{\Phi}^{id,\frac{1}{2}}[k] & ((\boldsymbol{y}^{EXP}[k+1]\boldsymbol{y}^{VIR}[k+1] - \boldsymbol{g}_{ext}[k+1])^T \end{bmatrix}$
    QR-factorization $\mathcal{A}^{id} = \mathcal{B}^{id}\mathcal{C}^{id}$
    Extract values $\begin{bmatrix} \mathcal{B}^{id}_{1,1} & \mathcal{B}^{id}_{1,2} \\ \mathcal{B}^{id}_{2,1} & \mathcal{B}^{id}_{2,2} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}^{id,\frac{1}{2}}[k+1] & \boldsymbol{0} \\ \tilde{\boldsymbol{P}}^{est,T}_{g,u}[k+1]\tilde{\boldsymbol{\Phi}}^{id,\frac{1}{2}}[k+1] & \mathcal{B}^{id}_{2,2} \end{bmatrix}$
    Solve $\mathcal{B}^{id}_{2,1} = \tilde{\boldsymbol{P}}^{est,T}_{g,u}[k+1]\mathcal{B}^{id}_{1,1}$ for $\tilde{\boldsymbol{P}}^{est}_{g,u}[k+1]$
    $k := k+1$
end while
```

3.3.1. Numerical experiment: Performance of adaptive feed-forward filters

In order to evaluate the performance of the different algorithms, we compare the learning curves of the LMS-based algorithm and the RLS-based algorithms. The system from Sec. 2.1 is excited using sinusoidal force $\boldsymbol{f}_{ext} = \cos(\omega_0 t)$ with the excitation frequency $\omega_0 = 2\pi{\cdot}8\ {}^1\!/\!_s$. The simulations were performed using four different values of μ_{LMS} and μ_{RLS}, since the performance of the algorithms is highly dependent on the choice of the adaption gains or forgetting factors. All algorithms were tested with

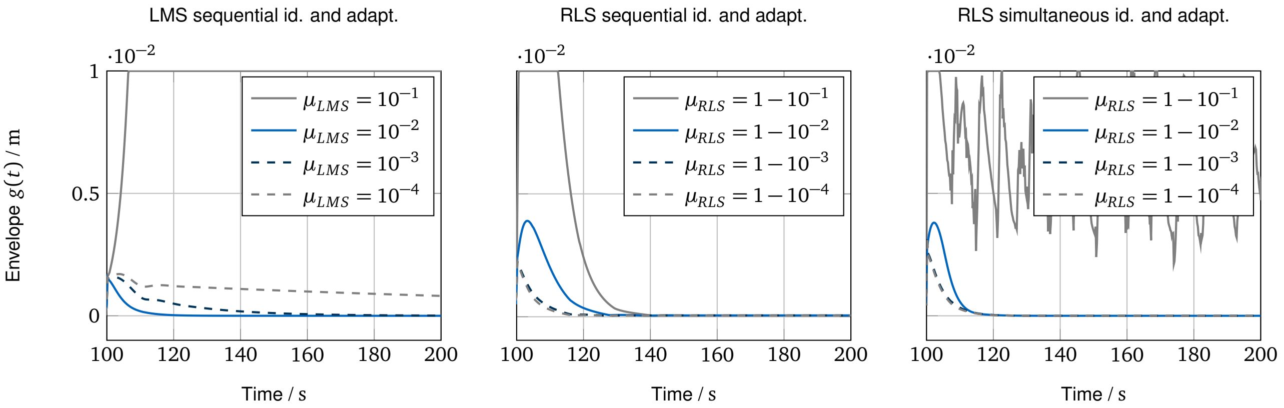


Figure 11: Learning curves for three adaption strategies. The curves show the peak envelope of the interface gap g after an identification period with a duration of $t_{id} = 100$ s.

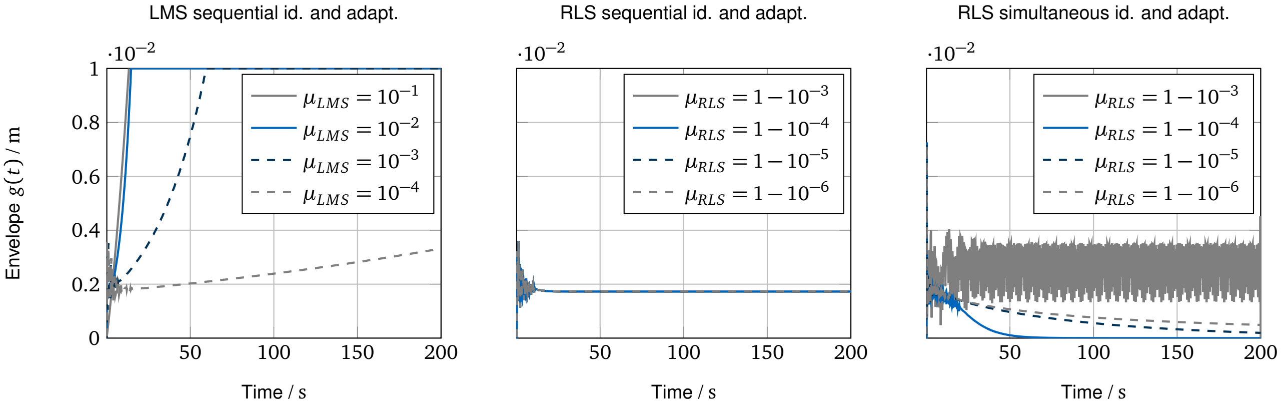


Figure 12: Learning curves for three adaption strategies. The curves show the peak envelope of the interface gap g after an identification period with a duration of $t_{id} = 0.01$ s.

a preceding identification phase (duration $t_{id} = 100$ s) using the identification gains $\mu_{id,LMS} = 0.01$ and $\mu_{id,RLS} = 0.99$. The duration of intervals, in which the identification signal parameters are updated, was chosen as $t_{rand} = 2$ s. Fig. 11 shows the envelopes of the interface gap $g[k]$. Note that the envelope of the oscillations are shown in order to improve readability of the graphs. The LMS-based adaption law exhibits an instable behavior for one value of the adaption gain μ_{LMS}. The less aggressive choices of μ_{LMS} result in a stable adaption. In contrary, the RLS-based algorithm with sequential identification and adaption shows fast convergence for all forgetting factors. The gap does, however, only go to zero if the dynamics are identified properly during the identification phase. The RLS-based algorithm with simultaneous identification and adaption converges within less than 20 s. In this case, the most aggressive choice for the forgetting factor μ_{RLS} leads to a behavior without convergence. The reason is the excitation of transient dynamics due to the fast changes in the interface force amplitudes, which is a violation of the steady-state assumptions. Note that the system remains stable, though.
Fig. 11 shows the envelopes of the interface gap $g[k]$ using an identification with a duration of $t_{id} = 0.01$ s. The short identification period has the effect that the transfer behavior is not identified correctly. The LMS-based adaption law exhibits an instable behavior for all values of the adaption gain μ_{LMS}. The RLS-based algorithm with sequential identification and adaption converges, but an error due to the wrong identification remains. For the RLS-based algorithm with simultaneous identification and adaption, the dynamics remain stable. The filter convergences and interface synchronization is achieved for choices of the forgetting factor $\mu_{RLS} \leq 1 - 10^{-}2$.

4. Interfaces

In many applications, forces, moments and displacements or rotations cannot be measured directly on the interface. In such cases, the quantities used by the coupling controller can be calculated from a model of the actuation system. Parts of the actuation system need a stiff design for the application forces and moments at the interface. One cannot avoid the fact that the improved stiffness properties go hand in hand with increased mass and inertia. Additionally, there may be stiffness and damping effects coming from the attachment of the actuators to the experimental component. An element used for the attachment as e.g. a stinger, can bring in forces or moments that cannot be measured. In the experiment described below, stingers and translational force sensors are used. Moments caused by the bending of the stinger are not measured. The following section describes procedures to compensate the dynamic effects of the actuation system and to calculate interface quantities necessary for the coupling. Starting point are the Eqs. (23), which represent the dynamics of the experimental component and the actuation system.

$$\begin{aligned}
\boldsymbol{M}^{EXP}\ddot{\boldsymbol{q}}^{EXP} + \boldsymbol{D}^{EXP}\dot{\boldsymbol{q}}^{EXP} + \boldsymbol{K}^{EXP}\boldsymbol{q}^{EXP} &= \boldsymbol{f}^{EXP}_{ext} + \boldsymbol{G}^{EXP,T}\boldsymbol{\lambda} \\
\boldsymbol{M}^{ACT}\ddot{\boldsymbol{q}}^{ACT} + \boldsymbol{D}^{ACT}\dot{\boldsymbol{q}}^{ACT} + \boldsymbol{K}^{ACT}\boldsymbol{q}^{ACT} &= \boldsymbol{B}^{ACT}_{f}\boldsymbol{f}^{ACT} + \boldsymbol{G}^{ACT,T}\boldsymbol{\lambda}
\end{aligned} \tag{23}$$

Note that the actuator input $\boldsymbol{u}^{ACT}$ in Eqs. (3) has been replaced by the actuator forces $\boldsymbol{f}^{ACT}$, since we assume that $\boldsymbol{f}^{ACT}$ can be measured during the test.

4.1. Interface Displacements and Accelerations

The compatibility between the two subsystems is given by Eq. (4). In order to obtain the contribution of the experimental component, it is necessary to measure the interface displacements or calculate them by time integration of acceleration signals. The second-order time derivatives $\ddot{\boldsymbol{q}}^{ACT}$ of the actuation system states are required for the compensation of the transfer systems' inertia effects. They can be calculated from acceleration measurements. Eqs. (24) define the relation between sensor outputs and states:

$$\boldsymbol{y}^{ACT}_{Sd} = \begin{bmatrix} y^{ACT}_{Sd,1} \\ y^{ACT}_{Sd,2} \\ \dots \end{bmatrix} = \boldsymbol{C}_{Sd}\boldsymbol{q}^{ACT} \qquad \boldsymbol{y}^{ACT}_{Sa} = \begin{bmatrix} y^{ACT}_{Sa,1} \\ y^{ACT}_{Sa,2} \\ \dots \end{bmatrix} = \boldsymbol{C}_{Sa}\ddot{\boldsymbol{q}}^{ACT} \tag{24}$$

Accordingly, $\boldsymbol{C}^{ACT}_{Sd}$ is the sensor output matrix for displacements, $\boldsymbol{C}^{ACT}_{Sa}$ is the sensor output matrix for accelerations, $\boldsymbol{y}^{ACT}_{Sd}$ is the vector of the displacement sensor outputs and $\boldsymbol{y}^{ACT}_{Sa}$ is the vector of the acceleration sensor outputs. The output matrices $\boldsymbol{C}^{ACT}_{S*} = \begin{bmatrix}\boldsymbol{C}^{ACT,T}_{S*1} & \boldsymbol{C}^{ACT,T}_{S*2} & \dots\end{bmatrix}^{T}$ are built up from submatrices representing the single sensor channels. Solving the equations using the Moore-Penrose pseudo-inverse yields the state vector $\boldsymbol{q}^{ACT}$ and its second-order time derivative $\ddot{\boldsymbol{q}}^{ACT}$:

$$\boldsymbol{q}^{ACT} = \boldsymbol{C}^{+}_{Sd}\boldsymbol{y}^{ACT}_{Sd} \qquad \ddot{\boldsymbol{q}}^{ACT} = \boldsymbol{C}^{+}_{Sa}\boldsymbol{y}^{ACT}_{Sa}$$

The required contribution to the compatibility equation, which is used for coupling virtual and experimental component, can be calculated using Eq. (25):

$$\begin{aligned}
\boldsymbol{G}^{EXP}\boldsymbol{q}^{EXP} &= \boldsymbol{G}^{ACT}\boldsymbol{q}^{ACT} = \boldsymbol{G}^{ACT}\boldsymbol{C}^{+}_{Sd}\boldsymbol{y}^{ACT}_{Sd} \\
\boldsymbol{G}^{EXP}\ddot{\boldsymbol{q}}^{EXP} &= \boldsymbol{G}^{ACT}\ddot{\boldsymbol{q}}^{ACT} = \boldsymbol{G}^{ACT}\boldsymbol{C}^{+}_{Sa}\boldsymbol{y}^{ACT}_{Sa}
\end{aligned} \tag{25}$$

For rigid body interfaces, this approach is equivalent to the virtual point method described in [37].

4.2. Interface Forces

Once the actuation systems state vectors $\boldsymbol{q}^{ACT}$, $\dot{\boldsymbol{q}}^{ACT}$ and $\ddot{\boldsymbol{q}}^{ACT}$ are identified, they can be used to calculate the interface forces $\boldsymbol{\lambda}$ from the model. To do so, one can solve the actuator Eqs. (23) for $\boldsymbol{\lambda}$:

$$\boldsymbol{\lambda} = \boldsymbol{G}^{ACT,T,+}\left(\boldsymbol{M}^{ACT}\ddot{\boldsymbol{q}}^{ACT} + \boldsymbol{D}^{ACT}\dot{\boldsymbol{q}}^{ACT} + \boldsymbol{K}^{ACT}\boldsymbol{q}^{ACT} - \boldsymbol{B}^{ACT}_{f}\boldsymbol{f}^{ACT}\right) \tag{26}$$

Neglecting all dynamics yields the pure kinematics of Eq. (27). This approach is equivalent to the calculation of the interface forces with the virtual point method in [37].

$$\boldsymbol{\lambda} = -\boldsymbol{G}^{ACT,T,+}\boldsymbol{B}^{ACT}_{f}\boldsymbol{f}^{ACT} \tag{27}$$

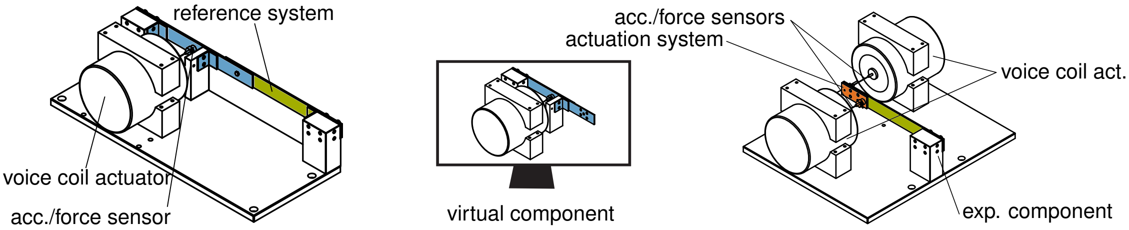


Figure 13: Reference system with actuator and sensors used for the admittance measurement. The test setup allows to couple the virtual component and the experimental component such that the dynamic behavior of the reference system is emulated.

4.3. Interface Modes

The above-mentioned theory allows to couple components with flexible interfaces. Generally, for a given sensor configuration, only a finite number of states can be observed. To overcome this problem, a reduced order model can be used. The full model of the actuation system, which can be retrieved e.g. from Finite Elements, is given by Eq. (28).

$$\boldsymbol{M}^{ACT}\ddot{\boldsymbol{u}}^{ACT}+\boldsymbol{D}^{ACT}\dot{\boldsymbol{u}}^{ACT}+\boldsymbol{K}^{ACT}\boldsymbol{u}^{ACT}=\boldsymbol{B}_A^{ACT}\boldsymbol{F}_A^{ACT}+\boldsymbol{G}^{ACT,T}\boldsymbol{\lambda} \tag{28}$$

The reduced system is given by Eq. (29) where $\boldsymbol{q}$ is the reduced coordinate vector and the matrices of the reduced system are indicated by bars:

$$\bar{\boldsymbol{M}}^{ACT}\ddot{\boldsymbol{q}}^{ACT}+\bar{\boldsymbol{D}}^{ACT}\dot{\boldsymbol{q}}^{ACT}+\bar{\boldsymbol{K}}^{ACT}\boldsymbol{q}^{ACT}=\bar{\boldsymbol{B}}_A^{ACT}\boldsymbol{f}_A^{ACT}+\bar{\boldsymbol{G}}^{ACT,T}\boldsymbol{\lambda} \tag{29}$$

The reduced matrices are retrieved using the reduction matrix $\boldsymbol{T}$:

$$\begin{aligned}
\bar{\boldsymbol{M}}^{ACT} &= \boldsymbol{T}^T\boldsymbol{M}^{ACT}\boldsymbol{T} & \bar{\boldsymbol{D}}^{ACT} &= \boldsymbol{T}^T\boldsymbol{D}^{ACT}\boldsymbol{T}\\
\bar{\boldsymbol{K}}^{ACT} &= \boldsymbol{T}^T\boldsymbol{K}^{ACT}\boldsymbol{T} & \bar{\boldsymbol{P}}_A^{ACT} &= \boldsymbol{T}^T\boldsymbol{P}_A^{ACT}\\
\bar{\boldsymbol{G}}^{ACT} &= \boldsymbol{T}^T\begin{bmatrix}\boldsymbol{I}\\ \boldsymbol{0}\end{bmatrix}
\end{aligned}$$

Reduction matrix $\boldsymbol{T}$ is retrieved e.g. according to Eq. (30) using the Craig-Bampton method with the constraint modes matrix $\boldsymbol{\Psi}^{ACT}$ and the truncated fixed interface modes matrix $\boldsymbol{\Phi}^{ACT}$. Because all states have to be observable it may be necessary to reduce the interface coordinates. $\boldsymbol{\Phi}_b$ is the interface reduction matrix and projects the coordinates of the interface between virtual and experimental component on the boundary coordinates $\boldsymbol{u}_b^{ACT}$ of the unreduced actuation system. The interface reduction matrix $\boldsymbol{\Phi}_b$ can be calculated using a model of the virtual component or a rough model of the overall coupled system. See [38] and [39] for methods of choosing an interface reduction basis.

$$\boldsymbol{u}^{ACT}=\begin{bmatrix}\boldsymbol{u}_b^{ACT}\\ \boldsymbol{u}_i^{ACT}\end{bmatrix}=\underbrace{\begin{bmatrix}\boldsymbol{\Phi}_b & \boldsymbol{0}\\ \boldsymbol{\Psi}^{ACT} & \boldsymbol{\Phi}^{ACT}\end{bmatrix}}_{\boldsymbol{T}}\underbrace{\begin{bmatrix}\boldsymbol{q}_b^{ACT}\\ \boldsymbol{q}_i^{ACT}\end{bmatrix}}_{\boldsymbol{q}^{ACT}} \tag{30}$$

5. Experiment: Coupling two cantilever beams

For an experimental demonstration of hybrid testing with adaptive feed-forward filters we use a system assembled from two cantilever beams. The excitation is chosen such that the coupling of virtual and experimental component is possible with only two actuators.

5.1. Test setup

5.1.1. Reference System

The objective of the experiment is to replicate the dynamics of a reference system as shown in Fig. 13. The two beams represent the two components of the system, which are referred to as the virtual component and the experimental component. Both subcomponents are cantilever beams connected with bolts. Each bolt is fixed with a torque of 20 Nm and washers

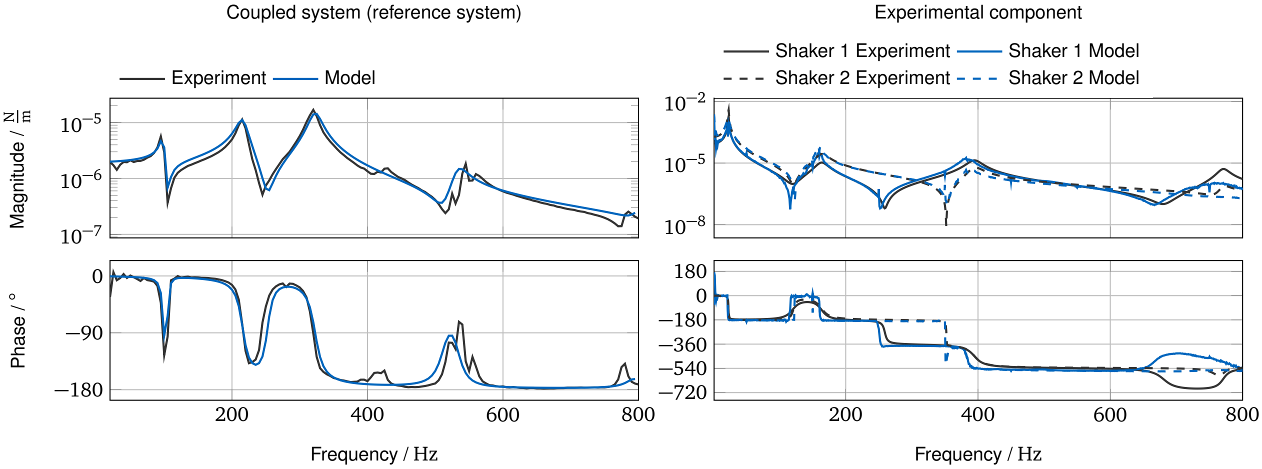


Figure 14: Measurement and model: Driving point receptance of coupled system (reference system) and test rig

Virtual Component VIR	
d_i^{VIR}	100 mm
h_{z1}^{VIR}	4 mm
h_{z2}^{VIR}	2 mm
h_y^{VIR}	40 mm
l^{VIR}	400 mm
l_f^{VIR}	30 mm
l_d^{VIR}	250 mm

Test Specimen (EXP)	
d_P^{EXP}	200 mm
h_z^{EXP}	4 mm
h_y^{EXP}	40 mm
h_f^{EXP}	30 mm
l^{EXP}	400 mm
l_f^{EXP}	30 mm

Actuation System ACT	
d^{ACT}	80 mm
h_z^{ACT}	6 mm
h_y^{ACT}	40 mm
l^{ACT}	100 mm
Material Parameters (All Components)	
Young's Modulus E	$210 \cdot 10^9 \, \frac{\mathrm{N}}{\mathrm{m}^2}$
Density ρ	$7850 \, \frac{\mathrm{kg}}{\mathrm{m}^3}$

Table 2: Dimensions of the structure used in the experiment

were placed between base and beam. Foam layers serve as dampers that are mounted on both subcomponents. The damping elements are necessary to keep the actuator displacements within its workspace at the resonances. The workspace is constrained as standard voice coil actuators use compliant elements for axial guidance. The subcomponents are bolted together with a screw applying a torque of $20\,\mathrm{Nm}$ at point B. A washer was placed in between the two subcomponents. Dimensions are given in Fig. 15 and Tab. 2. The system is excited with force $\boldsymbol{f}_{ext}$ at point P, which is part of the virtual component. Force and acceleration reference measurements are performed using a combined acceleration and force sensor (impedance head Dytran 5860B). The input force for the validation measurements is provided by a voice coil actuator (Tira S50018). In order to prevent the transmission of moments through the actuator, force sensor and voice coil actuator are connected through a thin stinger (length $100\,\mathrm{mm}$, diameter $1\,\mathrm{mm}$, spring-steel wire). Fig. 16 shows the first modes of the reference system obtained from a finite element model. Modes 1, 2, 3 and 6 are excited through the shaker in the experiment. Modes 4 and 5 are not excited due to the position of the excitation point P. The dynamics of the reference system result in the transfer functions shown in Fig. 14 as retrieved from a measurement and a calibrated model. The model consists of finite element models of the beams and spring-damper-mass elements, which represent the joints at the washers. The spring and mass coefficients were updated with the objective of matching the first four resonance frequencies. The resulting models were used for the virtual subcomponent.

5.1.2. Virtual component

We use an implicit Newmark time integration scheme (parameters $\beta = 0.25$ and $\gamma = 0.5$) to retrieve the response for the virtual component. The virtual component is represented by a calibrated finite element model. The model is reduced using the Craig-Bampton method (see [38]) using 20 fixed interface modes and the constraint modes. The interfaces used

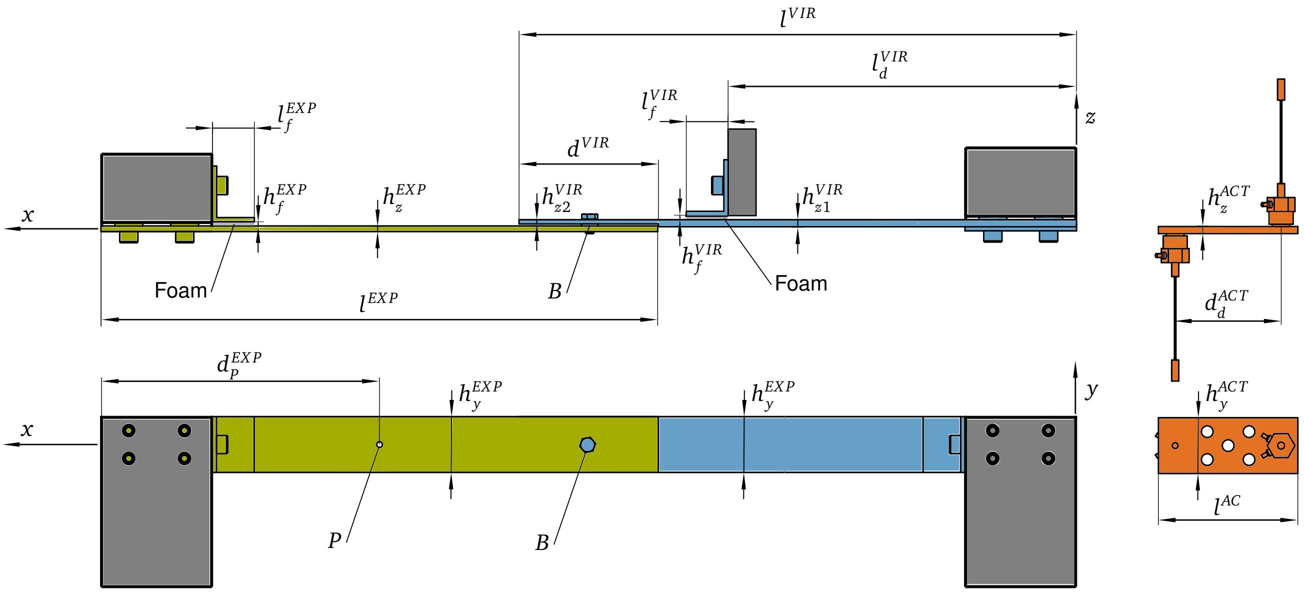
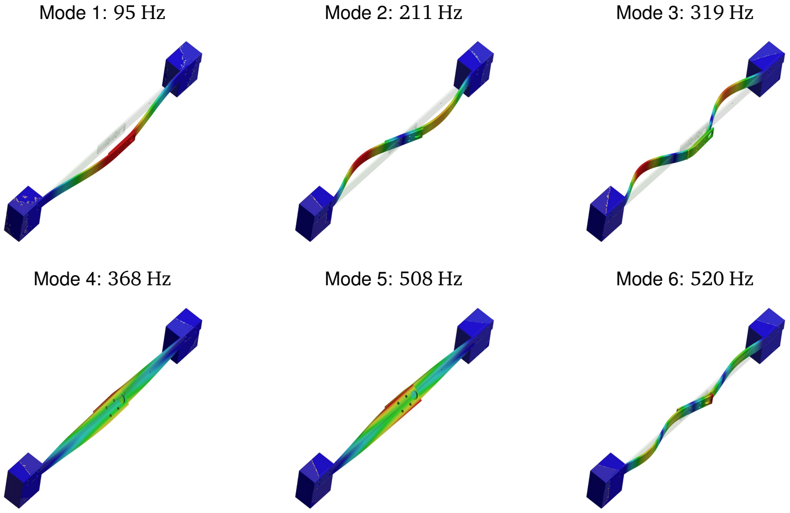


Figure 15: Dimensions of the structure used in the experiment: The values can be gathered from Tab. 2.


Mode 1: 95 Hz

Mode 2: 211 Hz

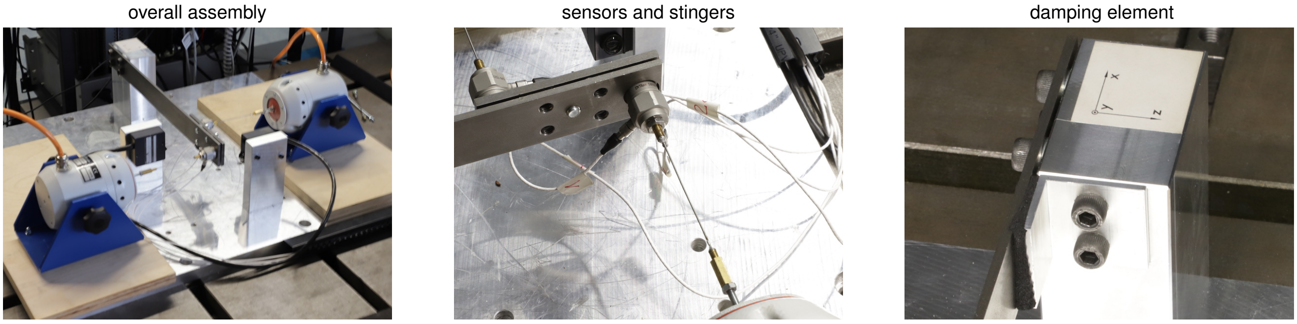
Mode 3: 319 Hz

Mode 4: 368 Hz

Mode 5: 508 Hz

Mode 6: 520 Hz

Figure 16: First normal modes of the reference system


overall assembly

sensors and stingers

damping element

Figure 17: Test rig

in the reduction are the excitation point and the interface between the substructures. The nodes which define the interfaces were rigidified and the interfaces are describes by rotations and displacements. The time step size for simulating the virtual component was $\Delta t = 0.2\,\mathrm{ms}$—as it was for the adaption law.

5.1.3. Test rig

The test rig, which is used for the hybrid coupling test, consists of the experimental component, an actuation system and a real-time computer. The set-up is shown in Fig. 17. We chose the excitation position and direction such that only deflections are excited that rotate the interface around the y-axis and cause displacements along the z-axis. As a result, we can realize the actuation system which imitates the coupling between the two beams with only two actuators. The actuation system contains voice coil actuators (Tira S50018), combined acceleration and force sensors (impedance head Dytran 5860B), thin stingers (length $100\,\mathrm{mm}$, diameter $1\,\mathrm{mm}$, spring-steel wire), and a steel plate. An ICP device supplies the impedance heads. The voice coil actuators are supplied by two amplifiers (Tira BAA120) driven in current control mode. A desktop PC running Simulink Real Time® serves as real-time computation platform. The model running on this machine includes adaption algorithm and simulation model of the virtual component and runs with a time step size of $\Delta t = 0.2\,\mathrm{ms}$. The steel plate is necessary to apply forces and moments to the interface of the experimental component and to measure rotational and translational quantities. Additionally, it imitates interface friction and stiffness of the reference system. In the following, the steel plate is assumed to be rigid.

5.1.4. Interface states and forces

The general procedure for retrieving interface states and forces is described in Sec. 4. We make use of translational acceleration ($\ddot{z}^{EXP}_{act,1}$ and $\ddot{z}^{EXP}_{act,2}$) and force sensors (actuator forces F_{A1} and F_{A2}). The coupling node of the virtual component, however, contains translational quantities (displacement z^{VIR}_B and force $F^{VIR}_{B,z}$) and rotational quantities (rotation α^{VIR} and moment $M^{VIR}_{B,y}$). Since the z^{VIR}_B and α^{VIR} have different units, the cost function resulting from their use in the interface gap $\boldsymbol{g}$ would be inconsistent in its units. For this reason, we use the actuator displacements $z_{act,1}$ and $z_{act,2}$ as coupling quantities. The displacements z^{VIR}_{act1} and z^{VIR}_{act2} used for the coupling on the virtual component's side are calculated according to Eq. (31).

$$\boldsymbol{y}^{VIR} = \begin{bmatrix} z^{VIR}_{act1} \\ z^{VIR}_{act2} \end{bmatrix} = \begin{bmatrix} 1 & -{}^{d^{ACT}}\!/_2 \\ 1 & {}^{d^{ACT}}\!/_2 \end{bmatrix} \begin{bmatrix} z^{VIR}_B \\ \alpha^{VIR} \end{bmatrix} \qquad \boldsymbol{y}^{EXP} = \begin{bmatrix} z^{EXP}_{act,1} \\ z^{EXP}_{act,2} \end{bmatrix} \tag{31}$$

On the experimental components' side we apply time integration to the acceleration signals to retrieve z^{EXP}_{act1} and z^{EXP}_{act2}. Possible drift effects are removed with a Peak filter at the excitation frequencies on the interface forces. Moment $M_{B,y}$ and the force $F_{B,z}$ define the interface force vector $\boldsymbol{\lambda}$. According to our sign conventions from Sec. 2.2, the interface forces from the experimental component are applied with opposite sign to the virtual component.

$$\boldsymbol{\lambda} = \begin{bmatrix} F^{VIR}_{B,z} & M^{VIR}_{B,y} \end{bmatrix}^T = -\boldsymbol{\lambda}_{EXP} = -\begin{bmatrix} F^{EXP}_{B,z} & M^{EXP}_{B,y} \end{bmatrix}^T \tag{32}$$

Fig. 18 illustrates the free-body diagram of the transfer system used in the experiment. The inertia properties are defined by the steel plate's mass m^{ACT} and moment of inertia θ^{ACT}_y. The interface forces are calculated by the solution of the dynamic equation of the actuation system (33).

$$\begin{bmatrix} m^{ACT} & 0 \\ 0 & \theta_y^{ACT} \end{bmatrix} \begin{bmatrix} \ddot{z}_B^{ACT} \\ \ddot{\alpha}^{ACT} \end{bmatrix} = -\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} F_{B,z}^{EXP} \\ M_{B,y}^{EXP} \end{bmatrix} + \begin{bmatrix} -1 & 1 \\ {d^{ACT}}/{2} & {d^{ACT}}/{2} \end{bmatrix} \begin{bmatrix} F_{A1} \\ F_{A2} \end{bmatrix} \tag{33}$$

5.1.5. Interface synchronization

The synchronization plots in Fig. 19 show the displacement of one side of the interface as a function of the displacement of the other side. They are a measure for the quality of the synchronization. The experiments were performed at 60 Hz, 320 Hz and 800 Hz. A QR-RLS adaption law with simultaneous adaption and identification was used for synchronization control. Here, we applied a forgetting factor $\mu_{ad} = 0.999$ for the adaption and a forgetting factor $\mu_{id} = 0.9999$ for the identification. In the synchronization plots ideal synchronization is indicated by a straight diagonal line with a slope of 1. Phase shifts cause an ellipsoidal shape of the synchronization curve while amplitude errors change the slope.

5.1.6. System transfer functions

To verify the adaptive feed-forward approach, the test setup should imitate the steady-state dynamics of the reference system. The resulting transfer functions of the virtually coupled test rig are recorded. In the following section, $H_{Pz}(\omega)$ is the driving point receptance for the assembled system, while $H_{Bz}(\omega)$ and $H_{B\alpha}(\omega)$ represent the transfer functions between the excitation forces at driving point P and the displacements on the interface B for the assembled system. We used a RLS-based adaption law with simultaneous adaption and identification as coupling algorithm. For the adaption, the forgetting factor $\mu_{ad} = 0.999$ and for identification the forgetting factor $\mu_{id} = 0.9999$ were applied. The test was conducted with step sine excitation. The reason for this approach is that phase-shifts—which cannot be instantly followed by the adaption algorithm when using a sine sweep excitation—occur at the resonances. The potentially incomplete interface synchronization would distort the results. The results in Fig. 20 show the agreement of the hybrid test with the simulated reference dynamics. The deviations are caused by the modeling errors, which can be seen in Fig. 14, and the fact that only two actuators are used to represent the interface. The high damping in the frequency range above 400 Hz is caused by friction of the sensor cables. We come to that conclusion from reference measurements without cables and a Laser-Doppler vibrometer. Moments are not accounted for by the sensors and effect the damping of rotational interface modes.

5.1.7. Learning Curves

Important properties that need to be considered for choosing one strategy in a practical application are the convergence speed and the difficulty of the adaption gain choice. For evaluating the two properties, we used sinusoidal excitation with a frequency of $\omega_{ext} = 2\pi \cdot 60\,{}^{1}/_{\mathrm{s}}$ and with $\omega_{ext} = 2\pi \cdot 320\,{}^{1}/_{\mathrm{s}}$. In order to make the two strategies comparable, we applied four different adaption gains with each strategy. For the analysis, we use a learning curve, which is the envelope of the normalized mean-square error (NMSE) of Eq. (34):

$$NMSE[i] = \frac{\mathbf{g}^T[i]\mathbf{g}[i]}{\mathbf{y}_{\max}^T \mathbf{y}_{\max}} \tag{34}$$

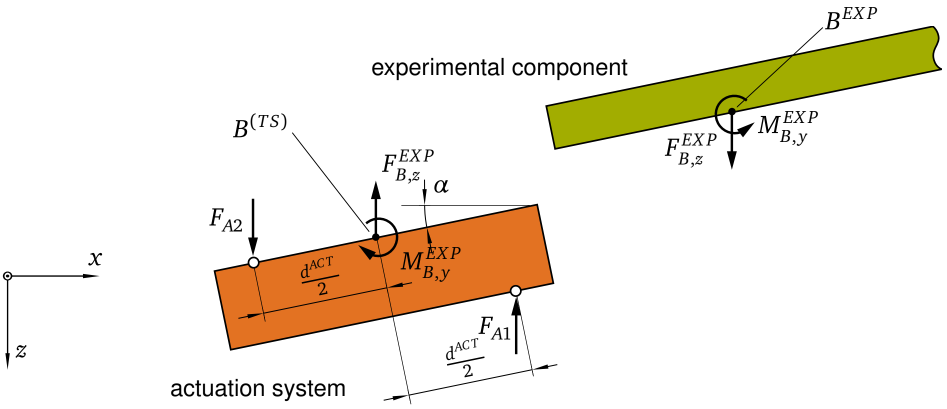


Figure 18: The interface state (y^{EXP} and $\boldsymbol{\lambda}$) includes interface rotation, displacement, force and moment. They are calculated via a rigid body model of the transfer system.

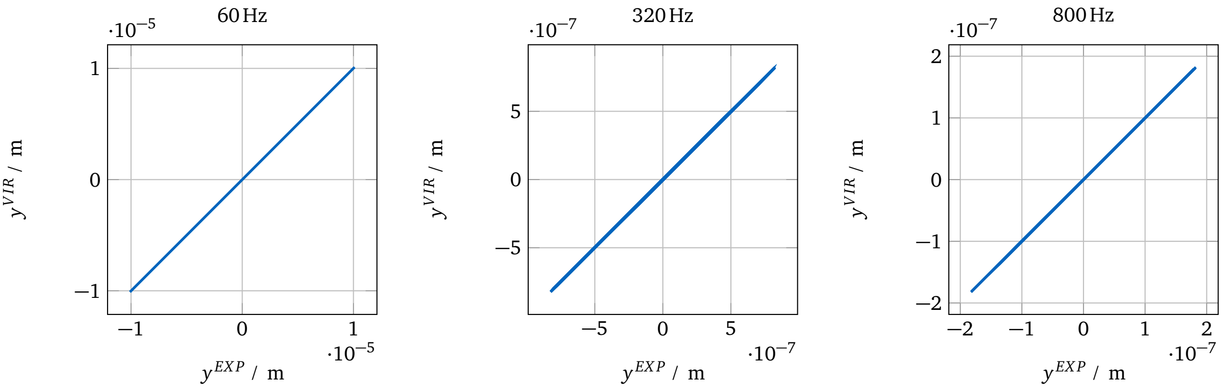


Figure 19: The synchronization plots show the relation between interface displacements of virtual and experimental component. Ideal synchronization is indicated by a straight diagonal line with a slope of 1. Phase shifts cause an ellipsoidal shape of the synchronization curve while amplitude errors change the slope.

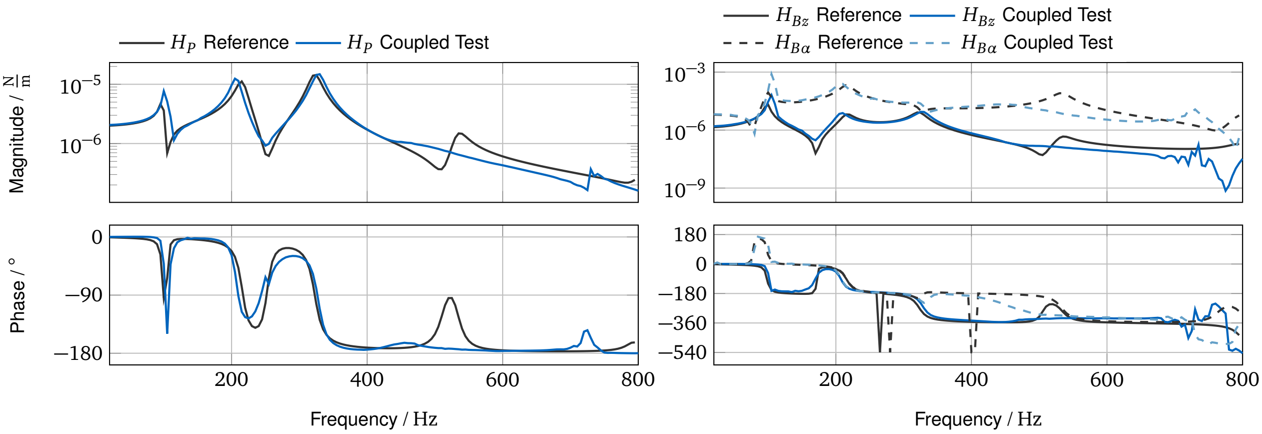


Figure 20: Driving point receptance $H_{Pz}(\omega)$, $H_{By}(\omega)$ and $H_{B\alpha}(\omega)$ of the reference system and the coupled test

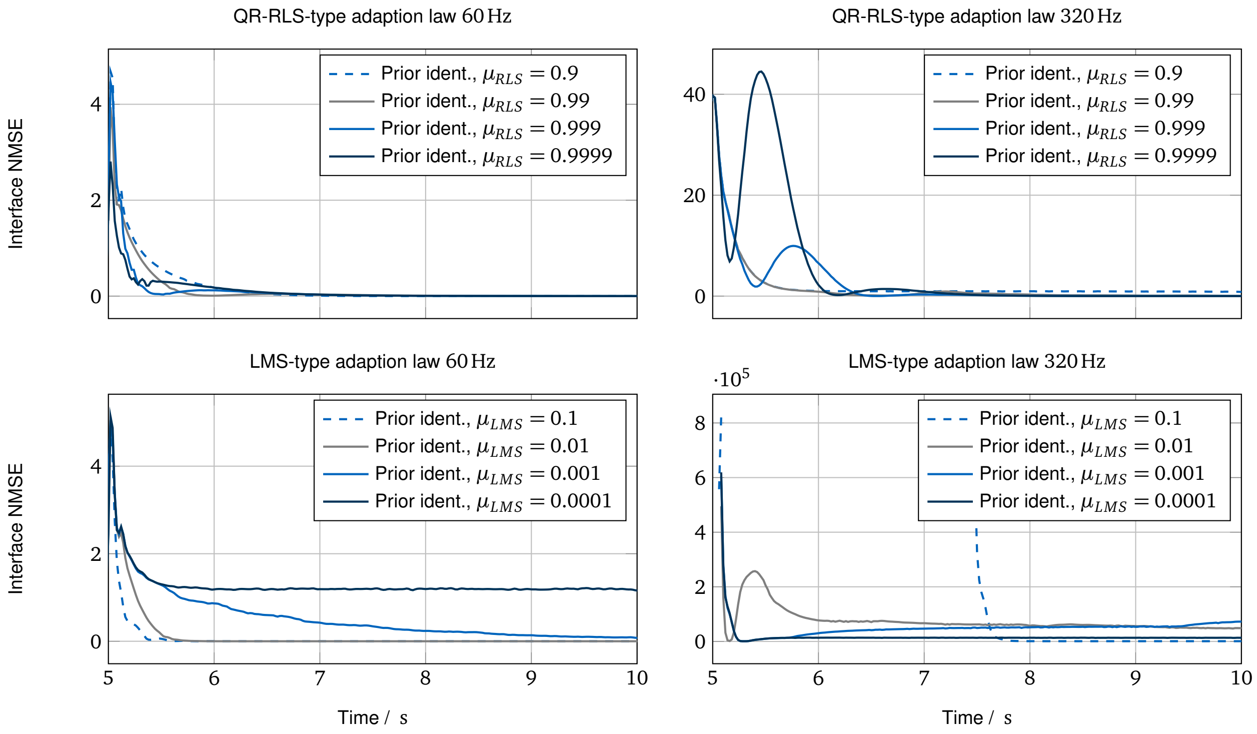


Figure 21: Learning curves for different adaption laws. The curves show the peak envelope of the normalized mean square error between virtual and experimental interface over time. The duration of the prior system identification phase is 5 s. The QR-RLS algorithm is used with simultaneous system identification.

The normalization is performed using the vector of the maximal amplitude $\boldsymbol{y}_{\max}$ of the coupled system. Fig. 21 shows the resulting learning curves. Even though the choice for an aggressive forgetting factor μ_{RLS}—this corresponds to a small value of μ_{RLS}—may deteriorate the performance of the adaption law, the results suggest that the stability of the RLS-based adaption is not affected. In contrast, the choice of a high adaption gain μ_{LMS} for the LMS adaption law may cause an instable behavior as seen at $\omega_{ext} = 2\pi \cdot 320\,\mathrm{Hz}$ for $\mu_{LMS} = 0.1$.

5.1.8. Non-sinusoidal periodic excitations

Up to this point, the experiments make use of one harmonic basis function, which is sufficient for coupling linear systems with a single excitation frequency. Additional harmonics in the basis function space allow—according to the theory—coupling of systems with arbitrary periodic excitations (period $T = {}^{1}/_{\omega_0}$). To do so, the frequencies ω_k of the harmonics in the basis function matrix $\boldsymbol{W}(t)$ are set to multiples of the periodic excitation basis frequency ω_0:

$$n_\omega \text{ Harmonics in } \boldsymbol{W}(t): \qquad \omega_k = k \cdot \omega_0 \qquad \text{with} \qquad k \in [1 \ldots n_\omega]$$

Fig. 22 shows the interface synchronization results for a square wave excitation as shown in plot (g). The basis function matrix $\boldsymbol{W}(t)$ contained one harmonic with frequency ω_0 (plot (a) and (b)), 5 harmonics (plot (c) and (d)) and 9 harmonics (plot (e) and (f)). A RLS-based adaption law with simultaneous adaption and identification was used as coupling algorithm. For the adaption the forgetting factor $\mu_{ad} = 0.999$ and for identification the forgetting factor $\mu_{id} = 0.9999$ were applied. The duration of the identification period was $t_{id} = 20s$. The results show that the increasing number of harmonics enriches the function space for the control signal and improves the interface synchronization. The methods are in general applicable to any periodic excitation: Since the mechanical system acts as a filter on the high frequency content, a finite number of harmonics is generally sufficient to couple two subcomponents.

6. Conclusion

This paper presents a framework for utilizing adaptive feed-forward filters in hybrid testing. Harmonic basis functions, which are an efficient way to reduce the filter size, are applied here. We propose algorithms based on LMS-type adaption laws

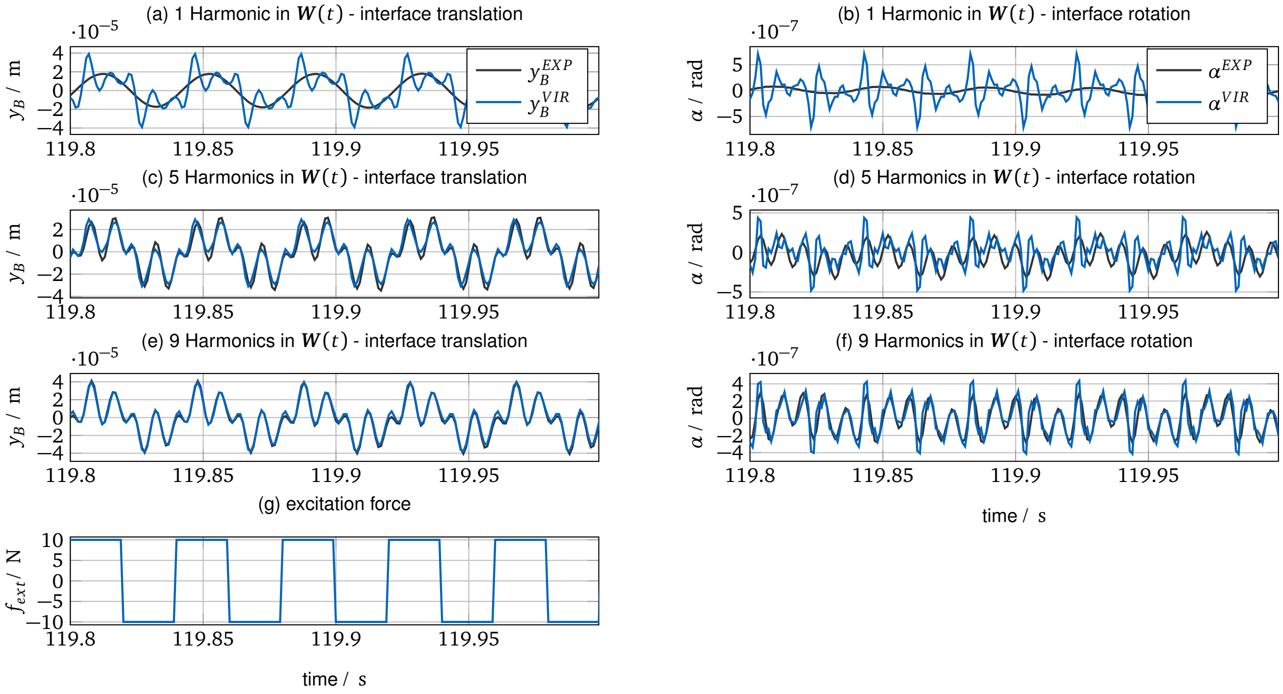


Figure 22: Interface synchronization for a square wave external excitation: The interface synchronization improves with additional harmonics in the basis function matrix $\boldsymbol{W}$.

and on RLS-type adaption laws. Both types of algorithms require a prior identification phase and assume steady-state. The LMS-based algorithm is robust against errors in the identification, since it will stay stable with a sufficiently small adaption gain for phase errors of $\pm\pi/2$. The drawback, however, is the slow convergence, which can make practical application unfeasible in some cases. Large identification errors can lead to instability. In general, the RLS-type adaption laws exhibit faster convergence. Sequential identification and adaption with identification errors can cause a remaining interface gap. The simultaneous identification and adaption process can overcome identification errors during the adaption phase. An aggressive choice of the forgetting factor, however, a can cause the violation of the steady-state assumption, since rapid changes in the harmonic interface forces excite transient dynamics. For the practical implementation of the method, it is important to compensate for unmeasured dynamics of the interface. The experiments on a beam test rig show that the methods can replicate a structure with a multiple DoF interface with standard testing equipment as modal shakers and acceleration sensors. The experiments support the presumption that it is possible to couple not only purely harmonically excited systems but also systems with arbitrary periodic excitations. Future research directions include improved model order reduction techniques for the virtual component and the combination of adaptive feed-forward filters with feed-back techniques.

Appendix A. Rearranging matrices for system identification

The following expression is a simple example for reformulation of $\boldsymbol{P}_{g,u}$ and $\boldsymbol{\theta}$ in $\tilde{\boldsymbol{P}}_{g,u}$ and $\tilde{\boldsymbol{\theta}}$, which is necessary for the identification process in Sec. 3.1.2 and Sec. 3.2.2.

$$\begin{aligned}
\boldsymbol{P}_{g,u}\boldsymbol{\theta} &= \begin{bmatrix} \mathrm{Re}\left(H_{g,u}(\omega_1)\right) & -\mathrm{Im}\left(H_{g,u}(\omega_1)\right) & 0 & 0 \\ \mathrm{Im}\left(H_{g,u}(\omega_1)\right) & \mathrm{Re}\left(H_{g,u}(\omega_1)\right) & 0 & 0 \\ 0 & 0 & \mathrm{Re}\left(H_{g,u}(\omega_2)\right) & -\mathrm{Im}\left(H_{g,u}(\omega_2)\right) \\ 0 & 0 & \mathrm{Im}\left(H_{g,u}(\omega_2)\right) & \mathrm{Re}\left(H_{g,u}(\omega_2)\right) \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} \\
&= \begin{bmatrix} \mathrm{Re}\left(H_{g,u}(\omega_1)\right)\theta_1 - \mathrm{Im}\left(H_{g,u}(\omega_1)\right)\theta_2 \\ \mathrm{Im}\left(H_{g,u}(\omega_1)\right)\theta_1 + \mathrm{Re}\left(H_{g,u}(\omega_1)\right)\theta_2 \\ \mathrm{Re}\left(H_{g,u}(\omega_2)\right)\theta_3 - \mathrm{Im}\left(H_{g,u}(\omega_2)\right)\theta_4 \\ \mathrm{Im}\left(H_{g,u}(\omega_2)\right)\theta_3 + \mathrm{Re}\left(H_{g,u}(\omega_2)\right)\theta_4 \end{bmatrix} \\
&= \underbrace{\begin{bmatrix} \theta_1 & -\theta_2 & 0 & 0 \\ \theta_2 & \theta_1 & 0 & 0 \\ 0 & 0 & \theta_3 & -\theta_4 \\ 0 & 0 & \theta_4 & \theta_3 \end{bmatrix}}_{\tilde{\boldsymbol{\theta}}} \underbrace{\begin{bmatrix} \mathrm{Re}\left(H_{g,u}(\omega_1)\right) \\ \mathrm{Im}\left(H_{g,u}(\omega_1)\right) \\ \mathrm{Re}\left(H_{g,u}(\omega_2)\right) \\ \mathrm{Im}\left(H_{g,u}(\omega_2)\right) \end{bmatrix}}_{\tilde{\boldsymbol{P}}_{g,u}}
\end{aligned}$$

Appendix B. Proof for RLS-based algorithm

The following derivation refers to the RLS algorithm, which is used in Sec. 3.2.1. In Eq. 21, the matrix $\boldsymbol{\Phi}$ is only changed by the rank-n_{int} matrix $\boldsymbol{P}_{g,u}^T\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{P}_{g,u}$ in one time step. Using this fact, an approach based on QR-factorization is an efficient and numerically stable way to obtain the solution of Eq. (20). $\boldsymbol{\Phi}$ is symmetric and positive-definite, since it is the sum of positive definite matrices. Hence, it can be decomposed into the so-called square-root factors as described by Eq. (B.1). The other way round, if $\boldsymbol{\Phi}[k]$ is built up from the square-root factors $\boldsymbol{\Phi}^{\frac{1}{2}}[k]$, positive-definiteness is ensured.

$$\boldsymbol{\Phi}[k] = \boldsymbol{\Phi}^{\frac{1}{2}}[k]\boldsymbol{\Phi}^{\frac{T}{2}}[k] \tag{B.1}$$

Using the definition of the square-root factors in Eq. (B.1), the update Eqs. (21) can be split into factors according to Eq. (B.2). One can verify that Eq. (B.2) is equivalent to Eqs. (21).

$$\begin{aligned}
&\mathcal{A}\mathcal{A}^T = \mathcal{B}\mathcal{B}^T \\
&\text{with} \quad \mathcal{A} = \begin{bmatrix} \sqrt{\mu_{RLS}}\boldsymbol{\Phi}^{\frac{1}{2}}[k] & \boldsymbol{W}[k+1]\boldsymbol{P}_{g,u}[k+1] \\ \sqrt{\mu_{RLS}}\boldsymbol{\theta}^T[k]\boldsymbol{\Phi}^{\frac{1}{2}}[k] & -\boldsymbol{g}_{ext}^T[k+1] \end{bmatrix} \\
&\text{and} \quad \mathcal{B} = \begin{bmatrix} \mathcal{B}_{1,1} & \mathcal{B}_{1,2} \\ \mathcal{B}_{2,1} & \mathcal{B}_{2,2} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}^{\frac{1}{2}}[k+1] & \boldsymbol{0} \\ \boldsymbol{\theta}^T[k+1]\boldsymbol{\Phi}^{\frac{1}{2}}[k+1] & \mathcal{B}_{2,2} \end{bmatrix}
\end{aligned} \tag{B.2}$$

If an orthogonal transformation with the matrix $\mathcal{C}$ ($\mathcal{C}^T\mathcal{C} = \boldsymbol{I}$) according to Eq. (B.3) exists, Eq. (B.2)—which corresponds to the update equations—holds (proof in [40]).

$$\mathcal{A} = \mathcal{B}\mathcal{C} \quad \text{or} \quad \mathcal{A}^T = \mathcal{C}^T\mathcal{B}^T \quad \text{or} \quad \mathcal{C}\mathcal{A} = \mathcal{B} \quad \text{with} \quad \mathcal{C}^T\mathcal{C} = \boldsymbol{I} \tag{B.3}$$

This orthogonal transformation can be seen as a sequence of rotations which transform $\mathcal{A}$ into $\mathcal{B}$. It corresponds to a QR-factorization of $\mathcal{A}^T$ into $\mathcal{B}^T$ and $\mathcal{C}^T$. This means that if $\mathcal{B}$ is obtained from $\mathcal{A}$ via QR-factorization, the entries of $\mathcal{B}$ — $\boldsymbol{\theta}^T[k+1]\boldsymbol{\Phi}^{\frac{1}{2}}[k+1]$ and $\boldsymbol{\Phi}^{\frac{1}{2}}[k+1]$ — obey the update Eqs. (21). It is noteworthy that it is not necessary to find $\mathcal{C}$ explicitly in order to retrieve $\mathcal{B}$. Since $\mathcal{A}$ contains the matrices $\boldsymbol{\theta}[k]\boldsymbol{\Phi}^{\frac{1}{2}}[k]$ and $\boldsymbol{\Phi}^{\frac{1}{2}}[k]$ of time step k and $\mathcal{B}$ contains the matrices $\boldsymbol{\theta}[k+1]\boldsymbol{\Phi}^{\frac{1}{2}}[k+1]$ and $\boldsymbol{\Phi}^{\frac{1}{2}}[k+1]$ from time step $k+1$, the QR-factorization of $\mathcal{A}$ can be used to update $\boldsymbol{\theta}$. To do so, the final step is to solve Eq. (B.4) for $\boldsymbol{\theta}[k+1]$.

$$\boldsymbol{\theta}^T[k+1]\mathcal{B}_{1,1} = \mathcal{B}_{2,1} \tag{B.4}$$

References

[1] R. Isermann, J. Schaffnit, S. Sinsel, Hardware-in-the-Loop Simulation for the Design and Testing of Engine-Control Systems, IFAC Proceedings Volumes 31 (4) (1998) 1–10. doi:10.1016/S1474-6670(17)42125-2.

[2] S. A. Mahin, P.-S. Shing, C. Thewalt, R. Hanson, Pseudodynamic Test Method -Current Status and Future Directions, Journal of Structural Engineering 115 (8) (1989) 2113–2128.

[3] P. Buchet, P. Pegon, PSD Testing with Substructuring: Implementation and Use, Tech. rep., Joint Research Centre, European Commission Special Publication No.I.94.25, Ispra, Italy (1994).

[4] M. Bolien, P. Iravani, J. Du Bois, Toward robotic pseudodynamic testing for hybrid simulations of air-to-air refueling, IEEE/ASME Transactions on Mechatronics 22 (2) (2017) 1004–1013. doi:10.1109/TMECH.2016.2644200.

[5] M. Nakashima, H. Kato, E. Takaoka, Development of Real-Time Pseudo Dynamic Testing, Earthquake Engineering & Structural Dynamics (1992) 79–92.

[6] T. Horiuchi, M. Inoue, T. Konno, Y. Namita, Real-time Hybrid Experimental System with Actuator Delay Compensation and its Application to a Piping System with Energy Absorber., Earthquake Engineering and Structural Dynamics 28 (10) (1999) 1121–1141.

[7] A. Blakeborough, A. Darby, M. Williams, The development of real-time substructure testing, Philosophical transactions: Mathematical, Physical and Engineering Sciences 359 (1786).

[8] A. Darby, M. Williams, A. Blakeborough, Stability and Delay Compensation for Real-Time Substructure Testing, Journal of Engineering Mechanics 128 (12) (2002) 1276–1284.

[9] M. Wallace, D. Wagg, S. 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 Sciences 461 (2064) (2005) 3807–3826. doi:10.1098/rspa.2005.1532.

[10] D. J. Wagg, D. P. Stoten, Substructuring of dynamical systems via the adaptive minimal control synthesis algorithm, Earthquake Engineering & Structural Dynamics 30 (6) (2001) 865–877.

[11] A. Plummer, Model-in-the-Loop Testing, Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 220 (3) (2006) 183–199. doi:10.1243/09596518JSCE207.

[12] D. Stoten, J. Tu, G. Li, Synthesis and control of generalized dynamically substructured systems, Systems and Control Engineering 223 (2010) 371–392.

[13] D. Stoten, G. Li, J. Tu, Model predictive control of dynamically substructured systems with application to a servohydraulically actuated mechanical plant, IET Control Theory & Applications 4 (2) (2010) 253–264.

[14] J. Tu, D. Stoten, R. Hyde, G. Li, A state-space approach for the control of multivariable dynamically substructured systems, Journal of Systems and Control engineering 225 (7) (2011) 935–953.

[15] G. Li, J. Na, D. Stoten, X. Ren, Adaptive neural network feedforward control for dynamically substructured systems, IEEE Transactions on Control Systems Technology 22 (3) (2014) 944–954. doi:10.1109/TCST.2013.2271036.

[16] R. Priesner, S. Jakubek, Mechanical impedance control of rotatory test beds, IEEE Transactions on Industrial Electronics 61 (11) (2014) 6264–6274. doi:10.1109/TIE.2014.2308159.

[17] G. Ou, S. Dyke, A. Prakash, Real time hybrid simulation with online model updating: An analysis of accuracy, Mechanical Systems and Signal Processing 84 (2017) 223–240. doi:10.1016/j.ymssp.2016.06.015.

[18] N. Hogan, S. Buerger, Impedance and Interaction Control, Springer Verlag, Berlin, 2005.

[19] W. Newman, Stability and Performance Limits of Interaction Controllers, Journal of Dynamic Systems, Measurement and Control 114 (583). doi:10.1109/TIE.2015.2448513.

[20] G. Zeng, A. Hemami, An overview of robot force control, Robotica 15 (5). doi:10.1017/S026357479700057X.

[21] V. Chabaud, S. Steen, R. Skjetne, Real-time hybrid testing for marine structures: challenges and strategies, in: ASME 2013 32nd International Conference on Ocean, Offshore and Arctic Engineering, American Society of Mechanical Engineers, 2013.

[22] T. Sauder, V. Chabaud, M. Thys, E. Bachynski, L. O. Sæther, Real-Time Hybrid Model Testing of Braceles Semi-Submersible Wind Turbine. Part I: The Hybrid Approach, in: Proceedings of OMAE 2016 35th International Conference on Ocean, Offshore and Arctic Engineering, 2016.

[23] V. Bayer, U. Dorka, U. Füllekrug, J. Gschwilm, On real-time pseudo-dynamic sub-structure testing: algorithm, numerical and experimental results, Aerospace Science and Technology 9 (3) (2005) 223–232. doi:10.1016/j.ast.2005.01.009.

[24] O. Ma, A. Flores-Abad, T. Boge, Use of industrial robots for hardware-in-The-loop simulation of satellite rendezvous and docking, Acta Astronautica 81 (2012) 335–347. doi:10.1016/j.actaastro.2012.08.003.

[25] H. Slater, D. Atkinson, A. Jack, Real-time emulation for power equipment development. II. The virtual machine, Electric Power Applications, IEE Proceedings - 145 (3) (1998) 153–158. doi:10.1049/ip-epa:19981849.

[26] M. Steurer, C. Edrington, M. Sloderbeck, W. Ren, J. Langston, A megawatt-scale power hardware-in-the-loop simulation setup for motor drives, IEEE Transactions on Industrial Electronics 57 (4) (2010) 1254–1260. doi:10.1109/TIE.2009.2036639.

[27] A. Facchinetti, S. Bruni, Hardware-in-the-loop hybrid simulation of pantograph-catenary interaction, Journal of Sound and Vibration 331 (12) (2012) 2783–2797. doi:10.1016/j.jsv.2012.01.033.

[28] W. Zhang, G. Mei, X. Wu, Z. Shen, Hybrid Simulation of Dynamics for the Pantograph-Catenary System, Vehicle System Dynamics 38 (6) (2010) 393–414.

[29] B. Cryer, P. Nawrocki, R. Lund, A Road Simulation System for Heavy Duty Vehicles, SAE Transactions 85 (1976) 1322–1334.

[30] C. Dodds, A. Plummer, Laboratory Road Simulation for Full Vehicle Testing: A Review, in: SAE Technical Paper, no. 2001, 2001, pp. 487–494. doi:10.4271/2001-26-0047.

[31] J.-M. Bras, Development of a Standardized Method for Actuator Characterization using Active Control of Impedance, Dissertation, Virginia Polytechnic Institute and State University (1999).

[32] T. Jungblut, S. Wolter, M. Matthias, H. Hanselka, Using Numerical Models to Complement Experimental Setups by Means of Active Control of Mobility, Applied Mechanics and Materials 70 (2011) 357–362.

[33] S. Kuo, D. Morgan, Active Noise Control Systems, John Wileys & Sons, New York, 1996.

[34] P. Ioannou, J. Sun, Robust adaptive control, Dover Publications, Mineola, 2013.

[35] A. Bartl, J. Mayet, D. Rixen, Adaptive Feedforward Compensation for Real Time Hybrid Testing with Harmonic Exciation., Proceedings of the 11th International Conference on Engineering Vibration (2015) 7–10.

[36] S. Haykin, Adaptive Filter Theory, Prentice Hall, Upper Saddle River, NJ, 2010.

[37] M. van der Seijs, D. Van Den Bosch, D. Rixen, D. De Klerk, An Improved Methodology for the Virtual Point Transformation of Measured Frequency Response Functions in Dynamic Substructuring, in: COMPDYN 2013, no. June, 2013, pp. 12–14. doi:10.13140/RG.2.1.2715.3126.

[38] R. Craig, C. Chang, Substructure Coupling for Dynamic Analysis and Testing, Tech. rep. (1977).

[39] E. Pasma, M. van der Seijs, S. Klaassen, M. van der Kooij, Frequency based substructuring with the virtual point transformation, flexible interface modes and a transmission simulator, in: IMAC-XXXVI: International Modal Analysis Conference, Orlando, FL, Society for Experimental Mechanics, Bethel, CT, 2018.

[40] V. Madisetti, Digital Signal Processing Fundamentals, Electrical Engineering Handbook, Taylor & Francis, Bosa Roca, 2009.