Fakultät für Maschinenwesen
Lehrstuhl für Angewandte Mechanik

Real-Time Hybrid Substructure Testing

Adaptive Approaches for Compliant Structures

Andreas Bartl

Vollständiger Abdruck der von der Fakultät für Maschinenwesen der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender: Prof. Dr.-Ing. habil. Boris Lohmann

Prüfer der Dissertation:

1. Prof. dr.ir. Daniel J. Rixen

2. Prof. Andrew Plummer, Ph.D.

Die Dissertation wurde am 27.06.2019 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 30.10.2019 angenommen.

Acknowledgement

This work was produced during my time from 2013 to 2019 at the Chair of Applied Mechanics at the Technical University of Munich. I would like to express my sincere thanks to the many people who have supported and promoted me during this time. My first thanks go to my PhD supervisor Prof. dr. ir. Daniel Rixen for the trust placed in me, the support and assistance. He gave me the opportunity to work on an exciting and multifaceted topic and provided me with advice and support. Both the intensive care and the technical support, as well as the large freedom for the development were and are for me of inestimable value.

My time at the Chair of Applied Mechanics will remain in the best of memories, above all because of my colleagues. I would like to thank them for the open and helpful atmosphere, the exciting exchange and the numerous and lively professional and nonprofessional discussions. In particular I would like to thank Alexander Ewald, Fabian Gruber, Arne-Christoph Hildebrandt, Oliver Hofmann, Morteza Karamooz Mahdiabadi, Johannes Maierhofer, Georg Mayr, Christina Insam, Johannes Mayet, Christian Meyer, Johannes Rutzmoser, Christoph Schütz and Daniel Wahrmann for their collegial and intensive cooperation in teaching and research.

I would also like to thank the students who have made a valuable contribution to the success of this work in the form of student research projects. Not only by the diligent work, but also by the discussions with the support there were always new thought impulses. I would especially like to thank Holger Hirschmann, Seyedbarzin Hosseini, Benjamin Kammermeier, Ron Reichart, Marcel Treis, Mathieu Wernsen and Kuiwei Zhang.

I would also like to thank Christina Insam, Christian Meyer and Carola Wieser for their valuable advice in reviewing the manuscript. Furthermore, I would like to thank Prof. Andrew Plummer and Jon du Bois for giving me the opportunity of a research stay at the Centre for Power Transmission & Motion Control at the University of Bath and Hashan Pereis for the fruitful collaboration at the University of Bath.

I would like to thank my family who have supported me in so many ways, especially my parents. My sincerest gratitude goes to my girlfriend Carola Wieser, who supported me in all matters. Without her patience and advice this work would be unthinkable.

Munich, June 24, 2019 *Andreas Bartl*

Abstract

The performance requirements of structural-dynamic systems are growing and generate the need for accurate testing methods. Real-time hybrid testing is an approach for assessing components of complex systems under realistic boundary conditions. In order to do so, systems are split into two parts: one part is challenging to model and serves as an experimental part in the hybrid test. The other part is subject to frequent changes in the design process or physically not existent and serves as the virtual component of the test. Virtual and experimental component are coupled using actuator and sensor system. Since the dynamics of the experimental component are unknown, real-time hybrid testing can pose a challenging control problem. The main goals of this thesis are to develop methods for the coupling of structures with high modal density and without prior system knowledge, to explore ways to efficiently implement the virtual component and to study methods for the accurate application of interface forces and displacements. In the context of simulating the virtual component, time integration schemes and model order reduction techniques are addressed. Finite impulse response filters are another option for a stable and efficient computation of the responses of the virtual component. Their use is analyzed in the thesis. Actuator systems are necessary to apply forces and displacements on the experimental component. They are shown based on two examples. Transmission simulators are structures which replicate the dynamics of interfaces. Force sensors cannot be placed directly at the interface if a transmission simulator is applied. For this reason, interface forces have to be estimated by removing the inertia effects of a transmission simulator from the measurements. Actuation system and delays in the control system have a significant influence on the stability and accuracy of the tests. A stability analysis is therefore performed based on a simplified model and characteristic mass, stiffness and damping ratios are discussed. The thesis examines the fundamental properties of systems with delays and actuator dynamics. Many mechanical configurations yield unstable tests without specific control algorithms. Adaptive feedforward filters offer the possibility to couple systems accurately while maintaining stability. In this work, methods based on least-mean-squares filters and based on recursive-least-squares filters are addressed. The methods do not require prior system knowledge and work for systems with interfaces with multiple degrees of freedom. Simulation results show the superior properties of the adaptation law which is based on recursive-least squares. Methods based on the power-flows in the actuation system help to simplify the choices of adaptation parameters. Experiments on a clamped beam test rig and a test rig with a cubic spring show the successful application of the method on a system with a two degree of freedom interface and on nonlinear systems. Finally, adaptive feedforward filters are combined with a feedback based control scheme. In order to stabilize the system, a passivity-based controller is augmented. Experiments on a non-linear system show that drift effects can be prevented and slow transient dynamics can be effectively coupled while keeping the accuracy properties of the adaptive feedforward filters.

Zusammenfassung

Die Anforderungen an strukturdynamische Systeme wachsen und erfordern genaue Testmethoden. Real-Time Hybrid Testing ist ein Ansatz zur Beurteilung von Komponenten komplexer Systeme unter realistischen Randbedingungen. Dabei wird das System in zwei Teile geteilt: Ein Teil ist schwierig zu modellieren und dient als experimenteller Teil im Hybridtest. Der andere Teil unterliegt häufigen Änderungen im Designprozess oder ist physisch nicht vorhanden und dient als virtuelle Komponente des Tests. Virtuelle und experimentelle Komponenten werden über Aktoren und Sensorsysteme gekoppelt. Da die Dynamik der experimentellen Komponente unbekannt ist, kann Real-Time Hybrid Testing ein anspruchsvolles Regelungsproblem darstellen. Die Hauptziele der Arbeit sind die Entwicklung von Methoden für die Kopplung von Strukturen mit hoher modaler Dichte und ohne Vorkenntnisse des Systems, die Erforschung von Möglichkeiten zur effizienten Implementierung der virtuellen Komponente und die Untersuchung von Methoden für die genaue Anwendung von Schnittstellenkräften und Schnittstellenverschiebungen. Im Rahmen der Simulation der virtuellen Komponente werden Zeitintegrationsschemata und Techniken zur Reduzierung der Modellordnung behandelt. Filter mit endlicher Impulsantwort können eine weitere Option für eine stabile und effiziente Berechnung der Antworten der virtuellen Komponente sein. Ihre Verwendung wird in der Arbeit analysiert. Aktorsysteme sind notwendig, um Kräfte und Verschiebungen auf die experimentelle Komponente auszuüben. Sie werden anhand von zwei Beispielen dargestellt. Transmission Simulators sind Strukturen, welche die Dynamik von Schnittstellen nachbilden. Kraftsensoren können nicht direkt an der Schnittstelle platziert werden, wenn ein Transmission Simulator eingesetzt wird. Aus diesem Grund müssen die Interfacekräfte geschätzt werden, indem die Trägheitseffekte eines Transmission Simulators aus den Messungen eliminiert werden. Die Aktordynamik und Verzögerungen im Regelsystem haben einen wesentlichen Einfluss auf die Stabilität und Genauigkeit der Tests. Eine Stabilitätsanalyse wird auf der Grundlage eines vereinfachten Modells durchgeführt und charakteristische Masse, Steifigkeit und Dämpfungsverhältnisse werden diskutiert. Die Dissertation untersucht die grundlegenden Eigenschaften von Systemen mit Verzögerungen und Aktordynamik. Ohne spezielle Regelalgorithmen ergeben viele mechanische Konfigurationen instabile Tests. Adaptive Feedforward-Filter bieten die Möglichkeit, Systeme bei gleichbleibender Stabilität exakt zu koppeln. In dieser Arbeit werden Methoden auf der Grundlage von Least-Mean-Squares Filtern und auf der Grundlage von Recursive-Least-Squares Filtern behandelt. Die Methoden erfordern keine Vorkenntnisse über das System und funktionieren für Systeme mit Schnittstellen mit mehreren Freiheitsgraden. Simulationsergebnisse zeigen die herausragenden Eigenschaften des Anpassungsgesetzes, das auf rekursiven kleinsten Quadraten basiert. Methoden, die auf den Leistungsflüssen im Aktorsystem basieren, helfen, die Auswahl der Adaptionsparameter zu vereinfachen. Experimente an einem Balkenprüfstand und einem Prüfstand mit einer kubischen Feder zeigen die erfolgreiche Anwendung des Verfahrens auf ein System mit einer Schnittstelle mit zwei Freiheitsgraden und auf nichtlineare Systeme. Abschließend werden adaptive Feedforward-Filter mit einem rückgekoppelten Regelverfahren kombiniert. Um das System zu stabilisieren, wird eine passivitätsbasierte Regelung ergänzt. Experimente an einem nichtlinearen System zeigen, dass Drifteffekte verhindert werden können und dass langsame transiente Dynamiken effektiv gekoppelt werden können, während die Genauigkeitseigenschaften der adaptiven Feedforward-Filter erhalten bleiben.

Contents

Nomenclature

Operators

$(\dot{\cdot})$	...	time derivative
$(\ddot{\cdot})$	...	second time derivative
$(\cdot) * (\cdot)$	...	convolution
$(\cdot)'$	...	first derivative
$(\cdot)''$	...	second derivative
$(\cdot)^{(n)}$	...	n-th derivative

$(\cdot)^T$	...	transpose
$(\cdot)^{-1}$	...	inverse
Δ	...	difference
$\mathcal{F}(\cdot)$	...	Fourier transform
$\mathcal{L}(\cdot)$	...	Laplace transform

Subscripts

$(\cdot)_\lambda$	...	interface force
$(\cdot)_{ext}$	...	external excitation
$(\cdot)_{nl}$	...	non-linear

$(\cdot)_u$	...	actuator input
${}_X(\cdot)$	...	object in coordinate system X
${}_X(\cdot)_Y$	...	transformation from system Y to system X

Superscripts

$(\cdot)^*$	...	predictor
$(\cdot)^{ACT}$	...	actuator system
$(\cdot)^{approx}$	...	approximated
$(\cdot)^{est}$	...	estimated

$(\cdot)^{EXP}$	...	experimental component
$(\cdot)^{filt}$	...	filtered
$(\cdot)^{TR}$	...	test rig
$(\cdot)^{VIR}$	...	virtual component

Greek Letters

α	...	HHT-α parameter
β	...	Newmark parameter
γ	...	Newmark parameter
$\gamma_{(\cdot)}$	...	regularization factor
δ	...	Dirac impulse
ζ	...	damping ratio
κ	...	stiffness ratio
λ	...	interface force

μ	...	adaptation gain
ν	...	leakage factor
σ	...	step function
ϕ	...	mass ratio
τ	...	delay ratio
ω	...	angular frequency
Λ	...	Laplace domain interface force
Ω	...	angular excitation frequency

Boldface Greek Letters

$\boldsymbol{\lambda}$	...	interface force vector	$\boldsymbol{\Phi}$	...	regression matrix
$\boldsymbol{\theta}$	...	parameter vector	$\boldsymbol{\Psi}$	...	amplitude and phase vector
$\boldsymbol{\Xi}$	...	confidence vector			

Latin Letters

a	...	adaptation variable	u	...	actuator input
b	...	adaptation variable	r	...	random number
c	...	noise amplitude	t	...	time
d	...	damping constant	y	...	interface displacement
f	...	force	A	...	amplitude
g	...	interface gap	B	...	magnetic flux
k	...	stiffness	E	...	energy
$i_{(\cdot)}$	...	current	F	...	Laplace domain force
i	...	index	H	...	transfer function
j	...	imaginary	J	...	cost function
m	...	mass	L	...	inductance
$k_{(\cdot)}$	...	stiffness	N	...	number
k	...	index	M	...	number of FIR coefficients
n	...	number	P	...	power
m	...	number	R	...	resistance
p	...	noise term	T	...	time period

Boldface Latin Letters

$\boldsymbol{a}$	...	joint position vector	$\boldsymbol{D}$	...	damping/feed-through matrix
$\boldsymbol{b}$	...	leg length vector	$\boldsymbol{G}$	...	interface matrix
$\boldsymbol{e}$	...	error vector	$\boldsymbol{H}[n]$	...	impulse response matrix
$\boldsymbol{f}$	...	force vector	$\boldsymbol{H}(s)$	...	transfer function matrix
$\boldsymbol{g}$	...	interface gap vector	$\boldsymbol{H}(t)$	...	impulse response matrix
$\boldsymbol{o}$	...	joint position vector	$\boldsymbol{H}(\omega)$	...	transfer function matrix
$\boldsymbol{q}$	...	coordinate vector	$\boldsymbol{I}$	...	identity matrix
$\boldsymbol{u}$	...	input vector	$\boldsymbol{K}$	...	stiffness matrix
$\boldsymbol{s}$	...	regression vector	$\boldsymbol{M}$	...	mass matrix
$\boldsymbol{o}$	...	tool-center point position vector	$\boldsymbol{Q}$	...	rotation matrix
$\boldsymbol{x}$	...	state vector	$\boldsymbol{R}$	...	reduction matrix
$\boldsymbol{y}$	...	interface displacement vector	$\boldsymbol{P}$	...	real valued transfer matrix
$\boldsymbol{A}$	...	system matrix	$\boldsymbol{U}$	...	input signal vector
$\boldsymbol{B}$	...	input matrix	$\boldsymbol{W}$	...	basis function matrix
$\boldsymbol{C}$	...	output matrix	$\boldsymbol{Y}$	...	interface displacement vector

Abbreviations

CMS	...	Component Mode Synthesis	IBS	...	Impulse Based Substructuring
DAE	...	Differential Algebraic Equation	IIR	...	Infinite Impulse Response
FBS	...	Frequency Based Substructuring	LMS	...	Least Mean Squares
FEM	...	Finite Element Method	LQR	...	Linear Quadratic Regulator
FIR	...	Finite Impulse Response	MEMS	...	Micro-Electro-Mechanical System
FRF	...	Frequency Response Function	ODE	...	Ordinary Differential Equation
fxLMS	...	filteres-x Least Mean Squares	PID	...	Proportional-Integral-Derivative
HiL	...	Hardware-in-the-Loop	RLS	...	Recursive Least Squares

Chapter 1

Introduction

Parts of this chapter have been submitted for publication in [7] and [9].

The role of structural dynamics in engineering is dominated by the demand for product quality, costs, and efficiency as well as by environmental considerations. Lighter structures require less material and lower life-cycle energy consumption and costs. This point is especially critical in the aerospace industry, where energy efficiency and costs have always been important innovation drivers. Other examples of industries where structural dynamics play a fundamental role include wind turbine engineering and micro-electro-mechanical systems (MEMS). In the wind energy market, costs and harsh offshore conditions require more and more advanced structural designs. The monolithic design of micro-electro-mechanical systems represents a classical structural dynamic problem. In the automotive industry, noise and vibration—which are strongly influenced by structural dynamics—are a primary quality measure.

The key to the design of better structures is the prediction of their behavior and tests. Hence, the high demands within the development process and for the end product require advanced methods for simulation, experimental parameter identification, and tests. Simulations are often efficient, fast and cheap and their prediction quality is improving. Significant research effort has been put into the development of methods which increase the prediction precision and accuracy of simulation models and reduce simulation time. The V-model in Fig. 1.1 visualizes the mechatronic development process (see [50]). In the first step, requirements are defined on the system level. Simulation models help to specify the requirements on the component level. After the design process, prototypes are tested on the component and the system level. Tests are an essential part of the development process of mechatronic systems. Since physical testing is often costly and time-consuming, it is desirable to replace tests with simulations. However, a simulation cannot always capture the complexity which is inherent to the physical system: Structural tests stay an essential part of the development process in many engineering fields as e.g. in aerospace, biomedical, or automotive engineering.

One way to keep costs and complexity of structural tests 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. This fact seems obvious—as shown in the example in Fig. 1.3—but is often neglected in practice. 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 the experimental component—and a simulated part—referred to as the virtual component in this work. An actuation system is controlled such that both components are coupled and the overall system behavior is imitated. This paradigm is referred to as hybrid testing.

The advantages of such approaches are numerous: Frequent changes in the virtual component during the development process can be flexibly integrated into tests. Boundary conditions which cannot be reproduced otherwise—such as aerodynamic or fluid interaction forces—can be replicated by actuators. Components which 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 which do not originate from the

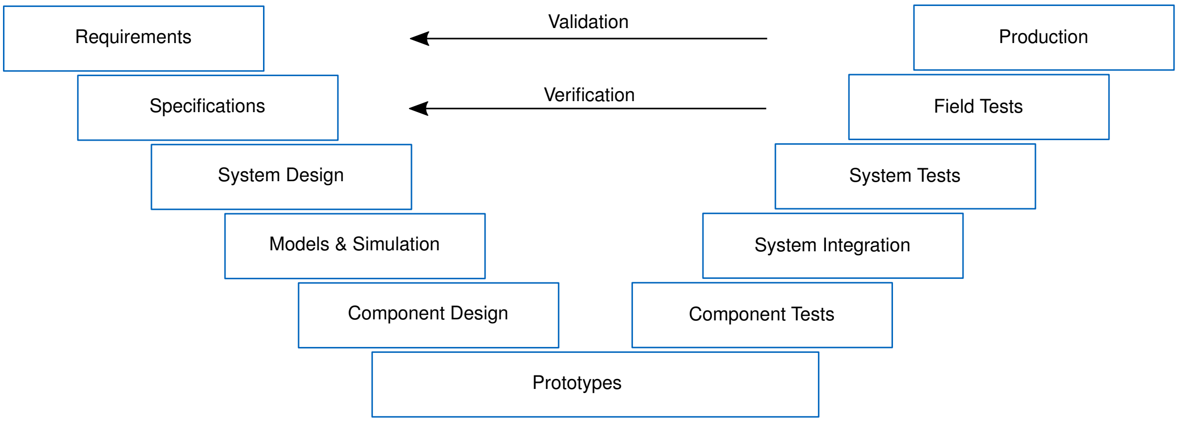


Figure 1.1: The V-model of the development process in mechatronics (adapted from [50]).

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. To summarize, reasons to perform a test rather than a computer simulation include:

- detection of unseen hardware defects
- physical effects which are not modeled in a simulation model
- physical effects which are difficult to model
- assessment of haptic behavior and component such as a sound source

The need for a hybrid testing approach can arise from the following applications:

- design optimization of a virtual component with frequent changes
- test of components of a physically non-existent system
- test with complex boundary conditions such as aerodynamic or fluid interaction forces
- test of a component of a large size structure
- controlled boundary conditions and repeatable tests
- operation point dangerous on system level

Example The idea of a hybrid test is illustrated in Fig. 1.2 for a simple lumped-mass system. Both, the virtual component (drawn in blue in Fig. 1.2) and the experimental component (drawn in green in Fig. 1.2) 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. 1.2) is represented by a mass-spring-damper system and the actuation force $\boldsymbol{f}_{act}$. A hybrid test aims to control $\boldsymbol{f}_{act}$ and $\boldsymbol{\lambda}^{VIR}$ such that the motion of the virtual component and the experimental system simulates the behavior of the coupled overall system (green and blue component). All properties, which were used for the numerical experiments below, are listed in Tab. B.1. Bode plots of the dynamics of the coupled reference system, the virtual component, and the experimental component are shown in Fig. 1.3.

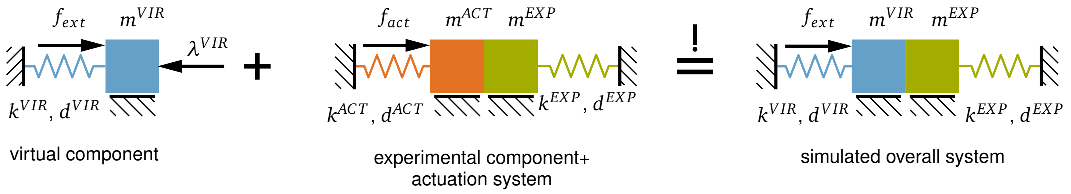


Figure 1.2: Lumped-mass system used for numerical investigations

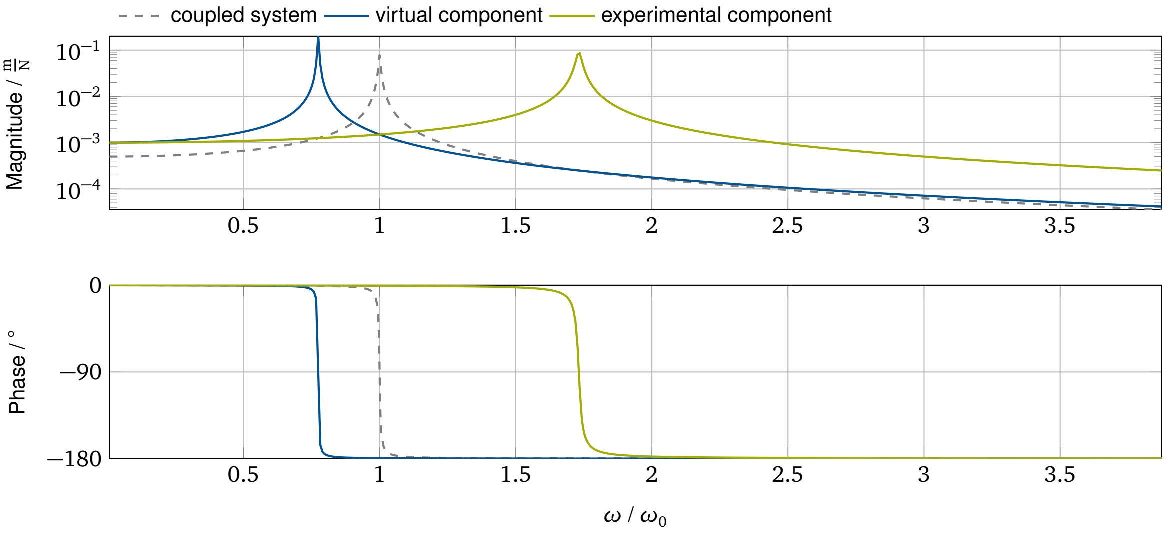


Figure 1.3: Bode plots of coupled and uncoupled dynamics of two mass-spring-damper systems. The angular frequency is normalized with the resonance frequency ω_0 of the reference system.

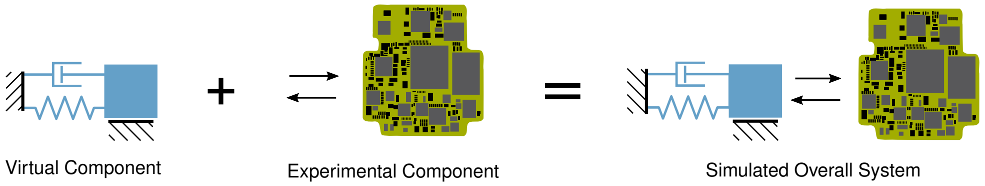


Figure 1.4: Concept scheme of the Hardware-in-the-Loop approach

1.1 Related Methods & Technology

The paradigm of coupling simulations and physical systems is related to several methods: Hardware-in-the-Loop testing, real-time hybrid testing, pseudo-dynamic testing, interaction control in robotics and dynamic substructuring are all techniques which allow combining simulation models with physical systems. An overview of the existing methods is given in Tab. 1.1.

1.1.1 Hardware-in-the-Loop

In the context of the development of mechatronic systems, Hardware-in-The-Loop (HiL) is a well-known approach, which couples simulation and hardware tests. The HiL approach is used to test controller hardware and software while the environment and hardware components such as mechanical components, hydraulic systems, sensors, and actuators are simulated in real-time. The concept is visualized in Fig. 1.4. Two methods related to HiL are Software-In-The-Loop (SiL) and Rapid-Control-Prototyping (RCD). In the Software-In-The-Loop approach, both controller software and environment are simulated while no hardware is present. This approach can serve to evaluate the system behavior and to allow changes early in the design process. Rapid-Control-Prototyping allows testing the control algorithms on the actual mechatronic system using general purpose real-time computers. The method is applied when the controller hardware is yet not available. An overview of the methods is given e.g. in [51]. HiL simulations are widely used in the automotive and the aerospace industry. 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 the interface between experimental and virtual component. Hence, the dynamics of the transfer or the actuation system do not deteriorate the test performance.

1.1.2 Pseudo Dynamic Testing

Pseudo-dynamic testing is a method which couples simulations and experiments within an extended time-scale. All time-dependent effects—as inertia or damping—are neglected or simulated using a model, while stiffness effects are measured in an experiment. This approach is feasible in applications where the modeling of the inertia and damping effects is uncomplicated, where the models can be reduced to a small number of degrees of freedom and where the stiffness effects are hard to model. The concept of pseudo-dynamic testing is visualized in Fig. 1.5: All displacements are calculated by time integration of the equations of motion. In contrast to a full simulation, the stiffness effects are retrieved directly from the experimental component. In order to do so, the calculated displacements are applied to the experimental component, and the restoring forces are measured. The structure under test has to be condensed to a lumped-mass system since each degree of freedom in the equations of motion has to be actuated as described in [31]. The method allows the analysis of effects such as crack propagation or other damages in structures. Since the test does not run in real-time, it can be stopped at any time step in order to evaluate the resulting modifications of the structure. The extended time-scale also means that stability issues caused by actuator delays do not occur. The concept of pseudo-dynamic testing has been first introduced by [103].

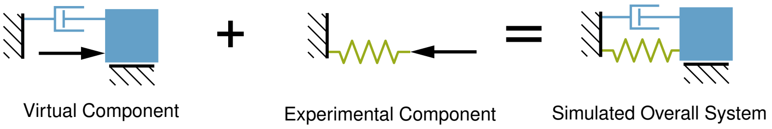


Figure 1.5: Concept scheme of the pseudo-dynamic testing approach

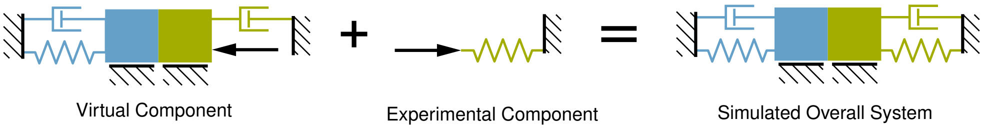


Figure 1.6: Concept scheme of the pseudo-dynamic substructure testing approach

[21] and [76] describe 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. A scheme of the concept is given in Fig. 1.6. Applications of the pseudo-dynamic tests can be found in civil engineering where structures were exposed to earthquake loads. One example are the facilities in the ELSA lab—as described by [76]—where test specimens up to the size of $10\ \mathrm{m}$ are tested against a reaction wall. More specifically, [76] reports on the test of piers of a bridge coupled to the simulated overall bridge-structure. Pseudo-dynamic tests are used where the dynamics of the actuators do not allow real-time tests: [16] describes an application of pseudo-dynamic testing in the aerospace industry where an air-to-air refueling process is simulated with a hybrid test. In this work, the probe-drogue impacts and contact are replicated using an industrial robot. One limit of application for the pseudo-dynamic testing approach lies in the fact that the rate dependent effects still have to be modeled for the simulation. This procedure is sometimes impossible and conflicts with the objective of hybrid tests, namely that mechanical properties of the experimental component are unknown.

1.1.3 Real-Time Hybrid Testing

Real-time hybrid testing is a method which allows implementing realistic dynamical tests of components of complex systems. The system is split into a virtual system—which is simulated in real-time—and an experimental component which is physically present in the test. The limitations of Pseudo-Dynamic testing are overcome by applying all forces in real-time and, hence, capturing also rate-dependent effects—such as damping and inertia—in the experimental component. Among the first applications that have been reported are publications by [68], [45] and [14]. In literature, the method is also referred to as real-time hybrid simulation, real-time hybrid substructure testing, Model-in-the-Loop testing or real-time substructuring.

Applications of the method have been reported in various fields such as in aerospace, automotive, civil and mechanical engineering. To name just a few examples: Satellite tests using hybrid approaches are proposed in [13], hybrid testing of off-shore structures is described in [24] and [79] proposes the tests on chassis dynamics of cars in combination with aerodynamic simulations.

The objective of any real-time hybrid testing technology is to test the experimental component under realistic boundary conditions. In order to do so, the dynamics of the virtual component are coupled to the test rig by using an actuation system. Actuators apply forces or displacements to the interface of the experimental component, while sensors measure interface forces and interface displacements. In contrast to standard HiL tests, most real- time hybrid testing setups make time lags and delays inevitable. The dynamics of the actuators cause frequency-dependent time lags and change due to 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. Other delay contributions are caused by the

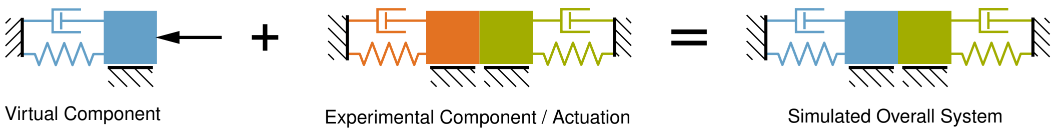


Figure 1.7: Concept scheme of the real-time hybrid testing approach

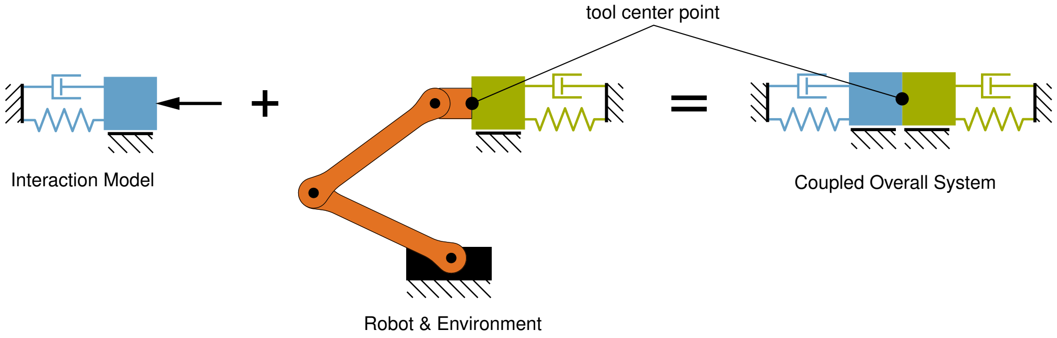


Figure 1.8: Concept scheme of interaction control in robotics

computational and communication processes as well as by signal-processing procedures. Time lags and delays can cause instability and deterioration in the test accuracy if they occur in a test without a further compensation technique. In order to tackle these stability problems, several methods have been proposed in hybrid-testing literature. [45] suggests using a polynomial forward prediction scheme which compensates for delay and amplitude errors. Based on this work, [109] introduced an additional adaptation scheme which tunes phase shifts and amplitude corrections according to errors at the zero crossings. Other methods for delay compensation and interface synchronization include the application of model reference adaptive control by [108], inverted models of the actuation system by [15] and model predictive control by [101]. [78] applies a passivity-based control method to real-time hybrid testing.

1.1.4 Interaction Control in Robotics

The interaction control in robotics poses a problem, which is similar to real-time hybrid testing. The objective of this type of robot control is not to follow defined forces or motions, but rather, to provide a certain impedance or admittance at the tool center point. This objective 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 [44]. Admittance determines velocity output as a result of force input. Admittance control stability issues were investigated by [69]. [110] gives 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.

1.1.5 Dynamic Substructuring

Dynamic substructuring is the generic term for methods using domain decomposition and substructure coupling in the domain of structural dynamics (Fig. 1.9). Although dynamic substructuring, in general, does not include methods which couple tests and simulation in real-time, its philosophy of coupling simulations and experiments exhibits close links to hybrid testing. Using the dynamic substructuring

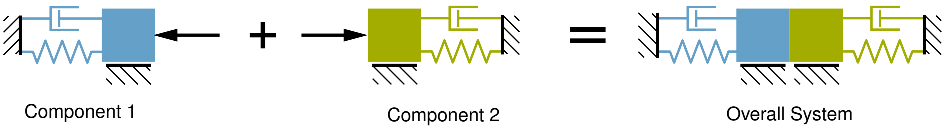


Figure 1.9: General concept scheme of the dynamic substructuring strategy

strategy, complex systems are split into substructures. Those substructures can be modeled, measured and analyzed separately.

Dynamic substructuring is an essential strategy in structural engineering since it facilitates simulation and experimental work in many ways: It enables local optimization of substructures, while structural behavior which does not influence the global dynamics is not accounted for on the global level. The philosophy of decomposing complex structures into smaller substructures also simplifies collaboration between various contributors. Dynamic substructuring can accelerate the solution of numerical models by parallelization and model order reduction. On the experimental side, the complexity of the measurements can be reduced by applying substructuring methods. The approach allows combining simulation models and models which were identified from experiments.

A framework for the classification of dynamic substructuring methods is presented e.g. in [30]. Usually, the complexity of substructure models which originate from numerical simulation models can be reduced by using model order reduction. A common technique for model order reduction of substructures used in many commercial codes is the method presented by [25]. It is referred to as the Craig-Bampton method. Other reduction techniques include the methods of [64], [91] and [90]. Experimental models can be modal models or can consist of impulse responses or frequency response functions (FRF). Modal models are retrieved by identifying the modal parameters such as modal damping, masses and resonance frequencies of a structure from measurements. The frequency response functions represent the dynamic relationship between force input and sensor output at each frequency. In order to analyze the global dynamic behavior, the substructures are numerically coupled in the final step. The coupling can be performed directly in the physical domain, in the modal domain, in the time domain or in the frequency domain. In experimental substructuring, usually, the responses to interface excitations have to be measured. The methods which make use of the modal domain are referred to as component mode synthesis (CMS). Component mode synthesis is preferable if the modes of the structure are clearly separated. Impulse-based substructuring (IBS) was proposed in [88]. It is suitable if impulses can describe the substructure responses. A frequency-based substructuring (FBS) method—as proposed by [52]—can be used if the substructures exhibit high modal densities. Since most experimental substructuring techniques work with linear models, they cannot represent parameter and state-dependent behavior. Real-Time Hybrid Testing, in turn, is a substructuring technique which extends the coupling approach to real-time tests, the main advantage, compared to classical substructuring, being that no model is required for the experimental part.

1.2 Application Cases

Applications of real-time hybrid testing exist in various fields of engineering such as aerospace, marine, biomedical, civil, rail or automotive engineering. Many methods are highly dependent on their application. As mentioned above, testing in earthquake engineering was one of the first applications of real-time hybrid testing. [68], [45] and [14] were among the first to report results in this area.

[24] and [92] 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.

[13] describes satellite testing scenarios. [62] proposes control strategies for contact simulation of space-

	Dynamics of experimental component simulated	Dynamics of experimental component measured	Actuator dynamics relevant
Hardware-in-the-Loop	-	●	-
Real Time Hybrid Testing	-	●	●
Pseudo Dynamic Testing	●	-	-
Interaction control in Robotics	-	●	●
Dynamic Substructuring	-	●	-

Table 1.1: Overview on methods related to hybrid testing

craft docking procedures. Automotive applications are proposed in [79]: 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. [98] proposes to test inverter hardware with virtual motors and associated mechanical loads. Megawatt-scale motor drives are tested using virtual gas turbine generator systems in [99]. [80] couples experimental components such as combustion engines and models of elastic drive-train dynamics. The test allows optimization of the coupled dynamics to reduce unwanted oscillations.

Friction forces, hydraulic valve dynamics, and unmodeled flexibilities make testing railway vehicle pantographs necessary. [33] and [112] propose real-time hybrid testing techniques as a replacement for expensive test drives: The pantograph is tested in the laboratory while a contact wire, a messenger wire, droppers, and suspensions are simulated.

1.3 Project Objectives

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. The need for a hybrid testing approach emerges from the fact that off-line substructuring techniques do not necessarily account for the correct amplitudes and do not measure at the correct working point.

Fig. 1.10 gives a potential example of an application scenario of a hybrid test with a structural dynamic system: The objective of the test is to analyze the influence of different designs of drive trains and transmission cross beams on the vibration behavior of a car. The body in white is a complex structure which is hard to model, while the transmission cross beam and the drive train are only available as models and their design frequently changes during the design process. A hybrid testing approach allows to couple the physical body in white to the model of the transmission system. Applications where structural systems are coupled often exhibit a high model density and low damping.

When coupling lightly damped structures, the poles of the coupled overall system often are close to instability. Complex and unknown dynamics make the design of hybrid testing controllers a difficult task. The application of a feedback-based methodology using low-order models remains challenging. Adaptive feedforward filters offer an alternative in this field. Instead of closing the control loop directly, feedforward filters generate the actuator input using the external excitation forces as an input signal. Since the structure of the filter is not known beforehand, information on the interface gap is used in

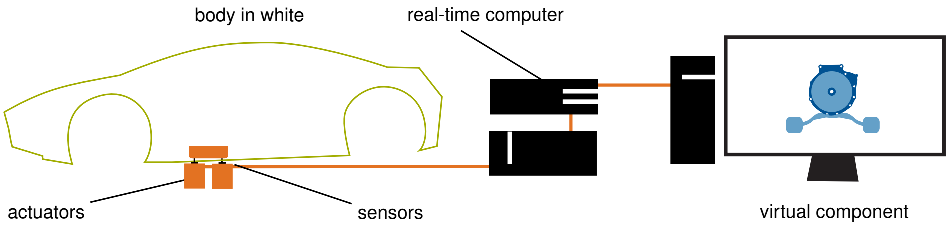


Figure 1.10: Application case: The virtual component can be optimized concerning its effect on the body-in-white's vibrations.

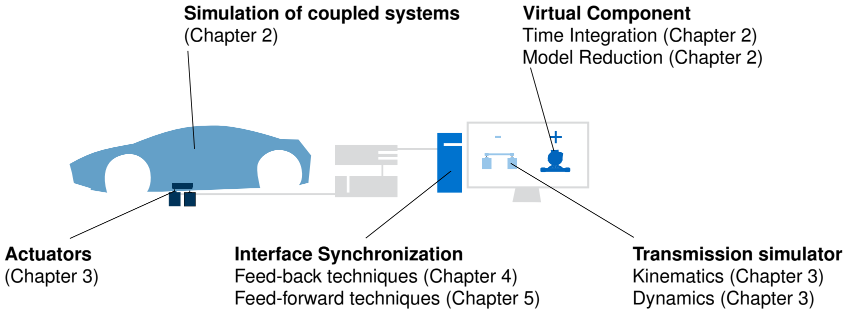

Simulation of coupled systems
(Chapter 2)
Virtual Component
Time Integration (Chapter 2)
Model Reduction (Chapter 2)
Actuators
(Chapter 3)
Interface Synchronization
Feed-back techniques (Chapter 4)
Feed-forward techniques (Chapter 5)
Transmission simulator
Kinematics (Chapter 3)
Dynamics (Chapter 3)

Figure 1.11: Overview on the topics considered in the thesis

an update-law in order to adapt the filter coefficient such that virtual and experimental component are coupled. The test dynamics are inherently more stable because the feedforward filters do not shift the poles of the system.

The adaptation allows the filters to be used without prior system knowledge. The objective of this thesis is to explore and discuss the possibilities of applying real-time hybrid testing to structural dynamic systems. More specifically, the following aspects are investigated:

- efficient ways to implement the virtual component
- accurate measurement of interface forces and states
- interface synchronization for systems with high modal density without prior system knowledge

1.4 Thesis Outline

In Chap. 2, the real-time hybrid testing problem is formulated mathematically using various model representations. Basic time integration schemes and model order reduction techniques used for implementing the virtual component are discussed. Chap. 3 explores ways to measure forces at the interface between the subcomponents and to estimate interface states. Kinematic aspects of actuation systems are described. Chap. 4 discusses feedback based interface synchronization techniques. It includes the state of the art methods for interface synchronization in real-time hybrid testing. Furthermore, the chapter analyses the effects of delays on the stability and accuracy of real-time hybrid testing. Chap. 5 investigates adaptive feedforward filters as a robust way to synchronize the interfaces. Possibilities to combine feedback and feedforward-based approaches and methods are analyzed. Additionally, the chapter discusses methods to improve the performance of adaptive feedforward filters in real-time hybrid testing. Chap. 7 concludes the thesis and discusses recommendations for future research.

1.5 Original Contributions

The following aspects of the thesis are the original scientific contributions from the author:

- A framework for real-time hybrid testing using adaptive feedforward filters with harmonic basis functions (published in [4] and[7])
- The application of recursive least-squares adaptation laws enabling coupling of multiple degree-of-freedom interfaces (published in [8], [7] and [3])
- The application of the method to an experimental clamped beam system with results over a frequency range up to 800Hz (published in [7])
- A technique to adjust the adaptation gain of adaptive feedforward filters based on power flows between the actuation system and the subcomponents (published in [9])
- The investigation of the comparison between feedback and feedforward techniques
- Discussion of the use of finite impulse response filters to simulate the virtual component
- An approach for applying and measuring forces and displacements at the interface between the subcomponents (published in [10] and [5])

Chapter 2

Simulation models and the virtual sub-component

Parts of this chapter have been submitted for publication in [7] and [9].

Three subsystems are part of any hybrid test: The experimental component is the subsystem under test. The actuation system is necessary in order to apply forces and displacements on the experimental component and to measure the responses of the experimental component. The virtual component is the part of the system which is simulated during the test. For the prediction of stability and accuracy of a test and for the evaluation of different methods, it is important to represent all three subsystems as mathematical models. In the first part of the chapter, the dynamics of the subcomponents and coupling constraints are listed in different representations. Since, for the virtual part, an efficient calculation of the response is important, in the second and third part of the chapter basic time integration and model reduction techniques are shortly described. The last part of the chapter discusses the use of time discrete filters for the simulation of the virtual component. The methods for this project were also analyzed in the student theses [19] and [67].

2.1 Model Representations

This thesis focuses on mechanical systems which can be described in the form of Eq. (2.1). Their equations of motion can originate from the discretization of the continuous mechanic field equations or from the mathematical representation of a multi-body system. Continuous mechanical systems are flexible structures which can be described by partial differential equations. The space-discretization of those partial differential equations is performed using techniques such as the finite element method (FEM). For details on the derivation and the mathematical formulation of the finite element method, the reader is referred to literature as e.g. [47], [11] and [113]. Multi-body systems consist of multiple, mechanically interacting, undeformable or deformable bodies. Their mathematical formulation is described in detail e.g. in [95], [77] and [12]. All these modeling techniques result in a set of second-order ordinary differential equations (ODEs).

Additional to the second-order equations of motion, the following section lists other representations of the subsystems' dynamics which are made use of in this thesis: The first-order state-space form is useful in the context of control engineering. The response of a linearized system can be represented as a convolution integral. The transfer function representation in the Laplace domain is useful for the analysis of the control approaches in Chap. 4 and the formulation of the adaptive feedforward approach of Chap. 5.

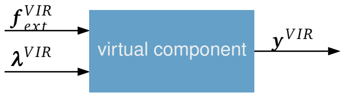


Figure 2.1: Block diagram of the virtual component.

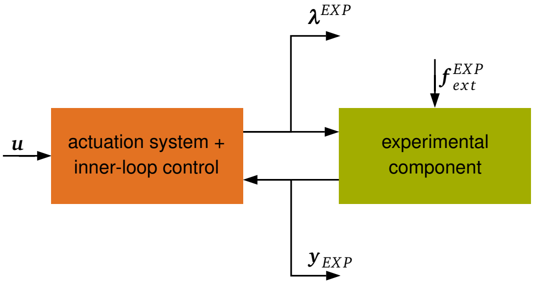


Figure 2.2: Block diagram of the interaction between actuation system and experimental component.

2.1.1 Second-Order Ordinary Differential Equations

The second-order form is the direct result of the mathematical description of any mechanical system. The dynamics of the virtual component (Fig. 2.1) with the system coordinates $\boldsymbol{q}^{VIR}$ are defined by Eqs. (2.1)

$$\begin{aligned} &\boldsymbol{M}^{VIR}(\boldsymbol{q})\ddot{\boldsymbol{q}}^{VIR}+\boldsymbol{f}_{nl}^{VIR}\left(\boldsymbol{q}^{VIR},\dot{\boldsymbol{q}}^{VIR}\right)=\left(\frac{\partial\boldsymbol{\Gamma}^{VIR}}{\partial\boldsymbol{q}^{VIR}}\right)^{T}\boldsymbol{\lambda}^{VIR}+\boldsymbol{f}_{ext}^{VIR} \\ &\boldsymbol{y}^{VIR}=\boldsymbol{\Gamma}^{VIR}\left(\boldsymbol{q}^{VIR}\right) \end{aligned} \tag{2.1}$$

where the mass matrix is denoted by $\boldsymbol{M}^{VIR}$, the internal non-linear forces by $\boldsymbol{f}_{nl}^{VIR}$, the external forces by $\boldsymbol{f}_{ext}^{VIR}$, and the function $\boldsymbol{\Gamma}^{VIR}\left(\boldsymbol{q}^{VIR}\right)$ maps the system coordinates to the interface displacements $\boldsymbol{y}^{VIR}$. Accordingly, the Jacobian matrix $\left(\frac{\partial\boldsymbol{\Gamma}^{VIR}}{\partial\boldsymbol{q}^{VIR}}\right)^{T}$ projects the interface forces $\boldsymbol{\lambda}^{VIR}$ on the system coordinates. The linearized form of Eq. (2.1) is

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

where the stiffness matrix is denoted by $\boldsymbol{K}^{VIR}$, the damping matrix is denoted by $\boldsymbol{D}^{VIR}$, and the mapping matrix is denoted by $\boldsymbol{G}^{VIR}$. The equations of motion can be solved for example using a Newmark time integration scheme as it is described in Sec. 2.3.

Actuation system and experimental component form a coupled system during the test (Fig. 2.2). At 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. (2.2) defines the dynamics of the experimental component with the system coordinates $\boldsymbol{q}^{EXP}$. Eq. (2.3) defines the dynamics of the actuators with the system coordinates $\boldsymbol{q}^{ACT}$.

$$\boldsymbol{M}^{EXP}\ddot{\boldsymbol{q}}^{EXP}+\boldsymbol{f}_{nl}^{EXP}\left(\boldsymbol{q}^{EXP},\dot{\boldsymbol{q}}^{EXP}\right)=\boldsymbol{f}_{ext}^{EXP}+\boldsymbol{G}^{EXP^{T}}\boldsymbol{\lambda}^{EXP} \tag{2.2}$$

$$\boldsymbol{M}^{ACT}\ddot{\boldsymbol{q}}^{ACT}+\boldsymbol{f}_{nl}^{ACT}\left(\boldsymbol{q}^{ACT},\dot{\boldsymbol{q}}^{ACT}\right)=\boldsymbol{B}^{ACT}\boldsymbol{u}^{ACT}+\boldsymbol{G}^{ACT^{T}}\boldsymbol{\lambda}^{EXP} \tag{2.3}$$

In these equations, the system matrices are defined correspondingly to the virtual component. $\boldsymbol{u}^{ACT}$ is the vector of the actuator input signal. $\boldsymbol{\lambda}^{EXP}$ 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 Chap. 3.
Linearization yields Eq. (2.4) and Eq. (2.5):

$$\boldsymbol{M}^{EXP}\ddot{\boldsymbol{q}}^{EXP}+\boldsymbol{D}^{EXP}\dot{\boldsymbol{q}}^{EXP}+\boldsymbol{K}^{EXP}\boldsymbol{q}^{EXP}=\boldsymbol{f}_{ext}^{EXP}+\boldsymbol{G}^{EXP^T}\boldsymbol{\lambda}^{EXP} \tag{2.4}$$

$$\boldsymbol{M}^{ACT}\ddot{\boldsymbol{q}}^{ACT}+\boldsymbol{D}^{ACT}\dot{\boldsymbol{q}}^{ACT}+\boldsymbol{K}^{ACT}\boldsymbol{q}^{ACT}=\boldsymbol{B}^{ACT}\boldsymbol{u}^{ACT}+\boldsymbol{G}^{ACT^T}\boldsymbol{\lambda}^{EXP} \tag{2.5}$$

The compatibility constraint between actuation system and experimental component is defined by

$$\boldsymbol{G}^{ACT}\boldsymbol{q}^{ACT}-\boldsymbol{G}^{EXP}\boldsymbol{q}^{EXP}=\boldsymbol{0}. \tag{2.6}$$

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

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

The system, consisting of actuation system and experimental component, does not have to be solved in real-time during the test. For the purpose of simulated tests, the equations for the experimental component and the actuators can be solved e.g. with a HHT-α method for time integration with constraints as it is described in Sec. 2.3.
In order to couple the experimental and the virtual component equilibrium constraints and compatibility constraints have to be met. It is the task of the control law to synchronize the interface of virtual and experimental component, and to enforce the compatibility constraint. The compatibility constraint is defined by (2.7).

$$\boldsymbol{y}^{EXP}=\boldsymbol{y}^{VIR} \tag{2.7}$$

If the constraint is not met, an interface gap $\boldsymbol{g}(t)$ remains. It defines the synchronization error in the test:

$$\boldsymbol{g}=\boldsymbol{y}^{EXP}-\boldsymbol{y}^{VIR} \tag{2.8}$$

The equilibrium constraint is defined by Eq. (2.9).

$$\boldsymbol{\lambda}^{VIR}=-\boldsymbol{\lambda}^{EXP} \tag{2.9}$$

2.1.2 State-Space Form

The state-space form is a widely used representation of dynamical systems in the control engineering community since it allows for the use of various analysis and design techniques. The state-space representation is defined by a first order ordinary differential equation (ODE) and an output equation. Eqs. (2.10) describe the virtual component's dynamics in state-space form making use of the system matrices from the second order form of Sec. 2.1.1

$$\begin{gathered}\dot{\boldsymbol{x}}^{VIR}=\underbrace{\begin{bmatrix}\boldsymbol{0} & \boldsymbol{I}\\ -\boldsymbol{M}^{VIR^{-1}}\boldsymbol{K}^{VIR} & -\boldsymbol{M}^{VIR^{-1}}\boldsymbol{D}^{VIR}\end{bmatrix}}_{\boldsymbol{A}^{VIR}}\boldsymbol{x}^{VIR}+\underbrace{\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{M}^{VIR^{-1}}\boldsymbol{G}^{VIR^T}\end{bmatrix}}_{\boldsymbol{B}_{\lambda}^{VIR}}\boldsymbol{\lambda}^{VIR}+\underbrace{\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{M}^{VIR^{-1}}\end{bmatrix}}_{\boldsymbol{B}_{ext}^{VIR}}\boldsymbol{f}_{ext}^{VIR}\\ \boldsymbol{y}^{VIR}=\underbrace{\begin{bmatrix}\boldsymbol{G}^{VIR} & \boldsymbol{0}\end{bmatrix}}_{\boldsymbol{C}_{y}^{VIR}}\boldsymbol{x}^{VIR}\quad\text{with}\quad\boldsymbol{x}^{VIR}=\begin{bmatrix}\boldsymbol{q}^{VIR^T} & \dot{\boldsymbol{q}}^{VIR^T}\end{bmatrix}^T\end{gathered} \tag{2.10}$$

The state-space matrices $\boldsymbol{A}^{VIR}$, $\boldsymbol{B}_{u}^{VIR}$, $\boldsymbol{B}_{ext}^{VIR}$ and $\boldsymbol{C}_{y}^{VIR}$ as well as the state vector $\boldsymbol{x}^{VIR}$ are used in the equations. Note that $\boldsymbol{M}^{VIR}$ is assumed to be invertible in this notation.
The experimental component's dynamics are coupled to the dynamics of the actuation system. Since it

is necessary to account for the coupled dynamics, actuation system and experimental component are combined in one state-space equation. Note, that the coupled dynamics of actuator system and experimental component are denoted with the superscript TR referring to the test rig. The actuation system may contain an inner-loop for force or displacement control with actuator-specific features such as friction compensation. Eqs. (2.11) describe the dynamics of the actuation system and the experimental component in state-space form

$$\begin{aligned}\dot{\boldsymbol{x}}^{TR} &= \boldsymbol{A}^{TR}\boldsymbol{x}^{TR} + \boldsymbol{B}_{u}^{TR}\boldsymbol{u} + \boldsymbol{B}_{ext}^{TR}\boldsymbol{f}_{ext}^{TR}\\ \boldsymbol{y}^{EXP} &= \boldsymbol{C}_{y}^{TR}\boldsymbol{x}^{TR}\\ \boldsymbol{\lambda}^{EXP} &= \boldsymbol{C}_{\lambda}^{TR}\boldsymbol{x}^{TR} + \boldsymbol{D}_{\lambda,u}^{TR}\boldsymbol{u} + \boldsymbol{D}_{\lambda,ext}^{TR}\boldsymbol{f}_{ext}^{TR}\end{aligned} \tag{2.11}$$

using state-space matrices $\boldsymbol{A}^{TR}$, $\boldsymbol{B}_{u}^{TR}$, $\boldsymbol{B}_{ext}^{TR}$, $\boldsymbol{C}_{y}^{TR}$, $\boldsymbol{C}_{\lambda}^{TR}$, $\boldsymbol{D}_{\lambda,u}^{TR}$ and $\boldsymbol{D}_{\lambda,ext}^{TR}$ as well as the state vector $\boldsymbol{x}^{TR}$. The inputs to the system are the actuator demand signal $\boldsymbol{u}$ and the external forces acting on the experimental component. The interface displacement of the experimental component $\boldsymbol{y}^{EXP}$ and the interface forces $\boldsymbol{\lambda}^{EXP}$ are defined as the outputs of the system. This state space form can for example be derived as follows: The second time derivative of the compatibility constraint is

$$\boldsymbol{G}^{ACT}\ddot{\boldsymbol{q}}^{ACT} - \boldsymbol{G}^{EXP}\ddot{\boldsymbol{q}}^{EXP} = \boldsymbol{0}. \tag{2.12}$$

Substituting $\ddot{\boldsymbol{q}}^{ACT}$ and $\ddot{\boldsymbol{q}}^{EXP}$ from Eq. (2.2) and Eq. (2.3) yields the expression

$$\begin{aligned}\boldsymbol{\lambda}^{EXP} =& \left(\boldsymbol{G}^{ACT}\boldsymbol{G}^{ACT^T} - \boldsymbol{G}^{EXP}\boldsymbol{G}^{EXP^T}\right)^{-1}\boldsymbol{G}^{ACT}\boldsymbol{M}^{ACT^{-1}}\left(\boldsymbol{D}^{ACT}\dot{\boldsymbol{q}}^{ACT} + \boldsymbol{K}^{ACT}\boldsymbol{q}^{ACT} - \boldsymbol{B}_{ACT}\boldsymbol{u}^{ACT}\right)\\ &-\left(\boldsymbol{G}^{ACT}\boldsymbol{G}^{ACT^T} - \boldsymbol{G}^{EXP}\boldsymbol{G}^{EXP^T}\right)^{-1}\boldsymbol{G}^{EXP}\boldsymbol{M}^{EXP^{-1}}\left(\boldsymbol{D}^{EXP}\dot{\boldsymbol{q}}^{EXP} + \boldsymbol{K}^{EXP}\boldsymbol{q}^{EXP} - \boldsymbol{f}_{ext}^{EXP}\right)\end{aligned} \tag{2.13}$$

Using the state vector $\boldsymbol{x}^{TR} = \begin{bmatrix}\boldsymbol{q}^{ACT^T} & \boldsymbol{q}^{EXP^T}\end{bmatrix}^T$ and substituting $\boldsymbol{\lambda}^{EXP}$ in Eq. (2.2) and Eq. (2.3) the state space matrices can be constructed.

2.1.3 Continuous Time Convolution Integrals

For linear time invariant systems, the relation between input and output can be described by a convolution integral. In other words, instead of time integration of the full system, convolution integrals are used to compute the system responses. For a general linear, time-invariant state-space system

$$\begin{aligned}\dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}\\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x},\end{aligned} \tag{2.14}$$

the analytical solution is

$$\boldsymbol{y}(t) = \int_0^t \underbrace{\boldsymbol{C}\mathrm{e}^{\boldsymbol{A}(t-\tau)}\boldsymbol{B}}_{\boldsymbol{H}(t)}\boldsymbol{u}(\tau)\mathrm{d}\tau + \boldsymbol{C}\mathrm{e}^{\boldsymbol{A}t}\boldsymbol{x}(0). \tag{2.15}$$

The solution consists of a term which originates from the initial condition $\boldsymbol{x}(\boldsymbol{0})$ and one term which originates from the response to the input signal. The $\boldsymbol{H}(t)$ is a matrix of the responses to Dirac impulses—shortly the impulse response matrix.

For the virtual subcomponent—considering the inputs from external forces and interface forces—the convolution integral is

$$\begin{aligned} \boldsymbol{y}^{VIR}(t) = & \int_0^t \underbrace{\boldsymbol{C}^{VIR} \mathrm{e}^{\boldsymbol{A}^{VIR}(t-\tau)} \boldsymbol{B}_\lambda^{VIR}}_{\boldsymbol{H}_\lambda^{VIR}(t-\tau)} \boldsymbol{\lambda}^{VIR}(\tau) \mathrm{d}\tau \\ & + \int_0^t \underbrace{\boldsymbol{C}^{VIR} \mathrm{e}^{\boldsymbol{A}^{VIR}(t-\tau)} \boldsymbol{B}_{ext}^{VIR}}_{\boldsymbol{H}_{ext}^{VIR}(t-\tau)} \boldsymbol{f}_{ext}^{VIR}(\tau) \mathrm{d}\tau + \boldsymbol{C}^{VIR} \mathrm{e}^{\boldsymbol{A}^{VIR} t} \boldsymbol{x}^{VIR}(0). \end{aligned} \tag{2.16}$$

Neglecting the initial condition $\boldsymbol{x}(0)$—and only considering the convolution integral—the expression can be written using the convolution operator $*$:

$$\boldsymbol{y}^{VIR}(t) = \boldsymbol{H}_\lambda^{VIR}(t) * \boldsymbol{\lambda}^{VIR}(t) + \boldsymbol{H}_{ext}^{VIR}(t) * \boldsymbol{f}_{ext}^{VIR}(t)$$

2.1.4 Discrete Time Convolution Integrals

The continuous convolution integral can be discretized in time assuming zero initial conditions ($\boldsymbol{q}(0) = 0$ and $\dot{\boldsymbol{q}}(0) = 0$) and using a simple approximation with the time step width Δt:

$$\boldsymbol{y}[n] = \sum_{k=0}^{n} \boldsymbol{H}[n-k] \boldsymbol{u}[k] \Delta t \tag{2.17}$$

The square-brackets notation is used to indicate the index of the time instance. $\boldsymbol{H}[n]$ is the matrix of the responses at time step n to impulses at the initial time step. The time-discrete impulse responses can be computed by using a time integration scheme as e.g. the commonly used method of Newmark (see Sec. 2.3 and Sec. 2.4.1). Using the fact that the impulse response $\boldsymbol{H}[0]$ at time step $n = 0$ is zero, the term containing $\boldsymbol{H}[0]$ can be omitted and the discrete convolution reads

$$\boldsymbol{y}[n] = \sum_{k=0}^{n-1} \boldsymbol{H}[n-k] \boldsymbol{u}[k] \Delta t \tag{2.18}$$

A more accurate approach is to apply the trapezoidal rule in order to approximate the input forces. The resulting approximation of the convolution integral is

$$\boldsymbol{y}[n] = \sum_{k=0}^{n-1} \boldsymbol{H}[n-k] (\boldsymbol{u}[k] + \boldsymbol{u}[k+1]) \frac{\Delta t}{2} \tag{2.19}$$

A proof is given in [88] that the formulation is equivalent to Newmark time integration of the system with $\beta = \frac{1}{4}$, $\gamma = \frac{1}{2}$. For the virtual subcomponent—considering the inputs from external forces and interface forces—the time-discrete convolution writes

$$\boldsymbol{y}_{VIR}[n] = \sum_{k=0}^{n-1} \boldsymbol{H}_\lambda^{VIR}[n-k] \boldsymbol{\lambda}^{VIR}[k] \Delta t + \sum_{k=0}^{n-1} \boldsymbol{H}_{ext}^{VIR}[n-k] \boldsymbol{f}_{ext}^{VIR}[k] \Delta t. \tag{2.20}$$

Using the trapezoidal rule, the time–discrete form of the convolution integral for the virtual subcomponent writes

$$\begin{aligned} \boldsymbol{y}_{VIR}[n] = & \sum_{k=0}^{n-1} \boldsymbol{H}_\lambda^{VIR}[n-k] (\boldsymbol{\lambda}^{VIR}[k] + \boldsymbol{\lambda}^{VIR}[k+1]) \frac{\Delta t}{2} \\ & + \sum_{k=0}^{n-1} \boldsymbol{H}_{ext}^{VIR}[n-k] (\boldsymbol{f}_{ext}^{VIR}[k] + \boldsymbol{f}_{ext}^{VIR}[k+1]) \frac{\Delta t}{2}. \end{aligned} \tag{2.21}$$

The trapezoidal approach yields an accurate and unconditionally stable approximation of the convolution integral. However, in practice, the impulse responses are truncated—as described in Sec. 2.4.2—which makes the convolution products inherently stable. The simple approximation of Eq. (2.17) has the advantage that it only requires forces from the preceding time steps as inputs. This fact makes it interesting in the context of real-time hybrid testing.

2.1.5 Transfer Functions

The transfer function description of the systems can be derived from the state-space form, the impulse responses, or directly from the second-order form. In the Laplace domain, the general linear system of Eq. (2.14) can be written as

$$\boldsymbol{Y}(s) = \boldsymbol{H}(s)\boldsymbol{U}(s) \tag{2.22}$$

using the Laplace transforms $\mathcal{L}(\boldsymbol{H}(t)) = \boldsymbol{H}(s)$, $\mathcal{L}(\boldsymbol{u}(t)) = \boldsymbol{U}(s)$ and $\mathcal{L}(\boldsymbol{y}(t)) = \boldsymbol{Y}(s)$ of the impulse responses.

For the virtual subcomponent, the inputs from the external forces and the interface forces have to be considered:

$$\boldsymbol{Y}^{VIR}(s) = \boldsymbol{H}_{\lambda}^{VIR}(s)\boldsymbol{\Lambda}^{VIR}(s) + \boldsymbol{H}_{ext}^{VIR}(s)\boldsymbol{F}_{ext}^{VIR}(s) \tag{2.23}$$

The input-output relationships of the coupled system which are composed of experimental subcomponent and the actuation subsystem can be described by Eqs. (2.24). The output includes the interface forces as well as the interface displacements.

$$\begin{aligned} \boldsymbol{Y}^{EXP}(s) &= \boldsymbol{H}_{y,u}^{TR}(s)\boldsymbol{U}(s) + \boldsymbol{H}_{y,ext}^{TR}(s)\boldsymbol{F}_{ext}^{EXP}(s) \\ \boldsymbol{\Lambda}^{EXP}(s) &= \boldsymbol{H}_{\lambda,u}^{TR}(s)\boldsymbol{U}(s) + \boldsymbol{H}_{\lambda,ext}^{TR}(s)\boldsymbol{F}_{ext}^{EXP}(s) \end{aligned} \tag{2.24}$$

Neglecting the initial conditions and using the Fourier transforms $\mathcal{F}(\boldsymbol{H}(t)) = \boldsymbol{H}(\omega)$, $\mathcal{F}(\boldsymbol{u}(t)) = \boldsymbol{U}(\omega)$, $\mathcal{F}(\boldsymbol{f}(t)) = \boldsymbol{F}(\omega)$, $\mathcal{F}(\boldsymbol{\lambda}(t)) = \boldsymbol{\Lambda}(\omega)$ and $\mathcal{F}(\boldsymbol{y}(t)) = \boldsymbol{Y}(\omega)$, one can equivalently write the input-output relationship in frequency domain:

$$\begin{aligned} \boldsymbol{Y}^{VIR}(\omega) &= \boldsymbol{H}_{\lambda}^{VIR}(\omega)\boldsymbol{\Lambda}^{VIR}(\omega) + \boldsymbol{H}_{ext}^{VIR}(\omega)\boldsymbol{F}_{ext}^{VIR}(\omega) \\ \boldsymbol{Y}^{EXP}(s) &= \boldsymbol{H}_{y,u}^{TR}(\omega)\boldsymbol{U}(\omega) + \boldsymbol{H}_{y,ext}^{TR}(\omega)\boldsymbol{F}_{ext}^{EXP}(\omega) \\ \boldsymbol{\Lambda}^{EXP}(\omega) &= \boldsymbol{H}_{\lambda,u}^{TR}(s)\boldsymbol{U}(\omega) + \boldsymbol{H}_{\lambda,ext}^{TR}(\omega)\boldsymbol{F}_{ext}^{EXP}(\omega) \end{aligned} \tag{2.25}$$

2.2 Substructure Model-Order Reduction

The discretization of mechanical problems using FE methods usually yields large models. The reason is that a fine meshing is often necessary to account for the geometry of the structures. It is not uncommon that the resulting models have sizes over a million DOF. The high number of DOF—in many cases—is not necessary to represent the relevant dynamics of the mechanical system. Making meshes coarser is, however, often not an option, since careful re-meshing is very labor intensive and fine meshes are needed to represent the geometry. Especially in real-time hybrid testing, efficient computation of the responses of the virtual subcomponent is essential to meet the real-time constraints. Model order reduction methods are used to systematically and automatically simplify models without sacrificing the accuracy of the results. The following section gives a short overview of the model order reduction techniques used in this thesis. It is based on the deeper and more complete discussions by [107].

The following considerations are limited to linear models. Model order reduction techniques in control engineering focus on the optimization of the accuracy of input-output relationships while in structural engineering the full displacement and stress fields are of interest. The methods described in the following section have their origin in the field of structural dynamics.

2.2.1 Subspace Projection

The full linear system is given by Eq. (2.26).

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{D}\dot{\boldsymbol{u}} + \boldsymbol{K}\boldsymbol{u} = \boldsymbol{f} + \boldsymbol{G}^T\boldsymbol{\lambda} \tag{2.26}$$

The general idea of model order reduction is to substitute the full coordinate vector $\boldsymbol{q} \in \mathbb{R}^m$ with a reduced set of coordinates $\tilde{\boldsymbol{q}} \in \mathbb{R}^n$ $(m > n)$ according to Eq. (2.27).

$$\boldsymbol{q} = \boldsymbol{R}\tilde{\boldsymbol{q}} \tag{2.27}$$

The reduction matrix $\boldsymbol{R} \in \mathbb{R}^{m\times n}$ contains the shapes of the possible displacements performed by the full coordinates. The reduced coordinate vector $\tilde{\boldsymbol{q}}$ represents the contributions of the shapes contained in the reduction matrix. Since the space which is spanned by the new coordinates is not sufficiently large for the equations to be satisfied fully, a residual error $\boldsymbol{r}$ remains in Eq. (2.28).

$$\boldsymbol{M}\boldsymbol{R}\ddot{\tilde{\boldsymbol{q}}} + \boldsymbol{D}\boldsymbol{R}\dot{\tilde{\boldsymbol{q}}} + \boldsymbol{K}\boldsymbol{R}\tilde{\boldsymbol{q}} = \boldsymbol{f} + \boldsymbol{G}^T\boldsymbol{\lambda} + \boldsymbol{r} \tag{2.28}$$

The residual forces do not perform mechanical work in the modes represented by $\boldsymbol{R}$. Hence, the residual $\boldsymbol{r}$ disappears if the equations are projected onto the space spanned by $\boldsymbol{R}$. In order to do so, Eq. (2.28) is premultiplied with the transpose $\boldsymbol{R}^T$. The resulting reduced system is given by Eq.(2.29).

$$\tilde{\boldsymbol{M}}\ddot{\tilde{\boldsymbol{q}}} + \tilde{\boldsymbol{D}}\dot{\tilde{\boldsymbol{q}}} + \tilde{\boldsymbol{K}}\tilde{\boldsymbol{q}} = \tilde{\boldsymbol{f}} + \tilde{\boldsymbol{G}}^T\boldsymbol{\lambda}, \tag{2.29}$$

$\boldsymbol{R}^T\boldsymbol{r}$ cancels to zero. The resulting reduced system matrices are given by Eq. (2.30).

$$\tilde{\boldsymbol{M}} = \boldsymbol{R}^T\boldsymbol{M}\boldsymbol{R}, \qquad \tilde{\boldsymbol{D}} = \boldsymbol{R}^T\boldsymbol{D}\boldsymbol{R}, \qquad \tilde{\boldsymbol{K}} = \boldsymbol{R}^T\boldsymbol{K}\boldsymbol{R}, \qquad \tilde{\boldsymbol{f}} = \boldsymbol{R}^T\boldsymbol{f}, \qquad \tilde{\boldsymbol{G}} = \boldsymbol{R}^T\boldsymbol{G} \tag{2.30}$$

Matrix Partitioning In practice, one is interested in the solution of the coupled system. Maintaining the interface coordinates is necessary. Typically, the reduced substructure matrices can be assembled similarly to the element matrices in the finite element method. This process is referred to as the primal assembly. In order to perform the assembly, it is necessary to distinguish internal and interface coordinates. In order to do so, the equations of motion can be rearranged such that the DOFs at the interface and the internal DOFs are separated according to Eqs. (2.31). Partitioning the equations of motion is also necessary for defining and computing the reduction matrix $\boldsymbol{R}$. The coordinates which are part of the interface (or boundary) are denoted with subscript b and internal DOF are denoted with subscript i. In the reduction process, the interface DOFs remain unchanged, while the internal DOF are replaced by a set of reduced, generalized coordinates.

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{M}_{bb} & \boldsymbol{M}_{bi} \\ \boldsymbol{M}_{ib} & \boldsymbol{M}_{ii} \end{bmatrix}, \quad \boldsymbol{K} = \begin{bmatrix} \boldsymbol{K}_{bb} & \boldsymbol{K}_{bi} \\ \boldsymbol{K}_{ib} & \boldsymbol{K}_{ii} \end{bmatrix}, \quad \boldsymbol{G} = \begin{bmatrix} \boldsymbol{G}_b & \boldsymbol{0} \end{bmatrix}, \quad \boldsymbol{f} = \begin{bmatrix} \boldsymbol{f}_b \\ \boldsymbol{f}_i \end{bmatrix}, \quad \boldsymbol{q} = \begin{bmatrix} \boldsymbol{q}_b \\ \boldsymbol{q}_i \end{bmatrix} \tag{2.31}$$

By definition, interface forces $\boldsymbol{\lambda}$ act on the boundary. The partitioned equations of motion are:

$$\begin{bmatrix} \boldsymbol{M}_{bb} & \boldsymbol{M}_{bi} \\ \boldsymbol{M}_{ib} & \boldsymbol{M}_{ii} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}}_b \\ \ddot{\boldsymbol{q}}_i \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_{bb} & \boldsymbol{K}_{bi} \\ \boldsymbol{K}_{ib} & \boldsymbol{K}_{ii} \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_b \\ \boldsymbol{q}_i \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_b \\ \boldsymbol{f}_i \end{bmatrix} + \begin{bmatrix} \boldsymbol{G}_b \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{\lambda} \tag{2.32}$$

Note that the damping term is omitted for readability reasons.

2.2.2 Reduction Basis: Component Modes

The reduction basis $\boldsymbol{R}$ is used to compute a reduced system representation as it is described in Sec. 2.2.1. The accuracy and efficiency of the reduction relies on the quality of the employed modes in the reduction matrix. Different reduction methods are obtained by combining different types of modes. In the context of substructuring, those modes are referred to as component modes. In the following section, a number of component mode types which are relevant for real-time hybrid testing are listed.

Constraint Modes Constraint modes are the static responses of the component interface displacements. They can be derived using the second line in the partitioned equations of motion (2.32):

$$\boldsymbol{M}_{ib}\ddot{\boldsymbol{q}}_b + \boldsymbol{M}_{ii}\ddot{\boldsymbol{q}}_i + \boldsymbol{K}_{ib}\boldsymbol{q}_b + \boldsymbol{K}_{ii}\boldsymbol{q}_i = \boldsymbol{0}$$

Neglecting the inertia contributions, one finds the static relationship between $\boldsymbol{q}_b$ and $\boldsymbol{q}_i$:

$$\boldsymbol{K}_{ib}\boldsymbol{q}_b + \boldsymbol{K}_{ii}\boldsymbol{q}_i = \boldsymbol{0}$$

The equation can be solved for the internal DOFs $\boldsymbol{q}_i$:

$$\boldsymbol{q}_i = -\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{ib}\boldsymbol{q}_b$$

The resulting modes $\boldsymbol{\Psi}_c$ represent the static deformations due to unit displacements of the interface DOFs. In fact, the internal DOFs are coupled to the interface DOFs through their static responses.

$$\boldsymbol{\Psi}_c = -\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{ib}$$

Free Interface Modes The free interface modes are the vibration modes of the component without any further constraints. They are defined by the system equations

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{K}\boldsymbol{q} = \boldsymbol{0}.$$

The eigenvalue problem which has to be solved is defined by

$$(\boldsymbol{K} - \boldsymbol{M}\omega_k^2)\boldsymbol{\phi}_k = \boldsymbol{0} \qquad \text{with} \qquad k \in [1 \ldots N]$$

where N is the number of modes considered in the reduction basis. The matrix of the free interface modes is denoted by $\boldsymbol{\Phi}_f$:

$$\boldsymbol{\Phi}_f = \left[\boldsymbol{\phi}_1, \boldsymbol{\phi}_2 \cdots \boldsymbol{\phi}_N\right]$$

The eigenmodes can be computed using so-called subspace iterations. The Lanczos method is another computationally more efficient technique of computing the eigenmodes. For details on the algorithms for the computation of eigenmodes, the reader is referred to the textbook of [36].

Fixed Interface Modes Fixed interface modes are vibration modes which occur when the boundary DOFs are fixed. They are computed by fixing the DOFs and solving the resulting eigenvalue problem. In order to fix the boundary DOFs the corresponding coordinate vector $\boldsymbol{q}_b$ is set to zero. Only the internal DOFs stay relevant in the equations of motion:

$$\boldsymbol{M}_{ii}\ddot{\boldsymbol{q}}_i + \boldsymbol{K}_{ii}\boldsymbol{q}_i = \boldsymbol{0}.$$

The associated eigenvalue problem is defined by

$$(\boldsymbol{K}_{ii} - \boldsymbol{M}_{ii}\omega_k^2)\boldsymbol{\phi}_k = \boldsymbol{0} \qquad \text{with} \qquad k \in [1 \ldots N]$$

where N is the number of modes considered in the reduction basis.

$$\boldsymbol{\Phi}_i = \left[\boldsymbol{\phi}_1, \boldsymbol{\phi}_2 \cdots \boldsymbol{\phi}_N\right]$$

The same eigensolvers as mentioned for the free interface modes can be used for the computation of the fixed interface modes.

Rigid-Body Modes Rigid-body modes are displacement shapes where the component does not undergo any deformations. They occur if the component is not fully constrained. Because in substructuring many constraints are only applied due to the substructure assembly, substructures often exhibit rigid body modes. Rigid body modes are mathematically defined by Eq. (2.33).

$$\boldsymbol{K}\boldsymbol{\phi}_k = \boldsymbol{0} \tag{2.33}$$

The computation of the rigid-body modes can be performed by computing the nullspace of the stiffness matrix, by computing the free interface modes with zero eigenfrequency or by geometric considerations. The rigid-body modes are used in the filter algorithms of Sec. 2.4.

2.2.3 Model Reduction Methods

Modal-Truncation Perhaps the simplest method of model order reduction is the modal truncation method. The reduction basis $\boldsymbol{R}$ is built up using free interface modes:

$$\boldsymbol{R} = \left[\boldsymbol{\phi}_1, \boldsymbol{\phi}_2 \dots \boldsymbol{\phi}_N\right]$$

According to [36], a rule of thumb can be applied for the selection of eigenmodes in the reduction basis: The N can be selected such that frequencies up to double the frequency of interest are considered. Usually, the eigenmodes are mass normalized such that

$$\boldsymbol{\phi}_i^T \boldsymbol{M}\boldsymbol{\phi}_i = 1$$

holds. Since the eigenmodes are mass and stiffness orthogonal, the reduced mass matrix then becomes the identity matrix.

$$\tilde{\boldsymbol{M}} = \boldsymbol{R}^T \boldsymbol{M}\boldsymbol{R} = \boldsymbol{I}$$

The reduced stiffness matrix is a diagonal matrix containing the squared eigenfrequencies.

$$\tilde{\boldsymbol{K}} = \boldsymbol{R}^T \boldsymbol{K}\boldsymbol{R} = \begin{bmatrix} \omega_1^2 & & \\ & \ddots & \\ & & \omega_N^2 \end{bmatrix}$$

If Rayleigh damping is applied, the reduced damping matrix is also decoupled according to

$$\tilde{\boldsymbol{D}} = \boldsymbol{R}^T \boldsymbol{D}\boldsymbol{R} = \begin{bmatrix} 2\zeta_1\omega_1 & & \\ & \ddots & \\ & & 2\zeta_N\omega_N \end{bmatrix}$$

where ζ_i is the damping ratio which can be experimentally identified. In real-time hybrid testing, the use of modal truncation has been proposed by [14] to address stability problems and to improve the computational effort for the time integration. The computational efficiency is essential due to the real-time requirements. The stability in conditionally stable integration schemes—as the explicit Newmark time integration method—depends on the highest frequency in the system. By applying the modal truncation method to the model of the virtual component and by removing high-frequency modes, the stability properties of a test setup can be improved significantly. Also, the accuracy is affected by the selection of the modal basis: according to [14], the time step size has to be selected such that the sampling frequency exceeds the highest mode in the system by factor 10. In order to ensure that the accuracy requirements are met, the experimental component dynamics have to be included in this consideration as well. In [97], a similar approach has already been proposed for pseudodynamic testing.

An advantage is the decoupling of the equations of motion. It further reduces the computational efforts for the solution of the system e.g. in time integration. The properties of the decoupled system can be obtained experimentally. Using experimental modal analysis, modal parameters such as the damping ratio can be identified directly. However, a reduction basis which makes use of pure modal truncation does not account for the deformations caused by the coupling of the substructure. This problem is addressed by substructure reduction methods such as the Craig-Bampton method.

Guyan's Reduction Method Guyan's reduction method which is also referred to as static condensation was proposed in [39]. In Guyan's reduction method, only constraint modes are used. That means that the inertia effects are neglected in the reduction basis. The reduction then reads

$$\begin{bmatrix} \boldsymbol{q}_b \\ \boldsymbol{q}_i \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{\Psi}_c \end{bmatrix} \boldsymbol{q}_b = \begin{bmatrix} \boldsymbol{I} \\ -\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{ib} \end{bmatrix} \boldsymbol{q}_b.$$

The resulting reduced system matrices are

$$\tilde{\boldsymbol{M}} = \boldsymbol{M}_{bb} - \boldsymbol{M}_{bi}\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{ib} - \boldsymbol{K}_{bi}\boldsymbol{K}_{ii}^{-1}\boldsymbol{M}_{ib} + \boldsymbol{K}_{bi}\boldsymbol{K}_{ii}^{-1}\boldsymbol{M}_{ii}\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{ib} \quad \text{and}$$
$$\tilde{\boldsymbol{K}} = \boldsymbol{K}_{bb} - \boldsymbol{K}_{bi}\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{ib}.$$

Since all DOFs are condensed on the boundary DOFs, the set of boundary coordinates corresponds to the reduced set of coordinates:

$$\tilde{\boldsymbol{q}} = \boldsymbol{q}_b$$

The reduced external force vector and the reduced interface matrix become

$$\tilde{\boldsymbol{f}} = \boldsymbol{f}_b + \Psi_c^T \boldsymbol{f}_i \quad \text{and} \quad \tilde{\boldsymbol{G}} = \boldsymbol{G}_b.$$

Guyan's reduction method is exact for static problems, because only the inertia forces are neglected in the derivation. The first eigenfrequency of the structure being clamped at the interface can serve as a good reference for the quality of the reduction: The approximation is valid, if the excitation frequencies are lower than the first eigenfrequency of the clamped structure.

Craig-Bampton Reduction Method The popular Craig-Bampton substructure reduction method was proposed in [25]. In the Craig-Bampton reduction basis, fixed interface vibration modes $\boldsymbol{\Phi}_i$ are used additionally to the static modes $\boldsymbol{\Psi}_c$ from Guyan's reduction. The reduced coordinate vector reads

$$\boldsymbol{q} = \boldsymbol{R}\tilde{\boldsymbol{q}} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{\Psi}_c & \boldsymbol{\Phi}_i \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_b \\ \tilde{\boldsymbol{q}}_i \end{bmatrix}.$$

Note that the full set of interface coordinates $\boldsymbol{q}_b$ is maintained. As a rule of thumb, N modes with eigenfrequencies up to double the relevant frequencies are added to the reduction basis $\boldsymbol{\Phi}_i$. Applying the reduction to the system matrix, the reduced system matrices are obtained according to Eq. (2.30). The reduced mass matrix can be written as

$$\tilde{\boldsymbol{M}} = \begin{bmatrix} \tilde{\boldsymbol{M}}_{bb} & \tilde{\boldsymbol{M}}_{bi} \\ \tilde{\boldsymbol{M}}_{ib} & \boldsymbol{I} \end{bmatrix}$$
$$\text{with} \quad \tilde{\boldsymbol{M}}_{bb} = \boldsymbol{M}_{bb} - \boldsymbol{M}_{bi}\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{ib} - \boldsymbol{K}_{bi}\boldsymbol{K}_{ii}^{-1}\boldsymbol{M}_{ib} + \boldsymbol{K}_{bi}\boldsymbol{K}_{ii}^{-1}\boldsymbol{M}_{ii}\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{ib}$$
$$\text{and} \quad \tilde{\boldsymbol{M}}_{ib} = \tilde{\boldsymbol{M}}_{bi}^T = \boldsymbol{\Phi}_i^T\left(\boldsymbol{M}_{ib} - \boldsymbol{M}_{ii}\boldsymbol{K}_{ii}^T\boldsymbol{K}_{ib}\right).$$

The reduced stiffness matrix reads

$$\tilde{K} = \begin{bmatrix} \tilde{K}_{bb} & & & \\ & \omega_1 & & \\ & & \ddots & \\ & & & \omega_N \end{bmatrix}$$

$$\text{with} \quad \tilde{K}_{bb} = K_{bb} - K_{bi} K_{ii}^{-1} K_{ib}.$$

The Craig-Bampton method is implemented in many finite element codes where the substructures are used as superelements. This means in practice that the reduced system matrices for real-time hybrid testing can be obtained directly from commercial software packages.

2.3 Time Integration

Initial value problems involving the dynamics of mechanical systems can be solved using time integration. Time integration schemes approximate the derivatives in an ordinary differential equation with finite differences. Starting from the initial value, the solution can then be obtained by stepping over all time steps and solving algebraic equations at each time step.

In order to perform a real-time hybrid test, it is an important step to compute the responses of the virtual system to interface forces and external excitations. Since stability and accuracy issues highly depend on the stepwise sequence of calculations, it is critical to have insight into the specific time integration schemes. In this section, the most popular techniques in structural dynamics are described.

For general first-order ODEs, Runge-Kutta or linear multi-step time integration methods can be applied. As described in Sec. 2.1.2, a second-order ordinary differential equation can always be transformed into the first-order form. However, since space-discretization of mechanical systems usually directly results in the form of second-order ordinary differential equations, this section focuses on methods for the solution of second order differential equations. A family of time integration methods was introduced in [70]. It allows describing several implicit and explicit time integration schemes. This family of methods is referred to as Newmark time integration methods and is widely used in the domain of structural dynamics. The choice of the specific method depends on the application.

Many time integration schemes used in real-time hybrid testing are based on the Newmark time integration schemes (see [18]) or can be derived starting from a Newmark based scheme. In the following section, Newmark-type time integration schemes and the implications of coupling forces during a hybrid test are discussed. For simulation purposes, it is crucial to compute the responses of coupled systems. In this context, the solution of constraint systems using the Newmark method and the so-called HHT-α method are discussed.

2.3.1 The Newmark Method for Time Integration

The Newmark time integration schemes are popular in the field of structural dynamics since they are directly applicable to second-order differential equations. The following section serves as a introduction to the methods specific to real-time hybrid testing and is based on the discussions in [35]. The core idea of the Newmark time integration is the approximation of system states $\dot{q}$ and q at time step $n+1$. Note that the superscript VIR is omitted in the following section for readability reasons. In order to perform the approximations, the states $\dot{\boldsymbol{q}}[n+1]$ and $\boldsymbol{q}[n+1]$ are expressed using the Taylor series of Eq. (2.34).

$$\begin{aligned}\dot{\boldsymbol{q}}[n+1] &= \dot{\boldsymbol{q}}[n] + \int_{t[n]}^{t[n+1]} \ddot{\boldsymbol{q}}(\tau)\mathrm{d}\tau \\ \boldsymbol{q}[n+1] &= \boldsymbol{q}[n] + \Delta t\dot{\boldsymbol{q}}[n] + \int_{t[n]}^{t[n+1]} (t[n+1]-\tau)\ddot{\boldsymbol{q}}(\tau)\mathrm{d}\tau\end{aligned} \tag{2.34}$$

The next step is to approximate $\ddot{\boldsymbol{q}}(\tau)$ at times $\tau \in [t[n], t[n+1]]$. In order to do so, the second time derivatives $\ddot{\boldsymbol{q}}[n]$ and $\ddot{\boldsymbol{q}}[n+1]$ at the limits of the interval are approximated using a Taylor series around the time τ:

$$\ddot{\boldsymbol{q}}[n] = \ddot{\boldsymbol{q}}(\tau) + \boldsymbol{q}^{(3)}(\tau)(t[n]-\tau) + \boldsymbol{q}^{(4)}(\tau)\frac{(t[n]-\tau)}{2} + \ldots \tag{2.35}$$

$$\ddot{\boldsymbol{q}}[n+1] = \ddot{\boldsymbol{q}}(\tau) + \boldsymbol{q}^{(3)}(\tau)(t[n+1]-\tau) + \boldsymbol{q}^{(4)}(\tau)\frac{(t[n+1]-\tau)}{2} + \ldots \tag{2.36}$$

Multiplying Eq. (2.35) with $1-\gamma$ and Eq. (2.36) with γ and adding both yields Eq. (2.37), which allows the interpolation between the contributions of $\ddot{\boldsymbol{q}}[n]$ and $\ddot{\boldsymbol{q}}[n+1]$. Equivalently, multiplying Eq. (2.35) with $1-2\beta$ and Eq. (2.36) with β and adding both yields Eq. (2.38).

$$\ddot{\boldsymbol{q}}(\tau) = (1-\gamma)\ddot{\boldsymbol{q}}[n] + \gamma\ddot{\boldsymbol{q}}[n+1] + (\tau - \Delta t\gamma - t[n])\boldsymbol{q}^{(3)}(\tau) + \mathcal{O}(\Delta t^2 \boldsymbol{q}^{(4)}) \tag{2.37}$$

$$\ddot{\boldsymbol{q}}(\tau) = (1-2\beta)\ddot{\boldsymbol{q}}[n] + 2\beta\ddot{\boldsymbol{q}}[n+1] + (\tau - 2\Delta t\beta - t[n])\boldsymbol{q}^{(3)}(\tau) + \mathcal{O}(\Delta t^2 \boldsymbol{q}^{(4)}) \tag{2.38}$$

Substituting Eq. (2.37) and Eq. (2.38) in Eqs. (2.34) yields the Newmark approximation formulas:

$$\begin{aligned}\dot{\boldsymbol{q}}[n+1] &= \underbrace{\dot{\boldsymbol{q}}[n] + \Delta t(1-\gamma)\ddot{\boldsymbol{q}}[n]}_{\dot{\boldsymbol{q}}^*[n+1]} + \Delta t\gamma\ddot{\boldsymbol{q}}[n+1] \\ \boldsymbol{q}[n+1] &= \underbrace{\boldsymbol{q}[n] + \Delta t\dot{\boldsymbol{q}}[n] + \Delta t^2\left(\frac{1}{2}-\beta\right)\ddot{\boldsymbol{q}}[n]}_{\boldsymbol{q}^*[n+1]} + \Delta t^2\beta\ddot{\boldsymbol{q}}[n+1]\end{aligned} \tag{2.39}$$

The terms in Eq. (2.39) depend on values from time step n. They can be seen as a prediction step and are denoted as $\boldsymbol{q}^*[n+1]$ and $\dot{\boldsymbol{q}}^*[n+1]$. The prediction is then updated using the acceleration $\ddot{\boldsymbol{q}}[n+1]$ from the new time step. The equations of motion of the virtual component are

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

Substituting the state vector and its time derivative with the Newmark approximations of Eq. (2.39) yields the algebraic equation which can be solved for the acceleration $\ddot{\boldsymbol{q}}[n+1]$:

$$\begin{aligned}\boldsymbol{S}\ddot{\boldsymbol{q}}[n+1] =& \boldsymbol{G}^T[n+1]\boldsymbol{\lambda}[n+1] + \boldsymbol{f}_{ext}[n+1] \\ & -\boldsymbol{D}\dot{\boldsymbol{q}}^*[n+1] - \boldsymbol{K}\boldsymbol{q}^*[n+1] \\ \text{with}\quad \boldsymbol{S} =& \boldsymbol{M} + \gamma\Delta t\boldsymbol{D} + \beta\Delta t^2\boldsymbol{K}\end{aligned} \tag{2.40}$$

β and γ are variables, which define the interpolation between contribution from the last time step n and the new time step $n+1$. By choosing γ and β, the various Newmark based algorithms are generated. Some important Newmark algorithms and their stability properties and accuracy are listed in Tab. 2.1. The stability depends on the product of time step width Δt and the highest frequency ω in the model. A detailed analysis of the properties of the algorithms can be found in [35].

Algorithm	γ	β	Stability limit $\omega\Delta t$	Amplitude error $\rho - 1$	Periodicity error $\Delta T/T$
Purely explicit	0	0	0	$\omega^2\Delta t^2/4$	0
Central difference	$1/2$	0	2	0	$\omega^2\Delta t^2/24$
Average constant acceleration	$1/2$	$1/4$	∞	0	$\omega^2\Delta t^2/2$
Average constant acceleration (modified)	$1/2+\alpha$	$(1+\alpha)^2/4$	∞	0	$\omega^2\Delta t^2/12$

Table 2.1: Stability and accuracy properties of algorithms from the Newmark family (modified from [35])

2.3.2 Non-Linear Systems

Newmarks approach can also be applied to non-linear systems of the form of Eq. (2.41): The prediction is performed as in the linear case but the algebraic equation has to be solved iteratively.

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{f}_{nl}(\boldsymbol{q},\dot{\boldsymbol{q}}) = \boldsymbol{G}(\boldsymbol{q})^T\boldsymbol{\lambda} + \boldsymbol{f}_{ext} \tag{2.41}$$

The prediction step—which is calculating the predicted states $\boldsymbol{q}^*[n+1]$ and $\dot{\boldsymbol{q}}^*[n+1]$—is performed according to Eq. (2.39). The residual of the nonlinear dynamic equation at time step $[n+1]$ is shown in Eq. (2.42).

$$\boldsymbol{r}_k = \boldsymbol{M}\left(\boldsymbol{q}_k\right)\ddot{\boldsymbol{q}}_k + \boldsymbol{f}_{nl}\left(\boldsymbol{q}_k,\dot{\boldsymbol{q}}_k\right) - \boldsymbol{G}\left(\boldsymbol{q}_k\right)^T\boldsymbol{\lambda}_k + \boldsymbol{f}_{ext} \tag{2.42}$$

Note that the time step number $[n+1]$ is dropped for readability reasons. In the following sections, the Newton iteration step is indicated by the subscript k.
$\ddot{\boldsymbol{q}}_{k+1}$ is expressed using the values from the last iteration step k and the acceleration update $\Delta\ddot{\boldsymbol{q}}_k$. Using the Newmark approximation formulas, $\boldsymbol{q}_{k+1}$ and $\dot{\boldsymbol{q}}_{k+1}$ can be formulated as a function of the values from the last iteration step k and the acceleration update $\Delta\ddot{\boldsymbol{q}}_k$:

$$\begin{aligned}\ddot{\boldsymbol{q}}_{k+1} &= \ddot{\boldsymbol{q}}_k + \Delta\ddot{\boldsymbol{q}}_k\\ \dot{\boldsymbol{q}}_{k+1} &= \dot{\boldsymbol{q}}_k + \gamma\Delta t\Delta\ddot{\boldsymbol{q}}_k\\ \boldsymbol{q}_{k+1} &= \boldsymbol{q}_k + \beta\Delta t^2\Delta\ddot{\boldsymbol{q}}_k\end{aligned} \tag{2.43}$$

The algebraic equation of the Newton-Ralphson method are obtained by setting the Taylor approximation of the residual of the next iteration step $\boldsymbol{r}_{k+1}$ to zero:

$$\boldsymbol{r}_{k+1} \approx \boldsymbol{r}_k + \frac{\partial\boldsymbol{r}_k}{\partial\ddot{\boldsymbol{q}}_k}\Delta\ddot{\boldsymbol{q}}_k \overset{!}{=} \boldsymbol{0} \tag{2.44}$$

Eq. (2.44) can be solved for $\Delta\ddot{\boldsymbol{q}}_k$ using the Jacobian matrix $\boldsymbol{S}_k$ from Eq. (2.45)

$$\begin{aligned}\frac{\partial\boldsymbol{r}_k}{\partial\ddot{\boldsymbol{q}}_k} &= \boldsymbol{S}_k = \boldsymbol{M}\left(\boldsymbol{q}_k\right)\\ &+\beta\Delta t^2\left(\frac{\partial\boldsymbol{M}\left(\boldsymbol{q}_k\right)\ddot{\boldsymbol{q}}_k}{\partial\boldsymbol{q}_k} + \frac{\partial\boldsymbol{f}_{nl}\left(\boldsymbol{q}_k,\dot{\boldsymbol{q}}_k\right)}{\partial\boldsymbol{q}_k} - \frac{\partial\boldsymbol{G}\left(\boldsymbol{q}_k\right)^T\boldsymbol{\lambda}_k}{\partial\boldsymbol{q}_k}\right)\\ &+\gamma\Delta t\left(\frac{\partial\boldsymbol{f}_{nl}\left(\boldsymbol{q}_k,\dot{\boldsymbol{q}}_k\right)}{\partial\dot{\boldsymbol{q}}_k} - \frac{\partial\boldsymbol{G}\left(\boldsymbol{q}_k\right)^T\boldsymbol{\lambda}_k}{\partial\dot{\boldsymbol{q}}_k}\right)\end{aligned} \tag{2.45}$$

Using the updated states from Eq. (2.43), the iteration can be continued until the norm of the residual has converged below the desired value.

2.3.3 Time Integration and Real-Time Hybrid Testing

In real-time hybrid testing, information on the interface forces and the interface states are exchanged between virtual and experimental component. That means that the actuator commands rely on the results from the time integration and the time integration is based on measurements of the interface forces. Due to this reason, not all time integration schemes which are applied to numerical simulations are applicable in real-time hybrid testing.

In general, one can distinguish between implicit and explicit time integration schemes. Implicit integration schemes can be unconditionally stable. This property is especially important if high frequencies are present in the model (see Tab. 2.1). Explicit time integration schemes, on the other hand, have stability boundaries. In the application to hybrid testing, however, the explicit time integration schemes have a crucial advantage: In explicit time integration schemes, the state at time step $n+1$ depends only on the state of the last time step n and its time derivatives. In contrast, the state at time step $n+1$ in the integration step of implicit time integration methods depends on the states from the current time step $n+1$. The consequence is that non-linear dynamics have to be solved in an iterative process. This makes $\boldsymbol{q}[n+1]$ much cheaper to compute in explicit schemes.

Several implicit and explicit time integration schemes have been adapted or developed for pseudo-dynamic testing. Iterations are possible in pseudo-dynamic tests. Nevertheless, they are not desirable since unwanted damages of the test specimen can be caused by overloading during the iterations. In contrast to pseudo-dynamic testing, the choice of time integration methods in real-time hybrid testing is restricted: Iterations—which are necessary to solve the non-linear dynamics in an implicit time integration scheme—are not possible in a real-time hybrid test since they would alter the dynamics of the experimental component.

The following paragraphs give a short overview of the most common integration schemes in real-time hybrid testing which are based on the Newmark family. A detailed evaluation of various time integration methods has been performed in [96], [17] and [18].

Pure explicit The only pure explicit time integration scheme with the Newmark parameters $\beta = 0$ and $\gamma = 0$ is unstable under all conditions. Therefore, it is not used in practice.

Central difference method The central difference method originates from the Newmark family and uses the parameters $\beta = 0$ and $\gamma = 1/2$. The method is not fully explicit, but the displacement command at time step $n+1$ can be computed from its time derivatives from time step n. The method is conditionally stable. The method has been used by [68], [45] and [28] in real-time hybrid tests.

The resulting time stepping laws in the linear case are given by Eqs. (2.46):

$$\begin{aligned}
\boldsymbol{q}^{VIR}[n+1] &= \boldsymbol{q}^{VIR}[n] + \Delta t \dot{\boldsymbol{q}}^{VIR}[n] + \Delta t^2 \frac{1}{2} \ddot{\boldsymbol{q}}^{VIR}[n] \\
\dot{\boldsymbol{q}}^{VIR}[n+1] &= \underbrace{\dot{\boldsymbol{q}}^{VIR}[n] + \Delta t(1-\gamma)\ddot{\boldsymbol{q}}^{VIR}[n]}_{\dot{\boldsymbol{q}}^{*}[n+1]} + \frac{\Delta t}{2} \ddot{\boldsymbol{q}}^{VIR}[n+1] \\
\boldsymbol{S}\ddot{\boldsymbol{q}}^{VIR}[n+1] &= \boldsymbol{G}^{VIR^T}[n+1]\boldsymbol{\lambda}^{VIR}[n+1] + \boldsymbol{f}_{ext}^{VIR}[n+1] \\
&\quad - \boldsymbol{D}^{VIR}\dot{\boldsymbol{q}}^{*}[n+1] - \boldsymbol{K}^{VIR}\boldsymbol{q}^{VIR,*}[n+1]
\end{aligned} \tag{2.46}$$

Iterations are necessary if the equations of motion are non-linear in $\dot{\boldsymbol{q}}^{VIR}$. See the Jacobian matrix from (2.53) in the non-linear case.

The task sequence of the explicit integration scheme is shown in Fig. 2.3: The displacement at time step $n+1$ only depends on the accelerations from the last time step n. Since the accelerations at time step n depend on the interface forces $\boldsymbol{\lambda}$ at time step n, only interface forces from the time step n are employed to generate the actuator command for the time step $n+1$.

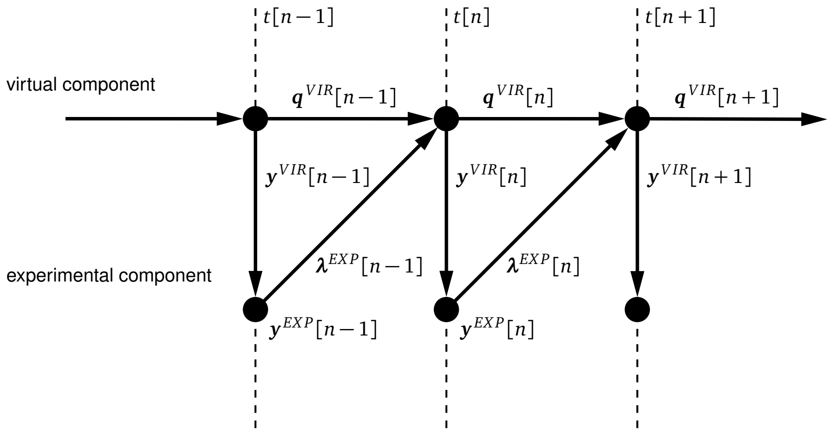

$t[n-1]$
$t[n]$
$t[n+1]$
virtual component
$\boldsymbol{q}^{VIR}[n-1]$
$\boldsymbol{q}^{VIR}[n]$
$\boldsymbol{q}^{VIR}[n+1]$
$\boldsymbol{y}^{VIR}[n-1]$
$\boldsymbol{y}^{VIR}[n]$
$\boldsymbol{y}^{VIR}[n+1]$
experimental component
$\boldsymbol{\lambda}^{EXP}[n-1]$
$\boldsymbol{\lambda}^{EXP}[n]$
$\boldsymbol{y}^{EXP}[n-1]$
$\boldsymbol{y}^{EXP}[n]$

Figure 2.3: Task sequence in a real-time test when using an explicit time integration scheme.

In [68], a staggered integration method is used, which allows performing the time integration with the time step width $2\Delta t$ and the control loop with the time step width Δt (see [96] for a discussion).

Constant-Average-Acceleration Method The Newmark parameters $\beta = 1/4$ and $\gamma = 1/2$ yield the unconditionally stable constant-average-acceleration method. The problem of the method in real-time hybrid testing is that the interface forces $\boldsymbol{\lambda}^{VIR}[n+1]$ as well as possible non-linear forces in the virtual component depend on the system state $\boldsymbol{q}^{VIR}[n+1]$. This makes iterations necessary. However, iterations which are performed on the experimental subcomponent oppose the objective of a real-time hybrid test which is to apply dynamically accurate displacements at the interface to the experimental component. The iteration laws for time step $n+1$ are given in Eq. (2.47):

$$\begin{aligned}
\ddot{\boldsymbol{q}}_{k+1} =& \ddot{\boldsymbol{q}}_k + \Delta\ddot{\boldsymbol{q}}_k \\
\dot{\boldsymbol{q}}_{k+1} =& \dot{\boldsymbol{q}}_k + \gamma\Delta t\Delta\ddot{\boldsymbol{q}}_k \\
\boldsymbol{q}_{k+1} =& \boldsymbol{q}_k + \beta\Delta t^2\Delta\ddot{\boldsymbol{q}}_k \\
\text{using } \boldsymbol{0} =& \boldsymbol{r}_k + \boldsymbol{S}_k\Delta\ddot{\boldsymbol{q}}_k \\
\text{with } \boldsymbol{r}_k =& \boldsymbol{M}^{VIR}\left(\boldsymbol{q}_k\right)\ddot{\boldsymbol{q}}_k^{VIR} + \boldsymbol{f}_{nl}^{VIR}\left(\boldsymbol{q}_k^{VIR},\dot{\boldsymbol{q}}_k^{VIR}\right) - \boldsymbol{G}^{VIR}\left(\boldsymbol{q}_k^{VIR}\right)^T\boldsymbol{\lambda}_k^{VIR} + \boldsymbol{f}_{ext}^{VIR} \\
\text{and } \boldsymbol{S}_k =& \boldsymbol{M}^{VIR}\left(\boldsymbol{q}_k\right) \\
&+ \beta\Delta t^2\left(\frac{\partial\boldsymbol{M}^{VIR}\left(\boldsymbol{q}_k\right)\ddot{\boldsymbol{q}}_k}{\partial\boldsymbol{q}_k} + \frac{\partial\boldsymbol{f}_{nl}^{VIR}\left(\boldsymbol{q}_k^{VIR},\dot{\boldsymbol{q}}_k^{VIR}\right)}{\partial\boldsymbol{q}_k} - \frac{\partial\boldsymbol{G}^{VIR}\left(\boldsymbol{q}_k^{VIR}\right)^T\boldsymbol{\lambda}_k^{VIR}}{\partial\boldsymbol{q}_k}\right) \\
&+ \gamma\Delta t\left(\frac{\partial\boldsymbol{f}_{nl}^{VIR}\left(\boldsymbol{q}_k^{VIR},\dot{\boldsymbol{q}}_k^{VIR}\right)}{\partial\dot{\boldsymbol{q}}_k} - \frac{\partial\boldsymbol{G}^{VIR}\left(\boldsymbol{q}_k^{VIR}\right)^T\boldsymbol{\lambda}_k^{VIR}}{\partial\dot{\boldsymbol{q}}_k}\right)
\end{aligned} \tag{2.47}$$

The procedure to apply an implicit time integration scheme to real-time hybrid testing is described in [96] and a similar approach has been described by [13]: In order to circumvent the problems related to the iterative process in the implicit time integration scheme, a sub-stepping approach is applied. The approach is visualized in Fig. 2.4. The integration time steps with time step width Δt are split into substeps which correspond to the controller time step width. The idea is to perform one iteration step at one sub-step k. The iteration is performed using the measured interface forces for calculating the residual forces but using an approximation of the overall system dynamics to compute the approximated Jacobian matrix $\boldsymbol{S}_k^{approx}$. In the approximated iteration step, $\boldsymbol{S}_k^{approx}$ is used instead of $\boldsymbol{S}_k$. The result from the approximated iteration step is the trial interface displacement $\boldsymbol{y}_k^{VIR}$. The actuator command $\boldsymbol{y}_k^{d,VIR}$ is computed by quadratic interpolation using the trial interface displacement $\boldsymbol{y}_k^{VIR}$ and the preceding

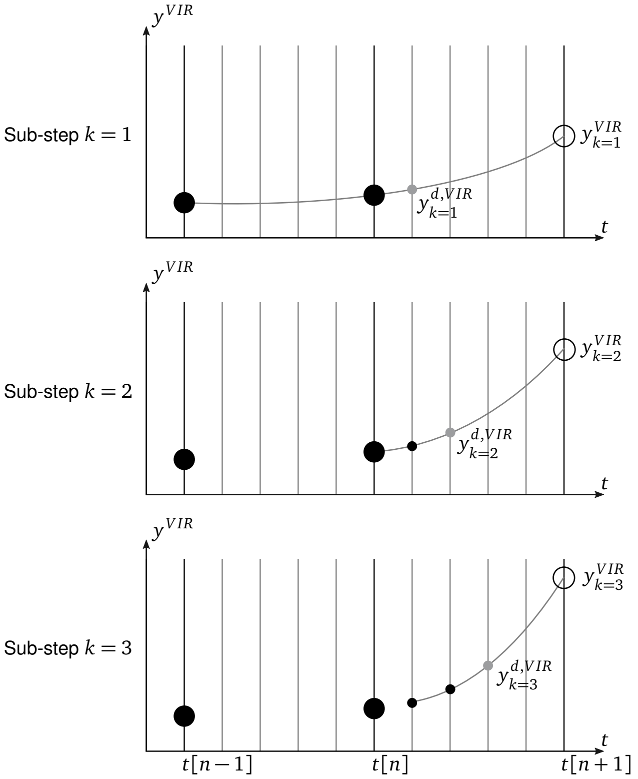


Figure 2.4: Constant-average-acceleration method with sub-stepping

actuator command values.

In contrast, when using the feedforward based control laws of Chap. 5, the constant-average-acceleration method can be applied without limitations. The time stepping laws in the case of linear equations of motion are given in eq. (2.48):

$$\begin{aligned}
\boldsymbol{q}^{VIR}[n+1] &= \underbrace{\boldsymbol{q}^{VIR}[n]+\Delta t\dot{\boldsymbol{q}}^{VIR}[n]+\frac{\Delta t^2}{4}\ddot{\boldsymbol{q}}^{VIR}[n]}_{\boldsymbol{q}^*[n+1]}+\frac{\Delta t^2}{4}\ddot{\boldsymbol{q}}^{VIR}[n+1] \\
\dot{\boldsymbol{q}}^{VIR}[n+1] &= \underbrace{\dot{\boldsymbol{q}}^{VIR}[n]+\frac{\Delta t}{2}\ddot{\boldsymbol{q}}^{VIR}[n]}_{\dot{\boldsymbol{q}}^*[n+1]}+\frac{\Delta t}{2}\ddot{\boldsymbol{q}}^{VIR}[n+1] \\
\boldsymbol{S}\ddot{\boldsymbol{q}}^{VIR}[n+1] &= \boldsymbol{G}^{VIR^T}[n+1]\boldsymbol{\lambda}^{VIR}[n+1]+\boldsymbol{f}^{VIR}_{ext}[n+1] \\
&\quad -\boldsymbol{D}^{VIR}\dot{\boldsymbol{q}}^{VIR,*}[n+1]-\boldsymbol{K}^{VIR}\boldsymbol{q}^{VIR,*}[n+1]
\end{aligned} \tag{2.48}$$

Fig. 2.5 shows the corresponding task sequence of the implicit time integration scheme. Note that the case with the control time step width being equal to the integration time step width is considered. At time step $n+1$, the interface forces $\boldsymbol{\lambda}^{VIR}[n+1]$ and the interface displacements $\boldsymbol{y}^{EXP}[n+1]$ are measured. The measurement of the interface forces $\boldsymbol{\lambda}^{VIR}[n+1]$ has to be used to compute the state vector of the virtual component $\boldsymbol{q}^{VIR}[n+1]$. However, $\boldsymbol{\lambda}^{VIR}[n+1]$ depends on $\boldsymbol{y}^{EXP}[n+1]$, which is not available when the actuator command is applied. Using adaptive feedforwad control, the interface displacements $\boldsymbol{y}^{VIR}[n]$ and $\boldsymbol{y}^{EXP}[n]$ are used to compute the interface gap $\boldsymbol{g}[n]$. The interface gap, in turn, is used to compute the actuator input at time step $n+1$.

Since the objective of the adaptive feedforward filter is to close the interface gap at all time steps, it does not deteriorate the performance of the algorithm to use the resulting interface gap $\boldsymbol{g}[n]$ from the previous time step as an input to the adaptation law at time step $n+1$. The adaptation law adjusts the

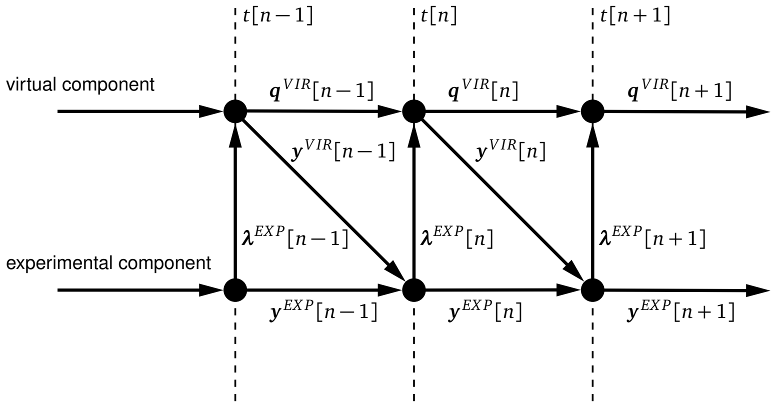


Figure 2.5: Task sequence in a real-time test when using an implicit time integration scheme.

actuator inputs such that the synchronization—at all time steps—improves iteratively from time step to time step. The adaptive algorithm depends only on the interface states from previous time steps.
The interface forces $\boldsymbol{\lambda}^{VIR}[n+1]$ are assumed to be independent of the system state $\boldsymbol{q}^{VIR}[n+1]$ for the integrator. This makes the problematic iteration on the interface forces unnecessary. The adaptive algorithm can be seen as a time-step-wise iteration on the interface forces.

2.3.4 Simulated tests

For the purpose of simulating hybrid tests it is important to compute the responses and the interface forces of the system assembled from the actuation system and the experimental component. The problem is described by a differential algebraic equation (DAE). One possibility to solve DAEs is the application of implicit Newmark time integration, which is described in the following section. For readability, the uncoupled system matrices, the interface gap, the state vectors, and the forces are written in the form of Eqs. (2.49):

$$\begin{aligned} \boldsymbol{M}(\boldsymbol{q}) &= \begin{bmatrix} \boldsymbol{M}^{EXP} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}^{ACT} \end{bmatrix}, \qquad \boldsymbol{G}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{G}^{EXP}\left(\boldsymbol{q}^{EXP}\right) & \boldsymbol{G}^{ACT}\left(\boldsymbol{q}^{ACT}\right) \end{bmatrix} \\ \boldsymbol{f}\left(\boldsymbol{q},\dot{\boldsymbol{q}}\right) &= \begin{bmatrix} \boldsymbol{f}_{ext}^{EXP} - \boldsymbol{f}_{nl}^{EXP}\left(\boldsymbol{q}^{EXP},\dot{\boldsymbol{q}}^{EXP}\right) \\ \boldsymbol{f}_{ext}^{ACT} - \boldsymbol{f}_{nl}^{ACT}\left(\boldsymbol{q}^{ACT},\dot{\boldsymbol{q}}^{ACT}\right) \end{bmatrix}, \qquad \text{with} \quad \boldsymbol{q} = \begin{bmatrix} \boldsymbol{q}^{EXP} \\ \boldsymbol{q}^{ACT} \end{bmatrix} \\ \boldsymbol{g}(\boldsymbol{q}) &= \boldsymbol{G}^{EXP}\left(\boldsymbol{q}^{EXP}\right)\boldsymbol{q}^{EXP} + \boldsymbol{G}^{ACT}\left(\boldsymbol{q}^{ACT}\right)\boldsymbol{q}^{ACT} \end{aligned} \tag{2.49}$$

The approach corresponds to the implicit time integration for non-linear systems as it is described in Sec. 2.3.2. Again, the iteration step is indicated by the subscript k, and the step number $[n+1]$ is omitted. The residual vector at time step $n+1$ reads

$$\begin{aligned} \boldsymbol{r}_{eq,k} &= \boldsymbol{M}\left(\boldsymbol{q}_k\right)\ddot{\boldsymbol{q}}_k - \boldsymbol{f}\left(\boldsymbol{q}_k,\dot{\boldsymbol{q}}_k\right) - \boldsymbol{G}\left(\boldsymbol{q}_k\right)^T \boldsymbol{\lambda}_k \\ \boldsymbol{r}_{const,k} &= \boldsymbol{G}(\boldsymbol{q}_k)\boldsymbol{q}_k. \end{aligned} \tag{2.50}$$

The iteration updates at time step $n+1$ are

$$\begin{aligned} \ddot{\boldsymbol{q}}_{k+1} &= \ddot{\boldsymbol{q}}_k + \Delta\ddot{\boldsymbol{q}}_k \\ \boldsymbol{q}_{k+1} &= \boldsymbol{q}_k + \beta\Delta t^2\Delta\ddot{\boldsymbol{q}}_k \\ \dot{\boldsymbol{q}}_{k+1} &= \dot{\boldsymbol{q}}_k + \gamma\Delta t\Delta\ddot{\boldsymbol{q}}_k \\ \boldsymbol{\lambda}_{k+1} &= \boldsymbol{\lambda}_k + \Delta\boldsymbol{\lambda}_k. \end{aligned} \tag{2.51}$$

The first-order Taylor approximations of the residuals at the next time step are

$$\begin{aligned} \boldsymbol{r}_{eq,k+1} &\approx \boldsymbol{r}_{eq,k} + \boldsymbol{S}_k \Delta\ddot{\boldsymbol{q}}_k + \boldsymbol{G}(\boldsymbol{q}_k)^T \Delta\boldsymbol{\lambda}_k \\ \boldsymbol{r}_{const,k+1} &\approx \boldsymbol{r}_{const,k} + \boldsymbol{G}(\boldsymbol{q}_k)\Delta\boldsymbol{q}_k \end{aligned} \tag{2.52}$$

using the Jacobian matrix

$$\boldsymbol{S}_k = \boldsymbol{M}\left(\boldsymbol{q}_k\right) + \gamma\Delta t \frac{\partial \boldsymbol{f}_{nl}\left(\boldsymbol{q}_k, \dot{\boldsymbol{q}}_k\right)}{\partial \dot{\boldsymbol{q}}_k} + \beta\Delta t^2 \left(\frac{\partial \boldsymbol{M}\left(\boldsymbol{q}_k\right)\ddot{\boldsymbol{q}}_k}{\partial \boldsymbol{q}_k} + \frac{\partial \boldsymbol{f}_{nl}\left(\boldsymbol{q}_k, \dot{\boldsymbol{q}}_k\right)}{\partial \boldsymbol{q}_k} - \frac{\partial \boldsymbol{G}\left(\boldsymbol{q}_k\right)^T \boldsymbol{\lambda}_k}{\partial \boldsymbol{q}_k} \right). \tag{2.53}$$

The equations for the Newton-Raphson iterations are obtained by setting the residuals of the next iteration step to zero: The corrections of the interface forces $\Delta\boldsymbol{\lambda}_k$ and the corrections of the acceleration vector $\Delta\ddot{\boldsymbol{q}}_k$ can be computed by solving Eq. (2.54):

$$\begin{bmatrix} \boldsymbol{S}_k & \boldsymbol{G}(\boldsymbol{q}_k)^T \\ \boldsymbol{G}(\boldsymbol{q}_k) & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \Delta\ddot{\boldsymbol{q}}_k \\ \Delta\boldsymbol{\lambda}_k \end{bmatrix} = \begin{bmatrix} -\boldsymbol{r}_{eq,k} \\ -r_{const,k}/\beta\Delta t^2 \end{bmatrix} \tag{2.54}$$

The states and interface forces are then corrected according to Eq. (2.51). The iterations can be repeated until the norms of the residuals fall below the desired values.

HHT-α Method for Coupled Systems

The constraints introduced by the coupling can cause an unstable behavior even for the unconditionally stable implicit Newmark time integration schemes. The reason is that constraints are dynamic equations which can add infinite resonance frequencies to the system. In order to solve the problem, [42] propose a time integration scheme which is based on the Newmark time integration but introduces numerical damping for the high frequencies. The Newmark approximation formulas are not changed, but the residuals are evaluated by interpolation between two time steps using the parameter α:

$$\begin{aligned} \boldsymbol{r}_{eq,k} =& \boldsymbol{M}\ddot{\boldsymbol{q}}[n+1] + (1-\alpha)\left(\boldsymbol{f}(\boldsymbol{q}[n+1], \dot{\boldsymbol{q}}[n+1]) + \boldsymbol{G}(\boldsymbol{x}[n+1])^T \boldsymbol{\lambda}[n+1]\right) \\ &+ \alpha\left(\boldsymbol{f}(\boldsymbol{q}[n], \dot{\boldsymbol{q}}[n]) + \boldsymbol{G}(\boldsymbol{x}[n])^T \boldsymbol{\lambda}[n]\right) \\ \boldsymbol{r}_{const,k} =& (1-\alpha)\boldsymbol{g}(\boldsymbol{x}[n+1]) + \alpha\boldsymbol{g}(\boldsymbol{x}[n]) \end{aligned} \tag{2.55}$$

The Newmark parameters are

$$\gamma = \frac{1}{2} + \alpha \quad \text{and} \quad \beta = \frac{1}{4}(1+\alpha)^2 \quad \text{with} \quad \alpha \in \left[0, \frac{1}{3}\right]. \tag{2.56}$$

The method is unconditionally stable and the damping is constrained to higher frequencies.

2.4 Time Discrete Filters

As shown in Sec. 2.1, impulse responses can be used to describe the dynamics of a linear mechanical system. The system responses can be computed using convolution integrals with impulse responses. The use of impulse responses in off-line substructure coupling has been described in [37], [86] and [88]. The results of the research show that impulse responses are an especially efficient system representation when a broad spectral bandwidth is analyzed or the substructures exhibit high modal densities. Broad spectral bandwidths occur e.g. in shock responses. High modal densities tend to occur in complex systems. Impulse responses have been used in order to represent flexible bodies in multi-body systems as proposed by [36]. In the specific problem of real-time hybrid testing interface displacements of the virtual component read

$$\boldsymbol{y}^{VIR} = \boldsymbol{H}_{\lambda}^{VIR}(t) * \boldsymbol{\lambda}^{VIR}(t) + \boldsymbol{H}_{ext}^{VIR}(t) * \boldsymbol{f}_{exp}^{VIR}(t) \tag{2.57}$$

where two types of inputs are distinguished: $\boldsymbol{H}_{ext}^{VIR}(t)$ contains the impulse responses at the interface to the external forces $\boldsymbol{f}_{ext}^{VIR}(t) \in \mathbb{R}^{N_{ext}}$ and $\boldsymbol{H}_{\lambda}^{VIR}(t)$ contains the impulse responses to the interface forces $\boldsymbol{\lambda}^{VIR}(t) \in \mathbb{R}^{N_{\lambda}}$. In the following section, the computation of impulse responses and their use in real-time hybrid testing is described.

2.4.1 Computing Impulse Responses

Impulse responses can be computed using Newmark's method as it is described in Sec. 2.3. The discretized impulse responses at the interface are defined by

$$\begin{aligned} \boldsymbol{H}_{ext}^{VIR}[n] &= \boldsymbol{G}^{VIR}\left[\boldsymbol{q}_{1,ext}^{VIR}[n] \quad \boldsymbol{q}_{2,ext}^{VIR}[n] \quad \cdots \quad \boldsymbol{q}_{N_{ext},ext}^{VIR}[n]\right] \quad \text{and} \\ \boldsymbol{H}_{\lambda}^{VIR}[n] &= \boldsymbol{G}^{VIR}\left[\boldsymbol{q}_{1,\lambda}^{VIR}[n] \quad \boldsymbol{q}_{2,\lambda}^{VIR}[n] \quad \cdots \quad \boldsymbol{q}_{N_{\lambda},\lambda}^{VIR}[n]\right] \end{aligned} \tag{2.58}$$

where $\boldsymbol{q}_{j,ext}^{VIR}[n]$ is the response of the full coordinate set to a Dirac impulse on the j-th entry of the external forces vector $\boldsymbol{f}_{ext}^{VIR}$ at time step n. $\boldsymbol{q}_{j,\lambda}^{VIR}[n]$ is the response of the full coordinate set to a Dirac impulse on the j-th entry of the interface forces vector $\boldsymbol{\lambda}^{VIR}$. The question of how to approximate the Dirac impulses remains: a Dirac impulse in the continuous formulation is an impulse with infinitely short time duration and an infinitely large amplitude. The integral over the impulse is one:

$$\int_{-\infty}^{\infty} \delta(t)\mathrm{d}t = 1 \tag{2.59}$$

In the time discrete form the Dirac impulse has to be formulated in a time sampled form, but the force integral has to remain one. [88] presents three approaches to approximating the Dirac impulse for the computation of the impulse responses: an equivalent velocity step, a force impulse at the first time step and a force impulse at the second time step. If the time step width goes against zero, all approaches are equivalent. The three approximations will be described in the following paragraphs. After the initial step—which contains the impulse—the time integration can proceed with Eqs. (2.39) and Eq. (2.40).
In the following equations, $\mathbf{1}_j$ is a vector of zeros with a one at position j. The superscript VIR is omitted for readability reasons.

Equivalent Velocity Step The idea is to apply a velocity step at the first time step based on the momentum equation, which compares a point in time directly before $t = 0$ and a point in time directly after $t = 0$. Since the time span is infinitely small, the contributions from stiffness and damping can be neglected. The momentum equation reads

$$\boldsymbol{M}\left(\dot{\boldsymbol{q}}(0^+) - \dot{\boldsymbol{q}}(0^-)\right) = \int_{0^-}^{0^+} \boldsymbol{f}_{ext}(t) + \boldsymbol{G}^T\boldsymbol{\lambda}(t)\mathrm{d}t. \tag{2.60}$$

Because the velocity $\dot{\boldsymbol{q}}(0^-)$ before time $t = 0$ is zero, the initial states are

$$\begin{aligned} &\boldsymbol{q}[0] = \mathbf{0} \\ &\dot{\boldsymbol{q}}[0] = \boldsymbol{M}^{-1}\mathbf{1}_j \text{ for } \boldsymbol{H}_{ext} \quad \text{and} \quad \dot{\boldsymbol{q}}[0] = \boldsymbol{M}^{-1}\boldsymbol{G}^T\mathbf{1}_j \text{ for } \boldsymbol{H}_{\lambda} \\ &\ddot{\boldsymbol{q}}[0] = -\boldsymbol{M}^{-1}\boldsymbol{D}\dot{\boldsymbol{q}}[0] \text{ (from the equations of motion)} \\ &\boldsymbol{f}[n] = \mathbf{0}\ \forall\ n \in \mathbb{N} \end{aligned} \tag{2.61}$$

Force Impulse at First Time Step The idea is to apply a force with unit integral at the first time step. The discretized force profile is shown in Fig. 2.6. Since the integral of the force has to be equal to one, the peak of the impulse has the amplitude $\frac{2}{\Delta t}$. Correspondingly, the initial states are:

$$\begin{aligned}
&\boldsymbol{q}[0] = \mathbf{0}\\
&\dot{\boldsymbol{q}}[0] = \mathbf{0}\\
&\ddot{\boldsymbol{q}}[0] = \frac{2}{\Delta t}\boldsymbol{M}^{-1}\mathbf{1}_j \text{ for } \boldsymbol{H}_{ext} \quad \text{and} \quad \ddot{\boldsymbol{q}}[0] = \frac{2}{\Delta t}\boldsymbol{M}^{-1}\boldsymbol{G}^T\mathbf{1}_j \text{ for } \boldsymbol{H}_\lambda\\
&\boldsymbol{f}[0] = \frac{2}{\Delta t}\mathbf{1}_j \text{ for } \boldsymbol{H}_{ext} \quad \text{and} \quad \boldsymbol{f}[0] = \frac{2}{\Delta t}\boldsymbol{G}^T\mathbf{1}_j \text{ for } \boldsymbol{H}_\lambda\\
&\boldsymbol{f}[n] = \mathbf{0}\ \forall\ n \in \{\mathbb{N} \setminus 0\}
\end{aligned} \tag{2.62}$$

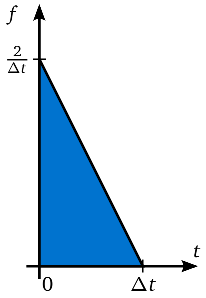


Figure 2.6: Force impulse at time $t = 0$ (modified from [67])

Force Impulse at Second Time Step The factorizing of the mass matrix—as it is necessary in the two above-mentioned methods—can be avoided by applying the force impulse at the second time step. The corresponding discretized force profile is shown in Fig. 2.7. Since the integral of the force has to be equal to one, the peak of the impulse has the amplitude $\frac{1}{\Delta t}$. The initial states are:

$$\begin{aligned}
&\boldsymbol{q}[0] = \mathbf{0}\\
&\dot{\boldsymbol{q}}[0] = \mathbf{0}\\
&\ddot{\boldsymbol{q}}[0] = \mathbf{0}\\
&\boldsymbol{f}[1] = \frac{1}{\Delta t}\mathbf{1}_j \text{ for } \boldsymbol{H}_{ext} \quad \text{and} \quad \boldsymbol{f}[1] = \frac{1}{\Delta t}\boldsymbol{G}^T\mathbf{1}_j \text{ for } \boldsymbol{H}_\lambda\\
&\boldsymbol{f}[n] = \mathbf{0}\ \forall\ n \in \{\mathbb{N} \setminus 1\}
\end{aligned} \tag{2.63}$$

Because the peak of the impulse is shifted to $t = \Delta t$, the impulse response has to be shifted back by one time step. Note that the impulse responses can also be retrieved directly from measurements as it is

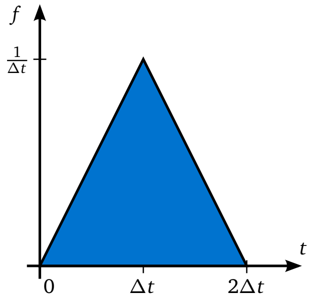


Figure 2.7: Force impulse at time $t = \Delta t$ (modified from [67])

proposed in [87]. In this case, the impulse responses are retrieved by applying a deconvolution process on the measured impulse responses and the measured force. impulses.

2.4.2 Finite Impulse Response Filters

Finite impulse response (FIR) filters are time discrete filters which are implemented without feedback from their output. The filters are referred to as non-recursive. The output of an FIR-filter is the sum of a finite number of filter coefficients multiplied with the input signal. For this reason, FIR-filters are inherently stable. It makes them an appealing candidate for the simulation of the virtual component in real-time hybrid testing: If the impulse responses—which can be used to describe the input/output behavior of the virtual component—are truncated after a finite number of coefficients, the time-discrete convolution products can be seen as FIR-filters. Recalling Sec. 2.1, the time-discretization of the convolution product which describes the virtual component is

$$\boldsymbol{y}[n] = \sum_{k=0}^{n-1} \boldsymbol{H}_\lambda[n-k]\boldsymbol{\lambda}[k]\Delta t + \sum_{k=0}^{n-1} \boldsymbol{H}_{ext}[n-k]\boldsymbol{f}_{ext}[k]\Delta t.$$

Since the convolution operation is commutative, the two sums can be rearranged such that the input and the impulse response are exchanged. Using the definitions

$$\boldsymbol{b}[k] := \begin{bmatrix} \boldsymbol{H}_\lambda[k] & \boldsymbol{H}_{ext}[k] \end{bmatrix} \Delta t \quad \text{and} \quad \boldsymbol{u}[k] := \begin{bmatrix} \boldsymbol{\lambda}[k]^T & \boldsymbol{f}_{ext}[k]^T \end{bmatrix}^T, \tag{2.64}$$

the sum can be split into two parts:

$$\boldsymbol{y}[n] = \sum_{k=1}^{n} \boldsymbol{b}[k]\boldsymbol{u}[n-k] = \underbrace{\sum_{k=1}^{M} \boldsymbol{b}[k]\boldsymbol{u}[n-k]}_{\text{FIR-filter}} + \underbrace{\sum_{k=M+1}^{n} \boldsymbol{b}[k]\boldsymbol{u}[n-k]}_{\text{approximation error}} \tag{2.65}$$

The first part is used as FIR-filter and the second part is the approximation error due to the truncation of the impulse response. The M-th order FIR-filter is

$$\boldsymbol{y}[n] = \sum_{k=1}^{M} \boldsymbol{b}[k]\boldsymbol{u}[n-k] \tag{2.66}$$

The approximation is valid if the impulse response has disappeared at time sample M. This makes an efficient implementation of the virtual component feasible. This type of filter is especially helpful in real-time hybrid testing, because the output at time sample n depends only on the inputs from time sample $n-1$ and previous time samples.

Using the trapezoidal rule for approximation of the input signal, the discretization of the convolution integral reads

$$\begin{aligned} \boldsymbol{y}[n] = &\sum_{k=0}^{n-1} \boldsymbol{H}_\lambda[n-k](\boldsymbol{\lambda}[k] + \boldsymbol{\lambda}[k+1])\frac{\Delta t}{2} \\ &+ \sum_{k=0}^{n-1} \boldsymbol{H}_{ext}[n-k](\boldsymbol{f}_{ext}[k] + \boldsymbol{f}_{ext}[k+1])\frac{\Delta t}{2}. \end{aligned} \tag{2.67}$$

As shown in [94], it can easily be verified that the time discrete convolution integral can be rewritten as

$$\boldsymbol{y}[n] = \sum_{k=1}^{n} \boldsymbol{b}[k]\boldsymbol{u}[n-k] + \boldsymbol{H}_\lambda[n]\boldsymbol{\lambda}[0]\frac{\Delta t}{2} + \boldsymbol{H}_{ext}[n]\boldsymbol{f}_{ext}[0]\frac{\Delta t}{2} \tag{2.68}$$

$$= \underbrace{\sum_{k=1}^{M} \boldsymbol{b}[k]\boldsymbol{u}[n-k]}_{\text{FIR-filter}} + \underbrace{\sum_{k=M+1}^{n} \boldsymbol{b}[k]\boldsymbol{u}[n-k] + \boldsymbol{H}_\lambda[n]\boldsymbol{\lambda}[0]\frac{\Delta t}{2} + \boldsymbol{H}_{ext}[n]\boldsymbol{f}_{ext}[0]\frac{\Delta t}{2}}_{\text{approximation error}} \tag{2.69}$$

using the definition

$$\boldsymbol{b}[k] := \left[\boldsymbol{H}_{\lambda}[k] + \boldsymbol{H}_{\lambda}[k+1] \quad \boldsymbol{H}_{ext}[k] + \boldsymbol{H}_{ext}[k+1]\right] \frac{\Delta t}{2}. \tag{2.70}$$

The corresponding M-th order FIR-filter is

$$\boldsymbol{y}[n] = \sum_{k=1}^{M} \boldsymbol{b}[k]\,\boldsymbol{u}[n-k]. \tag{2.71}$$

The sum can be split into a predictor and a corrector: The major part of the convolution can be performed using the input values which are available at time step $n-1$. In the corrector step, the response at time step n is computed with a single additional matrix multiplication. The predictor-corrector form reads

$$\boldsymbol{y}^{*}[n] = \sum_{k=2}^{M} \boldsymbol{b}[k]\,\boldsymbol{u}[n-k] \tag{2.72}$$

$$\boldsymbol{y}[n] = \boldsymbol{y}^{*}[n] + \boldsymbol{b}[1]\,\boldsymbol{u}[n-1]. \tag{2.73}$$

The predictor-corrector approach is helpful in time critical applications such as in real-time hybrid testing. The number of multiplications in the convolution—and as a consequence the computational time— grows linearly with the simulation time. Correspondingly, the memory usage by the input signals values and filter coefficients grows linearly with the simulation time. As mentioned above, the convolution product can be truncated at the point where the impulse response effectively has damped out ($\boldsymbol{H}[n] \approx 0$ for $n > M$). For floating structures, the impulse response never disappears. In this case, the rigid body modes have to be projected out, such that only flexible modes remain in the impulse response. This approach is described in Sec. 2.4.3. According to [89], a sudden cut-off of the impulse response which is caused by a truncation can generate perturbations of the output signal. The truncated impulse $\boldsymbol{H}_{*}[n]$ response can be written as

$$\boldsymbol{H}_{*}[n] = \boldsymbol{H}[n] \cdot w_{*}[n]$$

where $w_{*}[n]$ is a window function. The cut-off corresponds to a rectangular window of Eq. (2.74) as shown in Fig. 2.8.

$$w_{rect}[n] = \begin{cases} 1 & \text{for} \quad n \leq M \\ 0 & \text{for} \quad n > M \end{cases} \tag{2.74}$$

[89] propose to use smoothing of the truncation. Two examples of smooth window functions—a cosine window and a Hann window—are given in Eqs. (2.75) and (2.76).

$$w_{cos}[n] = \begin{cases} \cos(\frac{\pi n}{2M}) & \text{for} \quad n \leq M \\ 0 & \text{for} \quad n > M \end{cases} \tag{2.75}$$

$$w_{hann}[n] = \begin{cases} \cos^{2}(\frac{\pi n}{2M}) & \text{for} \quad n \leq M \\ 0 & \text{for} \quad n > M \end{cases} \tag{2.76}$$

Examples of the resulting truncated impulse responses are shown in Fig. 2.8.

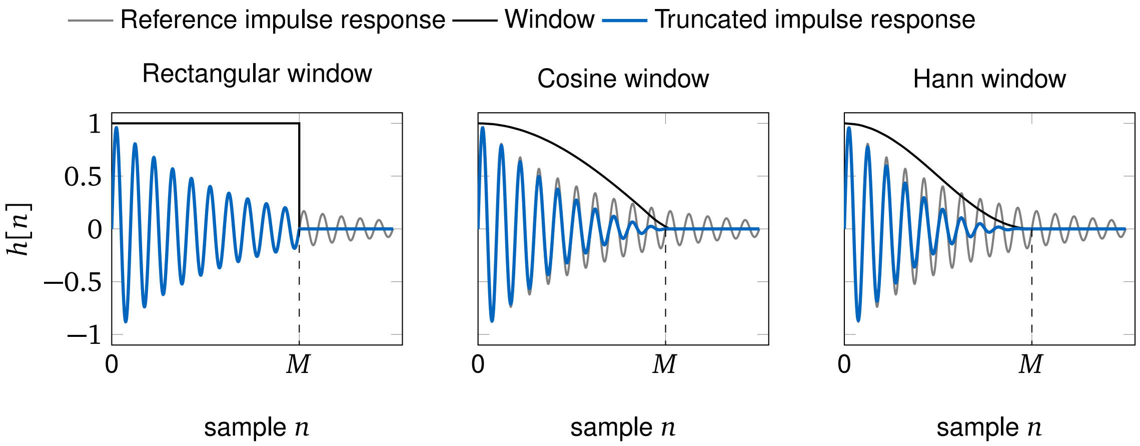


Figure 2.8: Windows for smoothening the impulse response truncation (modified from [19])

2.4.3 Rigid-Body Modes and Modes with Low Damping

Time integration of modal truncated or reduced complex systems with a high modal density can be less efficient than the computation of impulse-response convolution integrals. However, impulse responses, which do not decay or decay only slowly, can only be represented by a FIR-filter with infinite length or a large number of coefficients. Even for simple systems with only few DOFs, the computational effort only depends on the length of the impulse response. Since the high frequency content often damps out fast, the length of the impulse responses is dominated by low frequency contributions. [94] proposed to split the responses into contributions of rigid-body modes and slowly decaying modes, and into contribution from higher frequency modes which decay faster. Using the definitions of the free interface modes in Sec. 2.2.2 and their corresponding eigenfrequencies ω_i, all modes can split in three groups according to Eq. (2.77)

$$\boldsymbol{\Phi}_f = [\boldsymbol{\Phi}_R \, \boldsymbol{\Phi}_X \, \boldsymbol{\Phi}_V] \quad \text{with} \quad \begin{cases} \boldsymbol{\Phi}_R & \text{for } \omega_i = 0 \\ \boldsymbol{\Phi}_X & \text{for } 0 < \omega_i < \omega_X \\ \boldsymbol{\Phi}_V & \text{for } \omega_i \geq \omega_X \end{cases}, \tag{2.77}$$

where $\boldsymbol{\Phi}_R$ contains the rigid body modes, $\boldsymbol{\Phi}_X$ contains the elastic modes with low eigenfrequencies, $\boldsymbol{\Phi}_V$ contains modes with high eigenfrequencies and ω_X is the frequency which separates the high frequency modes from the low frequency modes. ω_X can be chosen freely and a problem specific choice is necessary. Using the above-mentioned definitions, the response $\boldsymbol{q}$ is split into the contribution $\boldsymbol{l}(t)$ from the rigid body modes and low frequency modes on the one hand, and the contribution $\boldsymbol{r}(t)$ of the high frequency modes on the other hand:

$$\boldsymbol{q}(t) = \underbrace{\boldsymbol{\Phi}_R \, \boldsymbol{\alpha}(t) + \boldsymbol{\Phi}_X \, \boldsymbol{\beta}(t)}_{\boldsymbol{l}(t)} + \underbrace{\boldsymbol{\Phi}_V \, \boldsymbol{\gamma}(t)}_{\boldsymbol{r}(t)} = \boldsymbol{l}(t) + \boldsymbol{r}(t) \tag{2.78}$$

The objective of the approach is to filter out the contribution of the low frequency modes and the rigid body modes $\boldsymbol{l}(t)$ and to compute the contribution of the high frequency modes $\boldsymbol{r}$ using convolutions. To do so, the matrix $\boldsymbol{Q}$ is needed, which projects the response $\boldsymbol{q}$ on $\boldsymbol{r}$. $\boldsymbol{Q}$ can be directly obtained from the normalized version of $\boldsymbol{\Phi}_V^T$, since all modes in $\boldsymbol{\Phi}_f$ are orthogonal. However, assuming a high modal density, this would be computationally expensive for large systems. A more efficient strategy to retrieve $\boldsymbol{Q}$ is to compute it indirectly using the orthogonal complement $\boldsymbol{Q}^\perp$ which projects $\boldsymbol{q}$ on $\boldsymbol{l}$ such that

$$\boldsymbol{q} = \boldsymbol{l} + \boldsymbol{r} = \boldsymbol{Q}^\perp \boldsymbol{q} + \underbrace{(\boldsymbol{I} - \boldsymbol{Q}^\perp)}_{\boldsymbol{Q}} \boldsymbol{q} = \boldsymbol{Q}^\perp \boldsymbol{q} + \boldsymbol{Q} \boldsymbol{q} \, . \tag{2.79}$$

Using the definitions $\boldsymbol{\zeta} = \begin{bmatrix}\boldsymbol{\alpha}^T & \boldsymbol{\beta}^T\end{bmatrix}^T$ and $\boldsymbol{L} = [\boldsymbol{\Phi}_R \; \boldsymbol{\Phi}_X]$, the $\boldsymbol{M}$-orthogonality between $\boldsymbol{\Phi}_V$—which spans the space of $\boldsymbol{r}$—, and the matrices $\boldsymbol{\Phi}_R$ and $\boldsymbol{\Phi}_X$ can be expressed according to Eq. (2.80):

$$\boldsymbol{L}^T\boldsymbol{M}\boldsymbol{r} = \boldsymbol{L}^T\boldsymbol{M}(\boldsymbol{q}-\boldsymbol{L}\boldsymbol{\zeta}) = \boldsymbol{0} \quad \Leftrightarrow \quad \boldsymbol{L}^T\boldsymbol{M}\boldsymbol{L}\boldsymbol{\zeta} = \boldsymbol{L}^T\boldsymbol{M}\boldsymbol{q} \quad \Leftrightarrow \quad \boldsymbol{\zeta} = \boldsymbol{L}^T\boldsymbol{M}\boldsymbol{q} \tag{2.80}$$

Substitution of $\boldsymbol{\zeta}$ in the definition of $\boldsymbol{l}(t)$ from Eq. (2.78) yields the orthogonal complement $\boldsymbol{Q}^{\perp}$

$$\boldsymbol{l} = \boldsymbol{L}\boldsymbol{\zeta} = \underbrace{\boldsymbol{L}\boldsymbol{L}^T}_{\boldsymbol{Q}^{\perp}\boldsymbol{M}}\boldsymbol{q}$$

Hence, the projection matrix $\boldsymbol{Q}$ is defined by

$$\boldsymbol{Q} = \boldsymbol{I} - \boldsymbol{L}\boldsymbol{L}^T\boldsymbol{M}. \tag{2.81}$$

The system response at the interface can be split into the contributions from the low frequency modes and the high frequency modes in the same way:

$$\boldsymbol{y}(t) = \boldsymbol{G}\boldsymbol{q}(t) = \boldsymbol{G}\boldsymbol{l}(t) + \boldsymbol{G}\boldsymbol{r}(t) = \boldsymbol{y}^{\downarrow}(t) + \boldsymbol{y}^{\uparrow}(t) \tag{2.82}$$

The high frequency contribution $\boldsymbol{y}^{\uparrow}(t)$ is obtained from the state vector by premultiplying the projection matrix $\boldsymbol{Q}$. $\boldsymbol{y}^{\uparrow}(t)$ can be equivalently computed using convolution integrals of the projected impulse responses $\boldsymbol{H}^{\uparrow}_{ext}[n]$ and $\boldsymbol{H}^{\uparrow}_{\lambda}[n]$.

$$\boldsymbol{y}^{\uparrow}(t) = \boldsymbol{G}\boldsymbol{r}(t) = \boldsymbol{G}\boldsymbol{Q}\boldsymbol{q}(t) = \boldsymbol{H}^{\uparrow}_{\lambda}(t) * \boldsymbol{\lambda}(t) + \boldsymbol{H}^{\uparrow}_{ext}(t) * \boldsymbol{f}_{exp}(t)$$

The projected impulse responses are obtained in the same way as the impulse responses in Eq. (2.58) by multiplying the projection matrix $\boldsymbol{Q}$ with the state vectors.

$$\begin{aligned} \boldsymbol{H}^{\uparrow}_{ext}[n] &= \boldsymbol{G}\boldsymbol{Q}\begin{bmatrix}\boldsymbol{q}_{1,ext}[n] & \boldsymbol{q}_{2,ext}[n] & \dots & \boldsymbol{q}_{N_{ext},ext}[n]\end{bmatrix} \quad \text{and} \\ \boldsymbol{H}^{\uparrow}_{\lambda}[n] &= \boldsymbol{G}\boldsymbol{Q}\begin{bmatrix}\boldsymbol{q}_{1,\lambda}[n] & \boldsymbol{q}_{2,\lambda}[n] & \dots & \boldsymbol{q}_{N_{\lambda},\lambda}[n]\end{bmatrix} \end{aligned} \tag{2.83}$$

The impulse responses $\boldsymbol{H}^{\uparrow}_{ext}[n]$ and $\boldsymbol{H}^{\uparrow}_{\lambda}[n]$ can be used in the high frequency FIR-filters to compute $\boldsymbol{y}^{\uparrow}(t)$. The low frequency contribution to the interface response is

$$\boldsymbol{y}^{\downarrow}(t) = \boldsymbol{G}\boldsymbol{l}(t)$$

The decoupled ODEs for the low frequency contribution are obtained by using the definition $\boldsymbol{l} = \boldsymbol{L}\boldsymbol{\zeta}$ from Eq. (2.78) and by premultiplying $\boldsymbol{L}^T$:

$$\boldsymbol{L}^T\boldsymbol{M}\boldsymbol{L}\ddot{\boldsymbol{\zeta}} + \boldsymbol{L}^T\boldsymbol{C}\boldsymbol{L}\dot{\boldsymbol{\zeta}} + \boldsymbol{L}^T\boldsymbol{K}\boldsymbol{L}\boldsymbol{\zeta} = \boldsymbol{L}^T\left(\boldsymbol{f}_{ext}(t) + \boldsymbol{G}_b^T\boldsymbol{\lambda}(t)\right) \tag{2.84}$$

$\boldsymbol{\zeta}[n]$ can then be obtained using a Newmark based time integration scheme as discussed in Sec. 2.3. The overall interface displacement is finally calculated by superposition of the result of the time integration and the result of the FIR convolution according to Eq. (2.82).

2.4.4 Infinite Impulse Response Filters

Oscillations of the low frequency and rigid-body modes—which are described in the previous section—damp out slowly. For this reason, they are not included in the FIR-filters, which represent finite impulse responses. Impulse responses can be computed using a Newmark time integration scheme. It is possible to compute the contribution of the lower frequency modes directly from time discrete filters. This type of filter is referred to as infinite impulse response (IIR) filters. IIR filters can be obtained by diagonalizing the system equations and discretizing the single DOF ODEs by using an analytic transformation. The structure of the system equations (2.84) is given in Eq. (2.85),

$$\begin{bmatrix} \ddot{\zeta}_1 \\ \ddot{\zeta}_2 \\ \vdots \\ \ddot{\zeta}_{N_L} \end{bmatrix} + \begin{bmatrix} 2\delta_1 & & & \\ & 2\delta_2 & & \\ & & \ddots & \\ & & & 2\delta_{N_L} \end{bmatrix} \begin{bmatrix} \dot{\zeta}_1 \\ \dot{\zeta}_2 \\ \vdots \\ \dot{\zeta}_{N_L} \end{bmatrix} + \begin{bmatrix} \omega_1^2 & & & \\ & \omega_2^2 & & \\ & & \ddots & \\ & & & \omega_{N_L}^2 \end{bmatrix} \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \vdots \\ \zeta_{N_L} \end{bmatrix} = \begin{bmatrix} \chi_1 \\ \chi_2 \\ \vdots \\ \chi_{N_L} \end{bmatrix}$$
$$\text{with} \quad \boldsymbol{\chi} = \begin{bmatrix} \chi_1 & \chi_2 & \cdots & \chi_{N_L} \end{bmatrix} = \boldsymbol{L}^T \left(\boldsymbol{f}_{ext} + \boldsymbol{G}_b^T \boldsymbol{\lambda} \right)^T \tag{2.85}$$

where N_L is the number of modes in the matrix $\boldsymbol{L}$. The equations of motion are uncoupled because of the diagonal form. The Laplace transform of a single line of the equations yields the transfer function between input force and modal displacements:

$$H_i(s) = \frac{1}{s^2 + 2\delta_i s + \omega_i^2} \qquad \text{für } i \in [1, \dots, N_L]. \tag{2.86}$$

This decoupled equations can now be transformed to th IIR-filters separately. In the signal processing literature, various techniques for the transformation of the continuous transfer function into a IIR-filter exist. They are summarized e.g. in the textbook of [81]. In this work, the impulse-invariance technique is applied, where the transformation is performed in three steps:

1. Transformation of the Laplace-domain transfer function into time-domain impulse response using analytic expressions

$$h_i(t) = \mathcal{L}^{-1}\{H_i(s)\} \tag{2.87}$$

2. Discretization of the impulse response with the sampling period Δt

$$h_i[n] = h_i(t = n\Delta t) \tag{2.88}$$

3. z-Transform of the time-discrete impulse response

$$h_i(z) = \mathcal{Z}\{h_i[n]\} \tag{2.89}$$

The transformation yields different results for the four different modal parameter configurations: Rigid body mode with $\omega_i = 0$ yield the transfer functions listed in Appendix A.1; underdamped modes with $\omega_i > 0$ and $\frac{\delta_i}{\omega_i} < 1$ yield the transfer functions listed in Appendix A.2; critically damped modes with $\omega_i > 0$ and $\frac{\delta_i}{\omega_i} = 1$ yield the transfer functions listed in Appendix A.3; and overdamped modes with $\omega_i > 0$ and $\frac{\delta_i}{\omega_i} > 1$ yield the transfer functions listed in Appendix A.4. The resulting transfer functions take the form

$$h_i(z) = \frac{b_1 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}}.$$

The coefficients a_1, a_2 and b_1 are given in Appendix A.5. The corresponding time stepping formula is

$$\zeta_i[n+1] = b_1 \zeta_i[n] + f_i[n+1] + a_1 f_i[n] + a_2 f_i[n-1]. \tag{2.90}$$

The time step formula allows to compute the responses of the modal coordinates. The response at the interface is obtained from projection

$$\boldsymbol{y}[n+1] = \boldsymbol{G}\boldsymbol{L}\boldsymbol{\zeta}[n+1]. \tag{2.91}$$

	multiplications	additions	divisions
Newmark	$\mathcal{O}(N^2)$	$\mathcal{O}(N^2)$	$\mathcal{O}(N)$
IIR-Filter	$\mathcal{O}(4N_L)$	$\mathcal{O}(3N_L)$	-
FIR-Filter	$\mathcal{O}(M)$	$\mathcal{O}(M)$	-

Table 2.2: Orders of computational operations in one time step.

	FIR/IIR	Newmark	HHT-alpha
sampling time Δt	$0,5\,\mathrm{ms}$	$0,5\,\mathrm{ms}$	$0,5\,\mathrm{ms}$
size of full coordinate set N	–	8066	8066
number of low frequency modes N_L	11	–	–
number of FIR coefficients M	58	–	–

Table 2.3: Configuration for the different algorithms in Fig. 2.9(modified from [67])

2.4.5 Real-Time Capability

In the previous section, various algorithms for time-stepping were presented. If the computations—which are necessary to retrieve the response at the following time step—can be executed within one-time step, a time stepping algorithm is referred to as real-time capable. A theoretical prediction of the real-time capability is a complex task since it depends on many factors like the architecture of the target system or the compiler. However, the choice of the algorithm and its specific number of computational operations has a large impact on the total computation time. Tab. 2.2 gives an overview on computational effort of three time stepping algorithms.

The order of the number of operations performed by Newmark algorithms depends on the size N of the coordinate sets. The number of computational operations performed by IIR filters mainly depends on the number of modes N_L which are included in the filter. In contrast, the order of the number of computational operations performed by an FIR-filter in one time step depends on the number of filter coefficients M. The truncation of the convolution product defines the number of filter coefficients. Since the low damping rates limit the truncation, damping indirectly affects the computational complexity of FIR-filters whereas Newmark algorithms and IIR filters are unaffected. The modal density, in turn, affects the computational effort for time integration of reduced order systems, whereas FIR-filters are unaffected by the modal density. Note that the structure of FIR-filters also allows for parallelization, which can positively influence the computational effort.

A 8066-DOF finite element model of an aluminum plate with three input channels and three output channels serves as a demonstration example: The average computation times for one time step are given in Fig. 2.9. An overview of the configuration for the different algorithms is given in Tab. 2.3. Five elastic modes and six rigid-body modes were included in the IIR filter. The computations were performed running in Matlab interpreter mode. The FIR/IIR program was compiled in order to indicate the performance of the compiled filters. The resulting time domain plot for the displacement of one reference point is given in Fig. 2.10. All average computation times in Fig. 2.9 are based on 100 simulation runs. The results show the potential performance gain by the FIR/IIR approach. Note, however, that the evaluation performance of the different approaches is difficult. It is highly dependent on problem size and damping properties. Reduced order models can improve the performance of the time integration approach significantly—if slight errors are acceptable. Hence, the example is not representative of all problems.

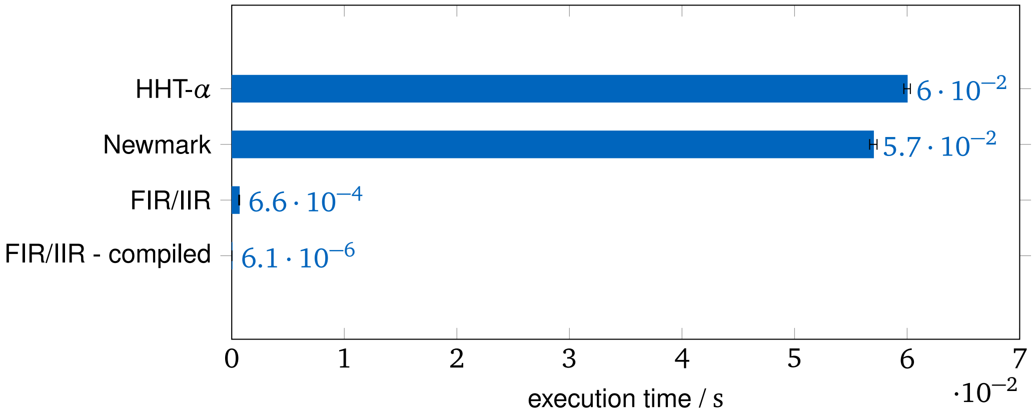


Figure 2.9: Average computation times for one time step using an example system (data from [67]). The black bars indicate the 95%-confidence intervals. (Newmark: $\beta = 0.25$ $\gamma = 0.5$ HHT-α: $\alpha = 0.3$ $\beta = 0.42$ $\gamma = 0.8$)

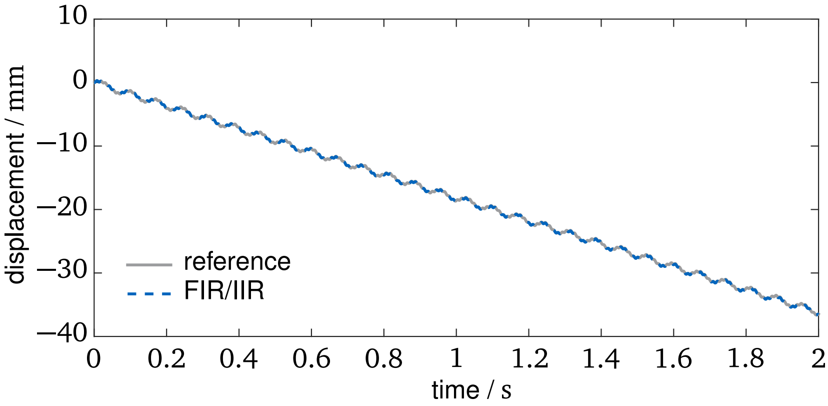


Figure 2.10: Comparison of reference simulation using commercial software and FIR/IIR-filters (data from [67])

Chapter 3

Actuation Systems

Parts of this chapter have been published in [5] and submitted for publication in [7].

It is inherent to any real-time hybrid test to couple virtual and experimental subcomponents. In Chap. 2, the implementation of the virtual component and simulation models was discussed. This chapter discusses the link between the physical part of the test and the computer system, which runs the virtual component. In general, actuators apply forces or displacements on the experimental component and sensors are used to measure interface displacements and reaction forces. The overall system which consists of sensors and actuators is referred to as the actuation system in this thesis. The superscript ACT is used to label variables belonging to actuation systems. In literature, actuation systems are sometimes referred to as transfer systems.

Actuation systems are an essential part of real-time hybrid tests because their dynamics severely influence stability and accuracy. Hence, the kinematics and dynamics of the actuation system have to be understood in order to be able to analyze the coupling of the experimental and the virtual component. In Sec. 3.1, two examples of actuation systems and their dynamics are studied. Sec. 3.2 discusses the estimation of interface forces and states for applications where direct measurements are not possible.

3.1 Actuator Dynamics

The design of the actuation system is highly dependent on the application. A wide range of physical effects can be made use of to apply forces, to measure displacements and to measure forces. Actuator types include hydraulic actuators, voice coil actuators, piezoelectric actuators, and others. In this section, two examples of actuator systems are given. The objective is to give an idea on how different actuator types are integrated into the framework which is presented in this thesis, and to describe the models which are used in Sec. 3.2, Chap. 4 and Chap. 5. The methods for this section were analyzed in the student theses by [43] and [105].

3.1.1 Example: Voice Coil Actuators

Voice-coil actuators are linear motors which are frequently used for force-excitation in modal testing. Their working principle is based on the Lorentz forces which act on current-carrying wires in magnetic fields. When used with the purpose of parameter identification or testing, they are often referred to as modal shakers or electro-dynamic shakers. Other applications of shakers in testing include durability tests and life cycle tests. Their properties make voice-coil actuators suitable for the use in vibration testing applications of real-time hybrid testing: Voice-coil actuators have a linear transfer behavior, they

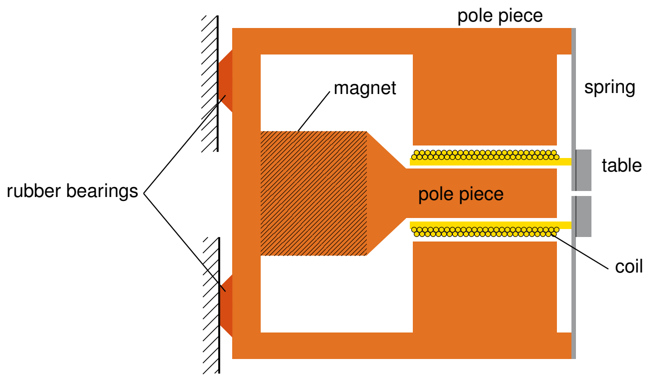


Figure 3.1: Scheme of a voice coil actuator (adapted from [59])

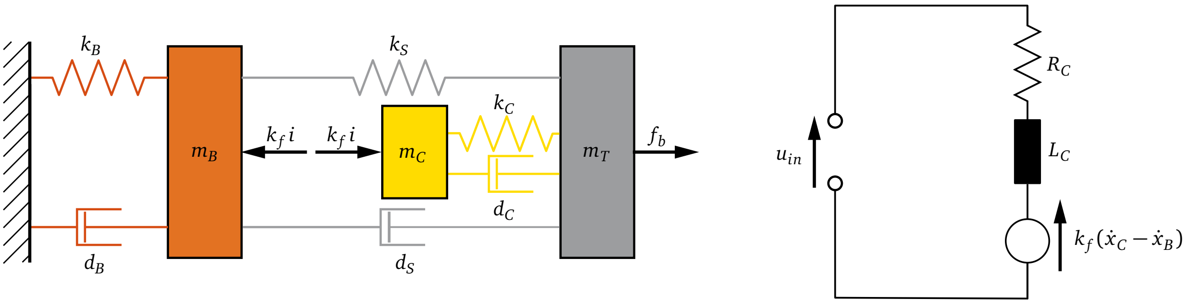


Figure 3.2: Model structure of a voice coil actuator (adapted from [59])

can be used in a reasonably wide frequency band, and their stroke length is high compared to piezoelectric actuators. Because they are made use of for experimental studies in this work, a simple shaker model is described in this chapter.

Voice coil actuators can be used to actuate more than one DOF if they are part of a multi-DOF actuation system. Actuation systems for real-time hybrid testing consist of the actuators and a kinematic chain, which enables to act on all required DOFs. Besides the dynamics of an electro-dynamic shaker—which is described in the first part of the section—a three DOF parallel mechanism for the actuation of a beam is described in the second part of the section.

Electro-Mechanical Model of Voice Coil Actuators A scheme of an electro-dynamic shaker is given in Fig. 3.1: The body of the shaker consists of an outer and an inner pole piece. The core of the design is a magnet. The soft iron body focuses the magnetic field in the gap between the pole pieces. The coil moves in this gap. It is supported by leaf springs which allow linear motion. The load is mounted on the table which, in turn, is mounted on the coil and the flexures. Elastic rubber bearings serve as isolation mounts. Voice-coil actuators are electro-mechanical systems, since the mechanical and the electro-dynamic domain are relevant when modeling them. Such models have been described in a wide range of publications, e.g. [63], [59] and [83]. [63] describes modeling approaches with three degrees of freedom for control applications. [59] discusses the performance of shakers and uses a model with four degrees of freedom. [83] uses shaker models to simulate vibration tests. In the following section, the model structure of [59]—with three mechanical DOF and one electrical DOF—is used and further simplified. A visual representation of this model structure is shown in Fig. 3.2. The ODE, which results from Kirchhoff's voltage law and the circuit diagram, is

$$R_C \cdot i_C + L_C \frac{\mathrm{d}i_C}{\mathrm{d}t} + k_f(\dot{x}_C - \dot{x}_B) = u_{in}, \tag{3.1}$$

where the coil resistance is denoted by R_C, the coil inductance is denoted by L_C, the current is denoted by i and the input voltage is denoted by u_{in}. The force constant $k_f = B \cdot l$ is the product of the magnetic flux density B and the length l of the coil wire in the magnetic field. The mechanical system can be modeled using lumped masses as shown in Fig. 3.2. The dynamics of the lumped mass system are given in Eq. (3.2):

$$\begin{bmatrix} m_C & 0 & 0 \\ 0 & m_T & 0 \\ 0 & 0 & m_B \end{bmatrix} \begin{bmatrix} \ddot{x}_C \\ \ddot{x}_T \\ \ddot{x}_B \end{bmatrix} + \begin{bmatrix} d_C & -d_C & 0 \\ -d_C & d_C + d_S & -d_S \\ 0 & -d_S & d_B \end{bmatrix} \begin{bmatrix} \dot{x}_C \\ \dot{x}_T \\ \dot{x}_B \end{bmatrix} + \\ + \begin{bmatrix} k_C & -k_C & 0 \\ -k_C & k_C + k_S & -k_S \\ 0 & -k_S & k_B \end{bmatrix} \begin{bmatrix} x_C \\ x_T \\ x_B \end{bmatrix} = \begin{bmatrix} k_f i_C \\ 0 \\ -k_f i_C \end{bmatrix} + \begin{bmatrix} 0 \\ f_b \\ 0 \end{bmatrix} \tag{3.2}$$

Here m_* are the lumped masses, d_* the damping constants and k_* the spring constants. The subscripts C, T, and B denote the coil, the table, and the body. x_T is the position of the table, x_C is the position of the coil, and x_B is the position of the body. $k_f \cdot i_C$ is the Lorentz force which is caused by the current in the coil, and the magnetic field. f_b are the boundary forces which act on the load table. Usually, in a real-time hybrid test, a force sensor is mounted on the table in order to measure the forces which are applied on the specimen. Electric circuit and mechanical system are coupled over the electromotive force and the Lorentz force. Note that the electrodynamic equation is obsolete if the system is under current control because the current is assumed to be imposed in this case. The necessary amplifier circuit is described by [83].

The term $k_f(\dot{x}_C - \dot{x}_B)$ is referred to as the electromotive force. It is the voltage which is induced by the moving coil in the magnetic field. The electric circuit introduces damping through the electromotive forces. The effect can be observed in Fig. 3.3, where the transfer function—between current and acceleration—of the current controlled system shows an undamped resonance. In contrast, the transfer function—between voltage and acceleration—of the voltage controlled system exhibits a damped resonance.

As described by [59], the system with three degrees of freedom exhibits three characteristic modes for typical shaker designs: In the first mode, all masses of the shaker move relative to the environment. The elasticity of the rubber mounts enables this mode. The mode can be seen as a small dent in the transfer function. In the second mode, the load table and the coil move against the body of the shaker and in the third mode, the coil moves against the load table. Note that the third mode is outside the operation range since its activation can lead to damages in the coil-table assembly. The frequency of the first mode is below the operational frequency band.

Because the first and third mode are not relevant under operational conditions, the mechanical part of the model can be reduced to a one DOF system by neglecting the elasticity of the rubber mounts and the coil. The resulting system is given in Eq. (3.3):

$$\begin{bmatrix} m_T + m_C & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \ddot{x}_C \\ 0 \end{bmatrix} + \begin{bmatrix} d_S & 0 \\ k_f & L_C \end{bmatrix} \begin{bmatrix} \dot{x}_C \\ \mathrm{d}i_C/\mathrm{d}t \end{bmatrix} + \begin{bmatrix} k_S & -k_f \\ 0 & R_C \end{bmatrix} \begin{bmatrix} x_C \\ i_C \end{bmatrix} = \begin{bmatrix} f_b \\ u_{in} \end{bmatrix} \tag{3.3}$$

If current control is applied, the circuit equation can be omitted, and the remaining equation of motion is

$$(m_T + m_C)\ddot{x}_C + d_S \dot{x}_C + k_S x_C = k_f \cdot i_C + f_b \tag{3.4}$$

The differences between the mechanical model with one DOF and the mechanical model with three DOFs can be observed in Fig. 3.3. It can be seen that the one DOF models are accurate up to the third resonance frequency which includes the entire operational frequency range.

Fig. 3.4 shows an identified one DOF model and the result of a measurement for a modal shaker (type Tira S50018).

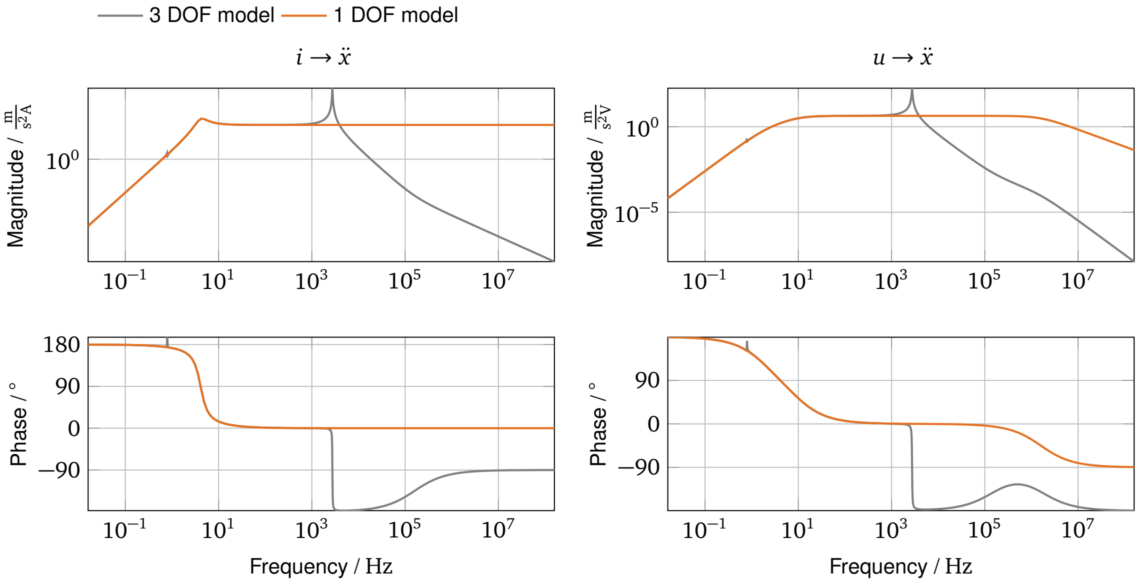


Figure 3.3: Transfer functions of an exemplary electro-dynamic shaker model

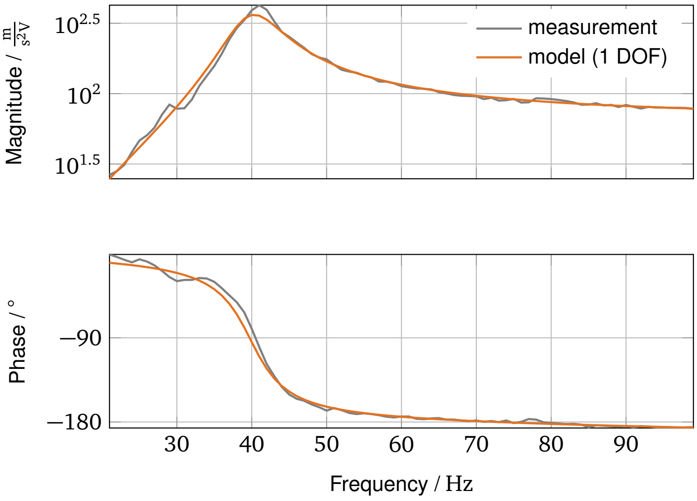


Figure 3.4: Transfer function between voltage u_{in} and acceleration $\ddot{x}$ of a shaker with current control

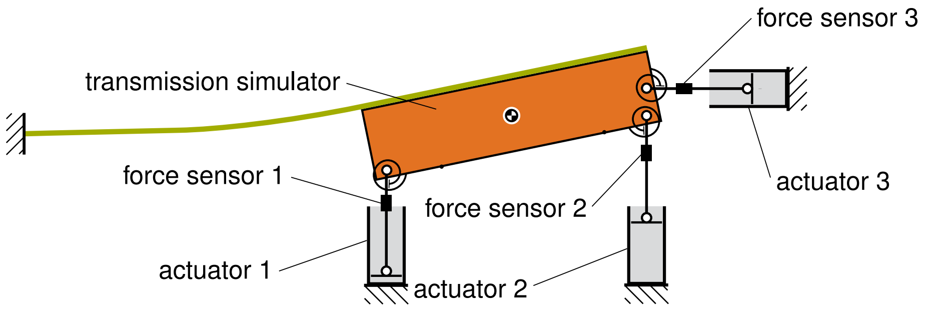


Figure 3.5: Assembly of actuation system and experimental component used in the numerical study.

Actuation System Dynamics In this work, the mechanism which allows the actuation of more than one DOF is referred to as the actuation system. In order to actuate a system with a three DOF interface, a system with a parallel kinematic is applied because the size and weight of the shakers make the use a serial kinematic mechanism unfeasible. A simplified model of an actuation system—which is used as an example in Sec. 3.2—is introduced. The test rig assembly consists of a flexible beam—which serves as the experimental component—and the actuation system. A schematic visualization of the system is shown in Fig. 3.5.

The actuation system is a rigid platform which is controlled with three actuators. Referring to the dynamic substructuring literature [2], this component is called transmission simulator in this thesis. The purpose of the transmission simulator is to replicate the properties of the real interface realistically. One can easily verify that the stiffness of the test specimen is significantly affected by the size of the interface. Additionally, the replication of interface friction effects can be improved by the use of a suitable transmission simulator.

In order to couple the electro-dynamic shakers to the rigid platform, so-called stingers can be used. Stingers are usually assembled by using compliant wires. The objective of their use is to apply pure translational forces on the structure. Since buckling has to be prevented, a minimum diameter of the stingers is necessary. As a consequence, rotational stiffness is imposed on the actuation system. The stingers are modeled as bars with attached rotational spring-damper elements (spring constant k_α, damping constant d_α). Note that in this chapter, the focus is on the modeling of the actuation system and the estimation of the interface states and forces. The objective of the approach presented in Sec. 3.2 is to estimate the interface forces, such that the beam can be coupled in a real-time hybrid test to a virtual component. The virtual component and the coupling techniques are not further discussed in this section. The reader is referred to Chap. 4 and Chap. 5 for a discussion of coupling techniques and experimental results. The linearized dynamics of the actuation system are given by Eq. (3.5).

$$\begin{aligned}
&\underbrace{\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & \theta \end{bmatrix}}_{\boldsymbol{M}^{ACT}} \underbrace{\begin{bmatrix} \ddot{x}_C \\ \ddot{y}_C \\ \ddot{\alpha} \end{bmatrix}}_{\ddot{\boldsymbol{q}}^{ACT}} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & d_\alpha \end{bmatrix}}_{\boldsymbol{D}^{ACT}} \underbrace{\begin{bmatrix} \dot{x}_C \\ \dot{y}_C \\ \dot{\alpha} \end{bmatrix}}_{\dot{\boldsymbol{q}}^{ACT}} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & k_\alpha \end{bmatrix}}_{\boldsymbol{K}^{ACT}} \underbrace{\begin{bmatrix} x_C \\ y_C \\ \alpha \end{bmatrix}}_{\boldsymbol{q}^{ACT}} = \\
&= \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -b & 0 & 1 \end{bmatrix}}_{\boldsymbol{G}^{ACT^T}} \underbrace{\begin{bmatrix} F_{Bx} \\ F_{By} \\ M_B \end{bmatrix}}_{\boldsymbol{\lambda}} + \underbrace{\begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ -\frac{d}{2} & \frac{d}{2} & 0 \end{bmatrix}}_{\boldsymbol{B}_A^{ACT}} \underbrace{\begin{bmatrix} F_{A1} \\ F_{A2} \\ F_{A3} \end{bmatrix}}_{\boldsymbol{f}_A^{ACT}}
\end{aligned} \tag{3.5}$$

where $\boldsymbol{q}^{ACT} = \begin{bmatrix} x_C & y_C & \alpha \end{bmatrix}^T$ are the generalized coordinates of the actuation system, x_C is the horizontal displacement of the center of gravity, y_C is the vertical displacement of the center of gravity and α is the rotation angle of the body. The geometric parameters as well as a free body diagram of interface and reaction forces are given in Fig. 3.6. The linearized position vectors of the force application points ($\boldsymbol{r}_{A1}$, $\boldsymbol{r}_{A2}$ and $\boldsymbol{r}_{A3}$), displacement sensors ($\boldsymbol{r}_{S1}$, $\boldsymbol{r}_{S2}$ and $\boldsymbol{r}_{S3}$), and acceleration sensors ($\boldsymbol{r}_{S4}$ and $\boldsymbol{r}_{S5}$) are given in Eqs. (3.6).

$$\boldsymbol{r}_{A1} = \boldsymbol{r}_{S1} = \begin{bmatrix} x_C - \frac{d}{2} \\ y_C - \frac{d}{2}\alpha \\ 0 \end{bmatrix} \qquad \boldsymbol{r}_{A2} = \ \boldsymbol{r}_{S2} = \begin{bmatrix} x_C + \frac{d}{2} \\ y_C + \frac{d}{2}\alpha \\ 0 \end{bmatrix} \qquad \boldsymbol{r}_{A3} = \ \boldsymbol{r}_{S3} = \begin{bmatrix} x_C + c \\ b + y_C + c\alpha \\ 0 \end{bmatrix}$$
$$\boldsymbol{r}_{S4} = \begin{bmatrix} x_C - a \\ y_C - a\alpha \\ 0 \end{bmatrix} \qquad \boldsymbol{r}_{S5} = \begin{bmatrix} x_C + a \\ y_C + a\alpha \\ 0 \end{bmatrix} \qquad \boldsymbol{r}_{B} = \begin{bmatrix} x_C - b\alpha \\ b + y_C \\ 0 \end{bmatrix} \tag{3.6}$$

The coupling between experimental component and virtual component is performed using the interface forces which are not directly measured by the force sensors. The computation of the actual interface forces by using a system model and the measurements is analyzed in Sec. 3.2.

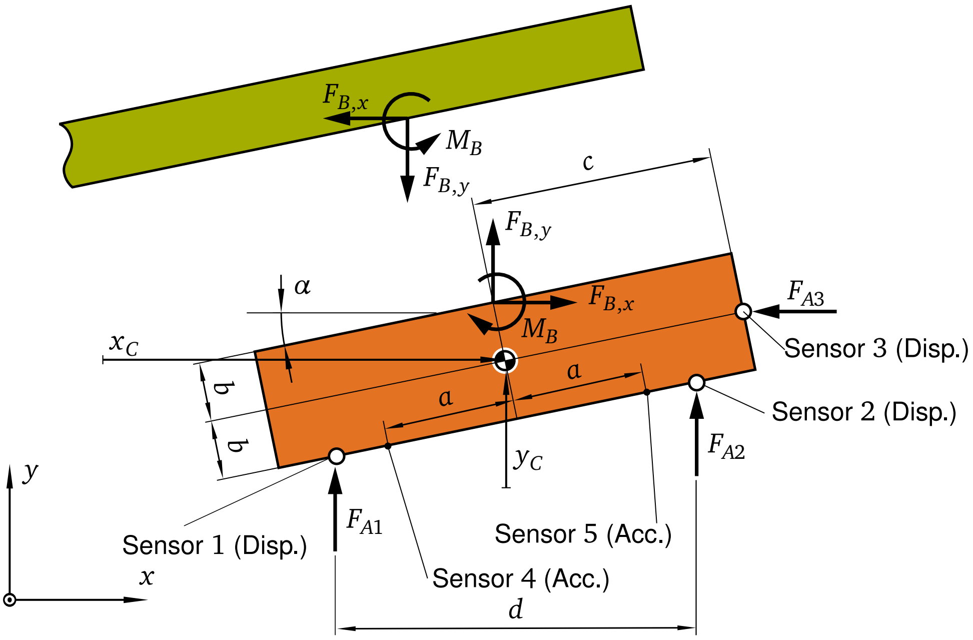


Figure 3.6: Actuation System used in the numerical study

Control In order to couple virtual and experimental substructure, it is necessary for the actuation system to follow demanded position- or force-trajectories. Those trajectories are prescribed time-step by time-step by an interface synchronization control algorithm based on the output from the virtual component. The interface synchronization algorithms are discussed in Chap. 4 and Chap. 5. The control loop which directly controls the actuators or the actuation system is usually referred to as inner loop control in literature. Mostly position control is made use of in the inner loop because this controller type is available commercially for many actuators. Fewer interface synchronization strategies work by prescribing the interface forces.

Linear-Quadratic-Regulators combine state feedback control with state observers (see e.g. the textbook [34]). They are especially suitable to control multiple-input/multiple-output systems. Fig. 3.7 shows

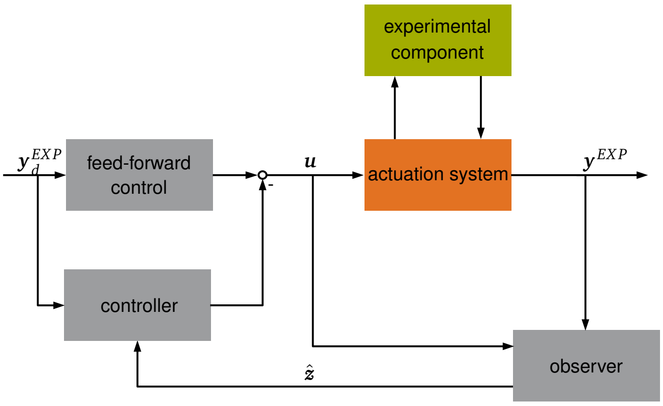


Figure 3.7: Block diagram of a Linear-Quadratic-Regulator applied to real-time hybrid testing

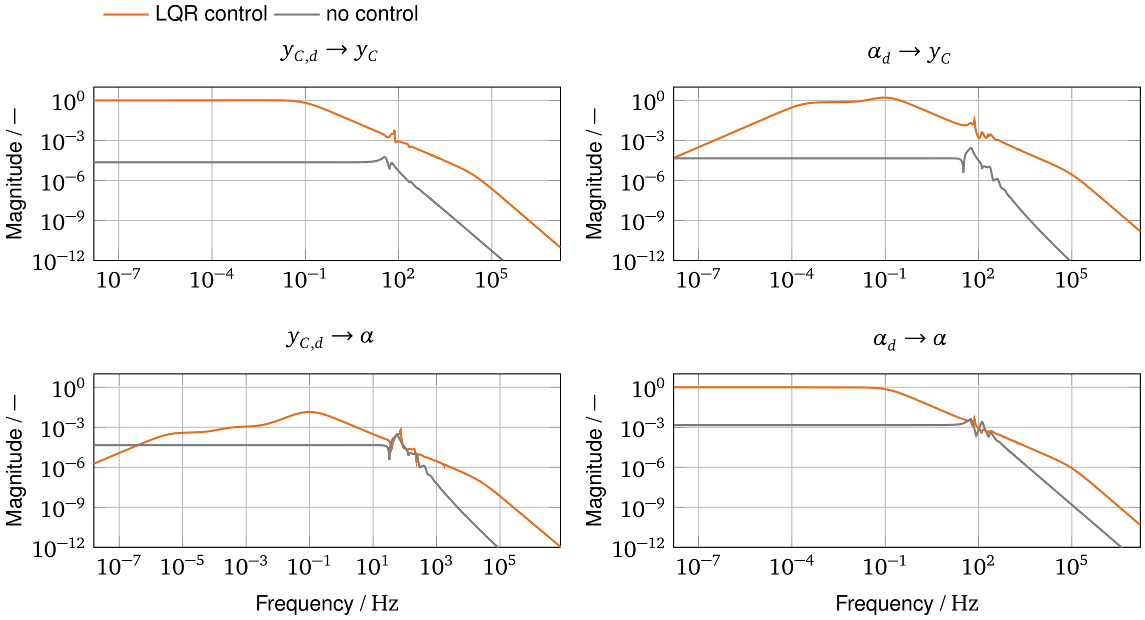


Figure 3.8: Example of LQR control of an actuation system for the position demands $y_{C,d}$ and α_d

the structure of a Linear-Quadratic-Regulator with a state observer. The demand signal $\boldsymbol{y}_d^{EXP}$ serves as an input to a feedforward controller. The states $\boldsymbol{z}$ of the system are estimated using a state observer (estimated states $\hat{\boldsymbol{z}}$). The state observer can be designed as a Kalman filter. Fig. 3.8 shows an exemplary simulation of a transfer behavior of the controlled actuation system using the system parameters from Appendix B and the actuation system design is shown in Fig. 3.6. The controller lets the system follow the prescribed position demand in the low-frequency range. Moreover, both input/output relations are decoupled in the low-frequency range, which means that an input channel only acts on the corresponding output.

3.1.2 Example: Stewart Platforms

The following example is intended to show the integration of a parallel robot into the real-time hybrid testing framework. The use of parallel robots may be necessary in order to simulate large amplitudes and multiple DOF interfaces. Hexapods or Stewart platforms are parallel robots which can position a rigid platform in all six DOF. The principle of a parallel robot with six DOF has been introduced by [38]

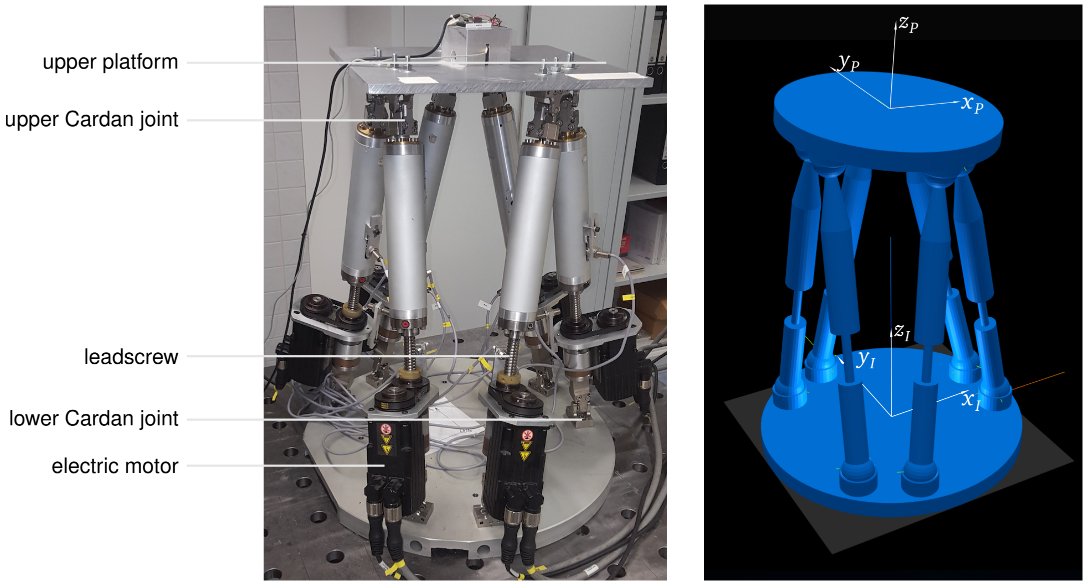


Figure 3.9: A Stewart platform: Test rig and visualization of simulation model with coordinate systems (modified from [105])

for the application of tire testing. Stewart platforms are named after the author of the publication [100], where a similar mechanism is presented for the application in flight simulators.

Compared to conventional serial robots, Stewart platforms have the advantage of an increased stiffness [66]. Since the actuators are not arranged in a serial kinematic chain but in a parallel structure, they undergo smaller accelerations, and the inertia effects of moving actuators are reduced. Those properties make them a suitable choice for real-time hybrid testing. An application of Stewart platforms in real-time hybrid testing of prosthetic feet has been presented in [48].

An example of a Stewart platform is shown in Fig. 3.9. The six legs are arranged in a parallel topology. The legs are mounted on the base platform and on the upper platform using Cardan joints. The object, which is manipulated by the robot, is mounted at the so-called tool center point. More specifically, in the case of real-time hybrid testing, a force sensor and the experimental component are mounted at the tool center point. The length of each of the six legs can be controlled via leadscrews and servo motors. Instead of servo motors and leadscrews, one can make use of other types of linear actuators such as hydraulic cylinders.

The basic structure of an actuator based control scheme which can be applied in real-time hybrid testing is shown in Fig. 3.10. Due to friction effects in the transmission system and in the joints, force control of a Stewart platform on the actuator level is usually not feasible. Using position control, the desired workspace position and orientation $\boldsymbol{y}_d^{EXP}$—which corresponds to the interface in real-time hybrid testing—serves as an input to the control system of the Stewart platform. The corresponding actuator positions—which are the leg lengths—are computed using the inverse kinematics of the system. A conventional control law on the single actuator level helps to follow the demanded actuator position. The platform is subject to inertia effects and friction in leadscrews and Cardan joints. More sophisticated control laws consider those effects. Note that the Stewart platform is coupled with the experimental component which can influence the dynamics of the system. However, due to the high transmission ratio the influence of inertia and friction on the transfer behavior can be neglected in many cases. Note that friction has indeed an influence on the actuator forces which is discussed below.

The direct kinematics of a robot describe the relation between a given actuator position and the workspace coordinates. For serial robots, this can be computed directly from a closed-form equation. For parallel robots, the direct kinematics have to be solved numerically. The inverse kinematics, in turn, describe the actuator positions based on a given workspace position and orientation. For the Stewart platform,

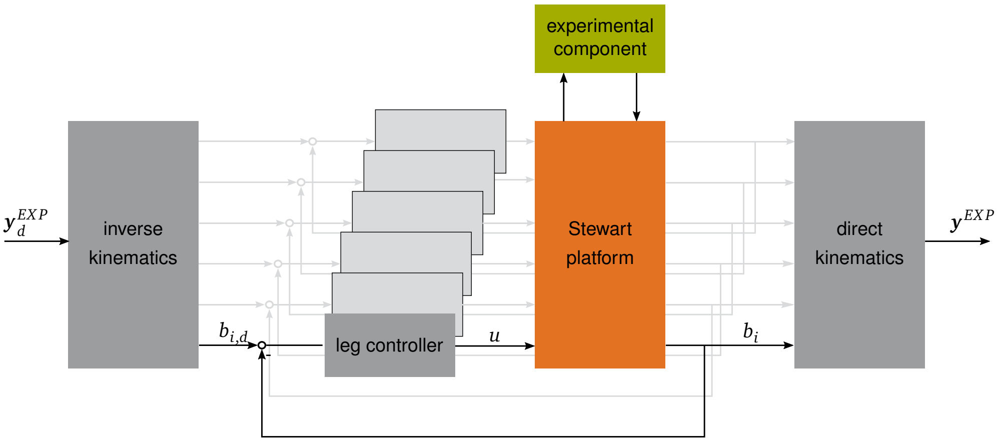


Figure 3.10: Control structure of a Stewart platform in real-time hybrid testing.

a closed-form solution exists. In real-time hybrid testing, workspace position and orientation are given as a demand signal, since they correspond to the interface coordinates $\boldsymbol{y}^{ACT}$. There are several ways to describe the spatial orientation of an object. The orientation of the interface can—for example—be described using Cardan angles. The vector of the generalized coordinates of the system reads

$$\boldsymbol{q} = \boldsymbol{y}^{ACT} = \begin{bmatrix} x & y & z & \alpha & \beta & \gamma \end{bmatrix}^T, \tag{3.7}$$

In order to compute the actuator positions, a vector chain—as it is described in [85]—is applied. Fig. 3.11 shows the topology of the vector chain which includes the position of the tool center point and the actuated leg. The vector chain can be expressed with the equation

$${}_I\boldsymbol{b}_i = {}_I\boldsymbol{t} + {}_I\boldsymbol{Q}_T \cdot {}_T\boldsymbol{o}_i - {}_I\boldsymbol{a}_i. \tag{3.8}$$

where ${}_I\boldsymbol{b}_i$ is the vector along leg $i \in (1 \dots 6)$ and between the two joints, ${}_T\boldsymbol{o}_i$ is the vector from the tool center point to the upper joint of leg i, ${}_I\boldsymbol{a}_i$ is the vector from origin of the inertial coordinate system to the lower joint of leg i, and the vector ${}_I\boldsymbol{t}$ represents the position of the tool center point with respect to the origin of the inertial coordinate system. All vectors are represented either in the inertial coordinate system I or in the coordinate system P which is fixed to the upper platform. The rotation matrix ${}_I\boldsymbol{Q}_P$ describes the transformation of vector representations from the P-coordinate-system into the I-coordinate-system. The rotation matrix ${}_I\boldsymbol{Q}_P(\alpha, \beta, \gamma) = \boldsymbol{Q}_z(\gamma) \cdot \boldsymbol{Q}_y(\beta) \cdot \boldsymbol{Q}_x(\alpha)$—which uses Cardan angles—is composed of a chain of three elementary rotations. Namely, the elemental rotations are $\boldsymbol{Q}_x(\alpha)$ with the angle α around the x-axis, $\boldsymbol{Q}_y(\beta)$ with the angle β around the y-axis and $\boldsymbol{Q}_z(\gamma)$ with the angle γ around the z-axis. Equivalently, the rotation matrix ${}_I\boldsymbol{Q}_P$ can be constructed e.g. from Euler angles or quaternions. The leg lengths—which serve as actuator inputs—are computed from the leg vector ${}_I\boldsymbol{b}_i$:

$$b_i = \sqrt{{}_I\boldsymbol{b}_i^T \cdot {}_I\boldsymbol{b}_i}. \tag{3.9}$$

The dynamics of the platform can be described in the form of Eq. (2.3). The dynamics of the specific Stewart platform, which is shown in Fig. 3.9, have been discussed in [85], [84] and [105]. Fig. 3.12 shows the exemplary responses to steps in different workspace coordinates which are retrieved from a simulation. Due to the actuator dynamics and the velocity constraints, the responses exhibit significant time lags which have to be considered in the design of a real-time hybrid test. The velocity constraints are implemented for safety reasons in order to keep the actuator workspace restrictions. Fig. 3.13 shows the actuator forces resulting from a sinusoidal input signal. Friction effects cause the jumps in the actuator forces. The friction parameters are subject to changes due to temperature conditions and wear [85]. Since those effects are hard to predict, it makes sense to measure the interface forces directly at the interface rather than computing them from a model of the actuation system.

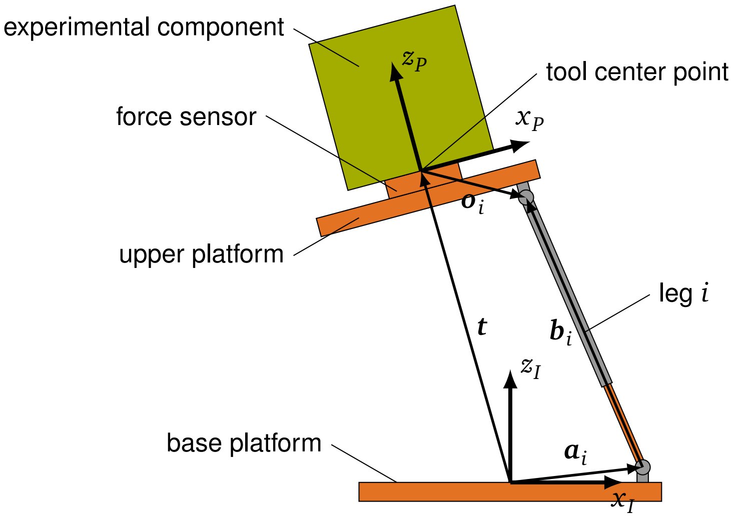


Figure 3.11: Vector chain used for computing the leg lengths of a Stewart platform for a given workspace configuration

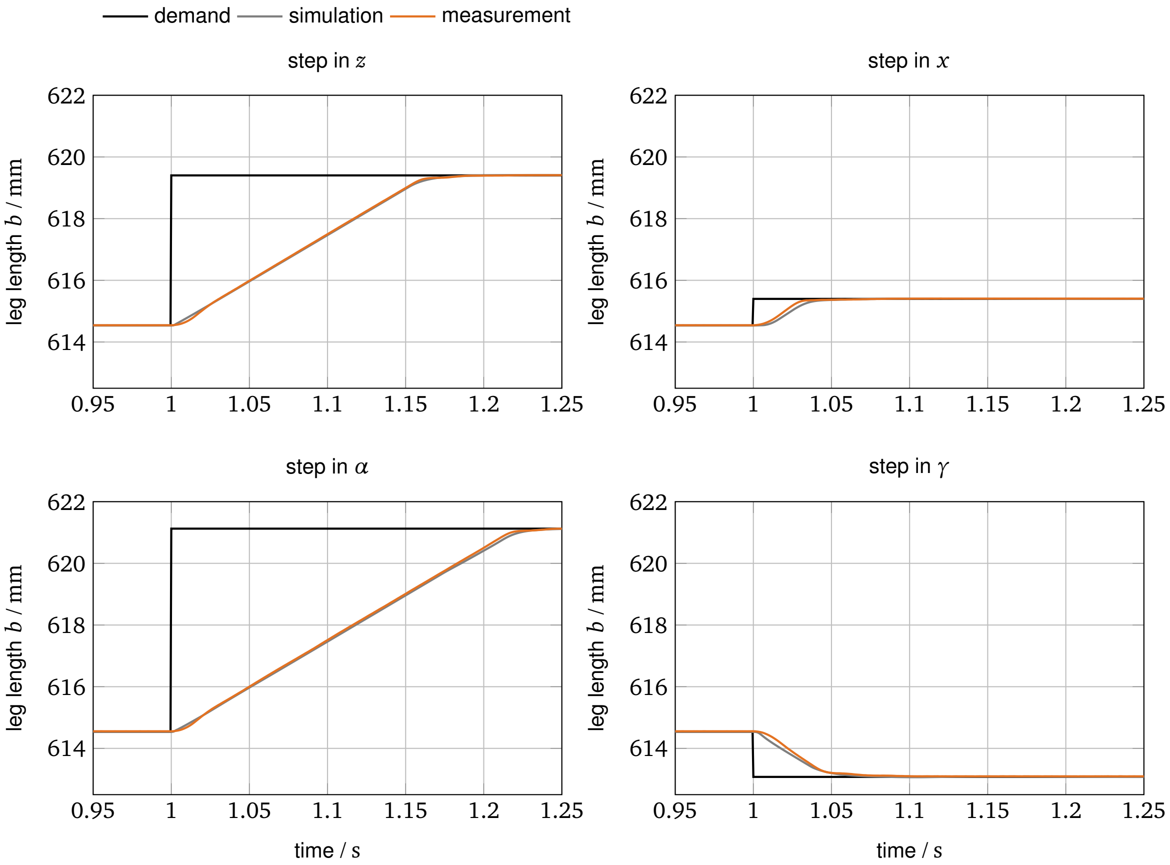


Figure 3.12: Example of responses of the leg length b_5 to steps in workspace coordinate demands.

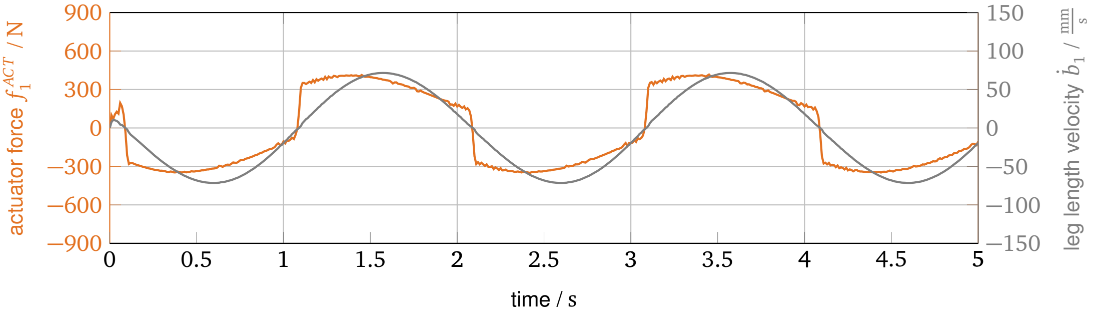


Figure 3.13: Example of an actuator force at one leg due to a sinusoidal workspace coordinate demand signal (modified from [105])

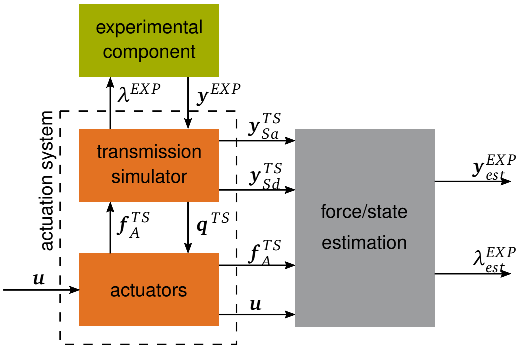


Figure 3.14: Block diagram of an actuation system with transmission simulator and force estimation

3.2 Estimation of Interface Force and States

In many applications, forces, moments and displacements or rotations cannot be measured directly at the interface. The reason can be the use of a transmission simulator. Here, the force sensors are not directly mounted at the interface but rather between actuators and transmission simulators. This type of assembly is shown in Fig. 3.5. 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 of forces and moments at the interface. One cannot avoid the fact that 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—such as a stinger—can bring in forces or moments that cannot be measured by one-DOF force sensors. The bending of the stinger causes unmeasurable moments.

The following section describes procedures to compensate for the dynamic effects of the actuation system and to calculate interface quantities which are necessary for the coupling. Fig. 3.14 shows the block diagram of an actuation system with transmission simulator and force estimation. The Eqs. (3.10) represent the corresponding dynamics of the experimental component and the actuation system.

$$\begin{aligned} \boldsymbol{M}^{EXP}\ddot{\boldsymbol{q}}^{EXP} + \boldsymbol{D}^{EXP}\dot{\boldsymbol{q}}^{EXP} + \boldsymbol{K}^{EXP}\boldsymbol{q}^{EXP} &= \boldsymbol{f}_{ext}^{EXP} + \boldsymbol{G}^{EXP^T}\boldsymbol{\lambda}^{EXP} \\ \boldsymbol{M}^{TS}\ddot{\boldsymbol{q}}^{TS} + \boldsymbol{D}^{TS}\dot{\boldsymbol{q}}^{TS} + \boldsymbol{K}^{TS}\boldsymbol{q}^{TS} &= \boldsymbol{B}_A^{TS}\boldsymbol{f}_A^{TS} + \boldsymbol{G}^{TS^T}\boldsymbol{\lambda}^{EXP} \end{aligned} \tag{3.10}$$

Here, the actuator forces $\boldsymbol{f}_A^{TS}$ are assumed to be measured during the test. The matrix $\boldsymbol{G}^{TS}$ relates the state vector of the transmission simulator to the interface displacements.

3.2.1 Interface States: Projected Displacements and Accelerations

The compatibility between the two subsystems is given by Eq. (2.6). In order to obtain the contribution $\boldsymbol{y}^{EXP}$ of the experimental component, it is necessary to measure the interface displacements directly or calculate them by time integration of acceleration signals. $\boldsymbol{y}^{EXP}$ is related to the state vector of the transmission simulator $\boldsymbol{q}^{TS}$ by the equation

$$\boldsymbol{y}^{EXP} = \boldsymbol{G}^{TS}\boldsymbol{q}^{TS} \tag{3.11}$$

using the output matrix $\boldsymbol{G}^{TS}$. The second-order time derivatives $\ddot{\boldsymbol{q}}^{TS}$ of the transmission simulator states are required for the compensation of the transmission simulator's inertia effects. They can be calculated from acceleration measurements. Eqs. (3.12) define the relation between sensor outputs and states:

$$\boldsymbol{y}_{Sd}^{TS} = \begin{bmatrix} y_{Sd,1}^{TS} \\ y_{Sd,2}^{TS} \\ \cdots \end{bmatrix} = \boldsymbol{C}_{Sd}\boldsymbol{q}^{TS} \qquad \boldsymbol{y}_{Sa}^{TS} = \begin{bmatrix} y_{Sa,1}^{TS} \\ y_{Sa,2}^{TS} \\ \cdots \end{bmatrix} = \boldsymbol{C}_{Sa}\ddot{\boldsymbol{q}}^{TS} \tag{3.12}$$

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

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

The required contribution to the compatibility equation, which is used for coupling virtual and experimental component, can be calculated using Eqs. (3.13):

$$\begin{aligned} \boldsymbol{G}^{EXP}\boldsymbol{q}^{EXP} &= \boldsymbol{G}^{TS}\boldsymbol{q}^{TS} = \boldsymbol{G}^{TS}\boldsymbol{C}_{Sd}^{+}\boldsymbol{y}_{Sd}^{TS} \\ \boldsymbol{G}^{EXP}\ddot{\boldsymbol{q}}^{EXP} &= \boldsymbol{G}^{TS}\ddot{\boldsymbol{q}}^{TS} = \boldsymbol{G}^{TS}\boldsymbol{C}_{Sa}^{+}\boldsymbol{y}_{Sa}^{TS} \end{aligned} \tag{3.13}$$

For rigid body interfaces, this approach is equivalent to the virtual point method described in [93].

3.2.2 Interface Forces: Projected Forces and Inverse Dynamics

Once the actuation system's state vectors $\boldsymbol{q}^{TS}$, $\dot{\boldsymbol{q}}^{TS}$ and $\ddot{\boldsymbol{q}}^{TS}$ are identified, they can be used to calculate an estimate of the interface forces $\boldsymbol{\lambda}_{est}^{EXP}$ from the model. To do so, one can solve the actuator Eqs. (3.10) for $\boldsymbol{\lambda}_{est}^{EXP}$:

$$\boldsymbol{\lambda}_{est}^{EXP} = \boldsymbol{G}^{TS^T,+}\left(\boldsymbol{M}^{TS}\ddot{\boldsymbol{q}}^{TS} + \boldsymbol{D}^{TS}\dot{\boldsymbol{q}}^{TS} + \boldsymbol{K}^{TS}\boldsymbol{q}^{TS} - \boldsymbol{B}_{A}^{TS}\boldsymbol{f}_{A}^{TS}\right) \tag{3.14}$$

Here $\boldsymbol{G}^{TS^T,+}$ is the Moore–Penrose inverse of $\boldsymbol{G}^{TS^T}$. This approach is referred to as inverse dynamics in the following numerical study. Neglecting all dynamics yields the pure kinematics of Eq. (3.15).

$$\boldsymbol{\lambda}_{est}^{EXP} = -\boldsymbol{G}^{TS^T,+}\boldsymbol{B}_{A}^{TS}\boldsymbol{f}_{A}^{TS} \tag{3.15}$$

This approach is referred to as projected forces approach in the following numerical study. The approach is equivalent to the calculation of the interface forces with the virtual point method in [93].

3.2.3 Interface Forces and States: Augmented-State Kalman Filter

The Kalman Filter—proposed by [54]—is an approach taking into account measurement and model uncertainties. It is based on a probabilistic prediction and update scheme. [22] proposes to apply Kalman filters to real-time hybrid testing for removing sensor signal noise. Note, that the filter is used in this case purely to smoothen the sensor signals. In contrast, in this section, the use of the Kalman filter is linked to the estimation of unmeasured interface forces and states. The predictions in a Kalman filter use a state space model as a basis, and the updates originate in measurements. Adding the interface force $\boldsymbol{\lambda}^{EXP}$ as an additional state to a state space model of the transmission simulator yields the augmented-state model

$$\underbrace{\begin{bmatrix} \dot{\boldsymbol{q}}^{TS} \\ \ddot{\boldsymbol{q}}^{TS} \\ \dot{\boldsymbol{\lambda}} \end{bmatrix}}_{\dot{\boldsymbol{x}}} = \underbrace{\begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ -\boldsymbol{M}^{TS^{-1}}\boldsymbol{K}^{TS} & -\boldsymbol{M}^{TS^{-1}}\boldsymbol{D}^{TS} & \boldsymbol{M}^{TS^{-1}}\boldsymbol{G}^{TS^{T}} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{A}} \underbrace{\begin{bmatrix} \boldsymbol{q}^{TS} \\ \dot{\boldsymbol{q}}^{TS} \\ \boldsymbol{\lambda}^{EXP} \end{bmatrix}}_{\boldsymbol{x}} + \underbrace{\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{B}_A^{TS} \\ \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{B}} \boldsymbol{f}_A^{TS} \tag{3.16}$$

$$\begin{aligned} \begin{bmatrix} \boldsymbol{y}_d^{TS} \\ \boldsymbol{y}_a^{TS} \\ \dots \end{bmatrix} &= \underbrace{\begin{bmatrix} \boldsymbol{C}_d & \boldsymbol{0} & \boldsymbol{0} \\ -\boldsymbol{C}_a\boldsymbol{M}^{TS^{-1}}\boldsymbol{K}^{TS} & -\boldsymbol{C}_a\boldsymbol{M}^{TS^{-1}}\boldsymbol{D}^{TS} & \boldsymbol{C}_a\boldsymbol{M}^{TS^{-1}}\boldsymbol{G}^{TS^{T}} \end{bmatrix}}_{\boldsymbol{C}} \begin{bmatrix} \boldsymbol{q}^{TS} \\ \dot{\boldsymbol{q}}^{TS} \\ \boldsymbol{\lambda}^{EXP} \end{bmatrix} \\ &\quad + \underbrace{\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{C}_a\boldsymbol{B}_A^{TS} \\ \dots \end{bmatrix}}_{\boldsymbol{D}} \boldsymbol{f}_A^{TS}. \end{aligned} \tag{3.17}$$

The model based prediction step of the Kalman filter at time step k is defined by the well known Kalman-filter Eqs. (3.18)

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= \mathbf{A}_k\hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k\mathbf{F}_A \\ \mathbf{P}_{k|k-1} &= \mathbf{A}_k\mathbf{P}_{k-1|k-1}\mathbf{A}_k^{\mathsf{T}} + \mathbf{Q}_k, \end{aligned} \tag{3.18}$$

where $\mathbf{A}_k$ is the time discretization of the system matrix $\boldsymbol{A}$, $\mathbf{B}_k$ is the time discretization of the input matrix $\boldsymbol{B}$, $\hat{\mathbf{x}}_{k|k-1}$ is the predicted state vector, $\mathbf{P}_{k|k-1}$ the estimated error covariance matrix and $\mathbf{Q}_k$ is the process noise covariance matrix. The update step is defined by Eqs. (3.19)

$$\begin{aligned} \mathbf{K}_k &= \mathbf{P}_{k|k-1}\mathbf{C}_k^T(\mathbf{C}_k\mathbf{P}_{k|k-1}\mathbf{C}_k^T + \mathbf{R}_k)^{-1} \\ \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{C}_k\hat{\mathbf{x}}_{k|k-1}) \\ \mathbf{P}_{k|k} &= (I - \mathbf{K}_k\mathbf{C}_k)\mathbf{P}_{k|k-1}, \end{aligned} \tag{3.19}$$

where $\mathbf{z}_k$ is the measurements vector, $\mathbf{R}_k$ is the measurement noise covariance matrix and $\mathbf{K}_k$ is the Kalman gain.

3.2.4 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. (3.20).

$$\boldsymbol{M}^{TS}\ddot{\boldsymbol{q}}^{TS} + \boldsymbol{D}^{TS}\dot{\boldsymbol{q}}^{TS} + \boldsymbol{K}^{TS}\boldsymbol{q}^{TS} = \boldsymbol{B}_A^{TS}\boldsymbol{f}_A^{TS} + \boldsymbol{G}^{TS^{T}}\boldsymbol{\lambda} \tag{3.20}$$

The reduced system is given by Eq. (3.21), where $\bar{\boldsymbol{q}}$ is the reduced coordinate vector and the matrices of the reduced system are indicated by bars:

$$\bar{\boldsymbol{M}}^{TS}\ddot{\bar{\boldsymbol{q}}}^{TS} + \bar{\boldsymbol{D}}^{TS}\dot{\bar{\boldsymbol{q}}}^{TS} + \bar{\boldsymbol{K}}^{TS}\bar{\boldsymbol{q}}^{TS} = \bar{\boldsymbol{B}}_A^{TS}\boldsymbol{f}_A^{TS} + \bar{\boldsymbol{G}}^{TS^T}\boldsymbol{\lambda} \tag{3.21}$$

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

$$\bar{\boldsymbol{M}}^{TS} = \boldsymbol{T}^T\boldsymbol{M}^{TS}\boldsymbol{T}, \quad \bar{\boldsymbol{D}}^{TS} = \boldsymbol{T}^T\boldsymbol{D}^{TS}\boldsymbol{T}, \quad \bar{\boldsymbol{K}}^{TS} = \boldsymbol{T}^T\boldsymbol{K}^{TS}\boldsymbol{T},$$

$$\bar{\boldsymbol{B}}_f^{TS} = \boldsymbol{T}^T\boldsymbol{B}_f^{TS} \quad \text{and} \quad \bar{\boldsymbol{G}}^{TS} = \boldsymbol{T}^T\begin{bmatrix}\boldsymbol{I}\\ \boldsymbol{0}\end{bmatrix}$$

The reduction matrix $\boldsymbol{T}$ is retrieved e.g. according to Eq. (3.22) using the Craig-Bampton method with the constraint modes matrix $\boldsymbol{\Psi}^{TS}$ and the truncated fixed interface modes matrix $\boldsymbol{\Phi}^{TS}$. 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 $\boldsymbol{q}_b^{TS}$ 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 [26] and [75] for methods of choosing an interface reduction basis.

$$\boldsymbol{q}^{TS} = \begin{bmatrix}\boldsymbol{q}_b^{TS}\\ \boldsymbol{q}_i^{TS}\end{bmatrix} = \underbrace{\begin{bmatrix}\boldsymbol{\Phi}_b & \boldsymbol{0}\\ \boldsymbol{\Psi}^{TS} & \boldsymbol{\Phi}^{TS}\end{bmatrix}}_{\boldsymbol{T}}\underbrace{\begin{bmatrix}\bar{\boldsymbol{q}}_b^{TS}\\ \bar{\boldsymbol{q}}_i^{TS}\end{bmatrix}}_{\bar{\boldsymbol{q}}^{TS}} \tag{3.22}$$

Numerical Experiment The objective of the case study is to show the effects of measurement noise on the estimation results. The different techniques were applied to the system consisting of a rigid actuation system and a flexible beam, which is described in Sec. 3.1. The system parameters are listed in Appendix B. In a first step, a simulation of the overall system, including beam and actuation system, is carried out. The sensor signals are used as an input for the estimation techniques. The resulting estimates are then compared to the reference signal obtained from the simulation. The following approaches for force estimation described above are investigated:

- projected forces
- inverse dynamics
- augmented-state Kalman filter

In order to analyze the basic effects of sensor noise on the estimation techniques, a sinusoidal excitation $F_{A1} = A\sin(2\pi \cdot f \cdot t)$ on actuator 1, with a constant excitation frequency $f = 50$ Hz and an amplitude of $A = 50$ N, was applied to the system. The noise term $p_{noise,*}$ was used to imitate experimental sensor signals.

$$p_{noise,*} = c_{n*}r_{noise}$$

Here r_{noise} is a uniformly distributed random number between -1 and 1 and c_{n*} is a coefficient specific to the signal. The noise was added to the force ($c_{nf} = 0.5\ \mathrm{N}$), displacement ($c_{nd} = 8 \cdot 10^{-5}\ \mathrm{m}$), and acceleration signals ($c_{na} = 0.1\ \frac{\mathrm{m}}{\mathrm{s}^2}$). Fig. 3.15 shows the estimates of the vertical interface forces. Both augmented-state Kalman filter and inverse dynamics follow the signal accurately. The high noise level in this study affects the signal of the inverse dynamics so that it can be directly seen in the graph. In contrast, the augmented-state Kalman filter smoothens the signals and ensures a more accurate estimation. The projected forces cannot estimate the interface forces correctly as they do not include information on the inertia of the transmission simulator. This fact can also be observed in Fig. 3.16, which shows the estimation of the interface force $F_{B,y}$ for an excitation with actuator 1 in the frequency domain. Three approaches for interface state estimation described above are investigated:

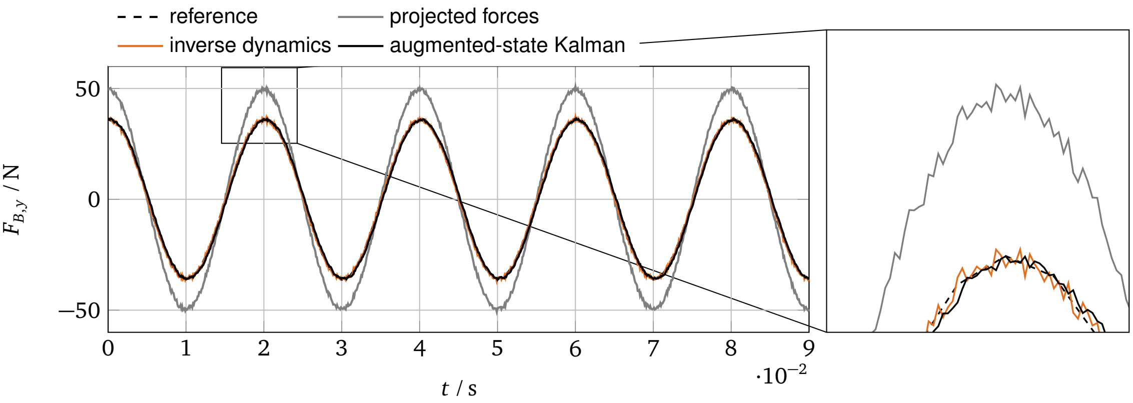


Figure 3.15: Estimates of the vertical interface forces $F_{B,y}$

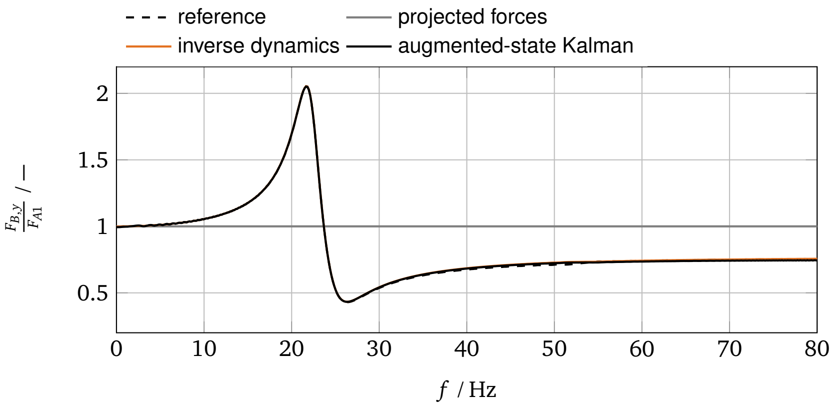


Figure 3.16: Response of estimations of the interface force $F_{B,y}$ to excitation with actuator 1

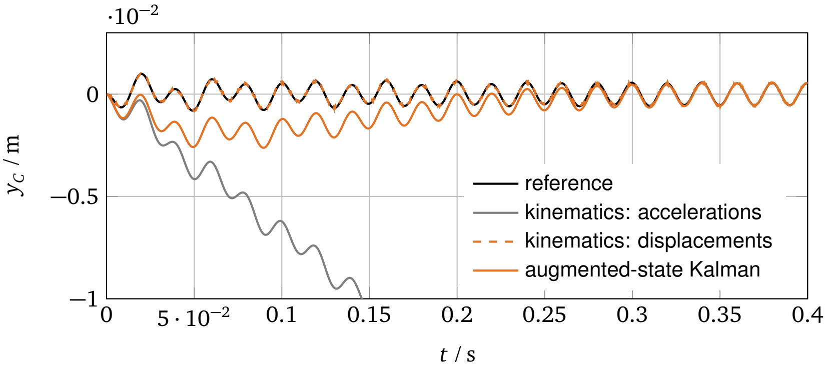


Figure 3.17: Estimates of the vertical interface displacements $r_{I,y}$

- projected displacements
- projected and integrated accelerations
- augmented-state Kalman filter

For the integration of acceleration signals, the initial conditions $\dot{\boldsymbol{y}}_{Sa}^{TS} = \mathbf{0}$ and $\ddot{\boldsymbol{y}}_{Sa}^{TS} = \mathbf{0}$ are used. Fig. 3.17 shows the estimates of the vertical interface displacements. The system is excited by a sinusoidal force $F_{A1} = A\sin(2\pi \cdot f \cdot t)$ on actuator 1 with a constant excitation frequency $f = 50$ Hz and an amplitude of $A = 50$ N. Noise was added to the force ($c_{nf} = 0.5$), displacement ($c_{nd} = 8 \cdot 10^{-5}$), and acceleration signals ($c_{na} = 0.1$). The noise in the acceleration signals and the unknown initial conditions of the integration cause a drift of the estimations. The projected displacements approach follows the reference signal, but the noise is fully passed to the estimated signal. The Kalman filter approach initially causes deviations from the reference, which are decreasing over the simulation period. The filter smoothens the signal without a significant phase lag. It is noteworthy that the quality of the results obtained with the augmented-state Kalman filter for both force and displacement estimation is dependent on the choice of the measurement covariance matrix and of the process covariance matrix. In comparison to other approaches, this fact can cause an increased implementation effort for an estimation based on an augmented-state Kalman filter.

In summary, the following observations can be made from numerical investigations:

- the augmented-state Kalman filter can smoothen the results without a significant phase lag
- model-based approaches (as inverse dynamics or Kalman filtering) are necessary as soon as the transmission simulator's inertia effects are not negligible
- the Kalman filter approach can be used as a consistent way of sensor data fusion

Chapter 4

Feedback Based Coupling

Parts of this chapter have been published in [6] and submitted for publication in [7].

In Chap. 3, it was shown that the actuation system exhibits phase lags, which are also introduced into any real-time hybrid test. In the same way, communication processes and the computation of the response of virtual components can introduce delays into the real-time hybrid test. The described delays and time lags cannot be avoided in most cases. As a consequence, the stability and accuracy of the tests are deteriorated.

In literature, the stability problems are addressed by different interface-synchronization methods. The term interface synchronization refers to the equilibrium of interface forces and to the compatibility of interface displacements which are necessary to mechanically couple two subcomponents. [45] proposes an algorithm for compensating constant actuator delay, which uses polynomial forward prediction. [29] and [109] 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. [108] 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. [79] introduces a framework for the linear analysis of real-time hybrid testing systems, referred to as Model-In-The-Loop testing. [102] 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. [101], [106] and [61] present work within this framework. [101] propose a Model-Predictive Control strategy for the synchronization of the virtual and the experimental component which makes use of models of the experimental component and the virtual component. The approach allows actuator saturation and limits to be accounted for. Model-Predictive Control is also applied by [80]. [61] makes use of neural networks as an adaptive feedforward controller. To do so, the control problem is formulated as disturbance rejection, which extends the existing framework. [106] uses linear state space control to synchronize the interface. [73] addresses the problem of testing a structure which consists 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.

This chapter gives an overview of coupling a virtual and an experimental component in the presence of actuator dynamics. In Sec. 4.1, general control structures are described. Instability issues and their causes are addressed in Sec. 4.2. Two of the methods mentioned above for synchronizing the interfaces—while keeping the test stable and accurate—are characterized in Sec. 4.3 and Sec. 4.4. The methods for this section were analyzed in the student theses by [111], [56], and [82].

4.1 Coupling Formulation

The dynamic equations and constraints which define the coupled overall system are given in Sec. 2.1. It is the objective of any real-time hybrid test to follow the dynamics of a coupled reference system. If no actuator dynamics were present, a real-time hybrid test would have the structure shown in the block diagram of Fig. 4.1 and it would exactly follow the emulated reference dynamics. The interface forces and the interface displacements are exchanged between the two subcomponents. Both subcomponents can be subject to external excitation forces $\boldsymbol{f}_{ext}^{VIR}$ and $\boldsymbol{f}_{ext}^{EXP}$. In order to couple two components of a mechanical system, two constraints have to be met: Coupling between virtual and experimental component is achieved if the interface displacements of virtual and experimental component are compatible and if the interface forces are in equilibrium. Both constraints are formulated in the compatibility constraint of Eq. (2.7) and the equilibrium constraint of Eq. (2.9). The coupled dynamics of the reference system are stable by definition. As mentioned above, a test performed with perfect actuators would yield precisely the reference dynamics of Sec. 2.1 and, hence, would be stable. When analyzing the structure of real-time hybrid testing control schemes, the use of displacement-controlled actuators and force-controlled actuators have to be distinguished. Similar to the control schemes described in the paragraphs below, frameworks for real-time hybrid testing have been proposed by [79] or [102].

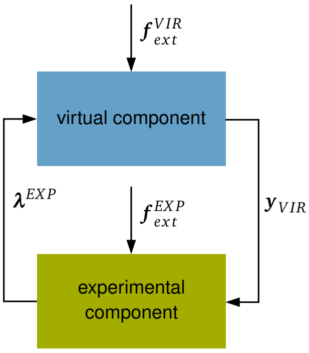


Figure 4.1: Block diagram of the coupled system

Displacement-controlled actuators As mentioned in Chap. 3, commercial displacement controllers are available for many actuators. In many cases, the application of force-controlled actuators is not feasible due to friction effects. Fig. 4.2 shows a block diagram of a real-time hybrid test: The output $\boldsymbol{y}^{VIR}$ of the virtual component—which is the interface displacement of the virtual component—serves as demand signal for the feedback controlled actuation system. The input $\boldsymbol{u}$ of the actuation system —which can include an inner loop control—is generated by an arbitrary interface synchronization control law. The control objective is that the interface displacements of the experimental component $\boldsymbol{y}^{EXP}$ follows the virtual component as exactly as possible in a wide frequency range. If the feedback controller is designed with this control objective, the dynamics of the virtual component and the feedback loop—resulting from the interconnection with the virtual component—are not considered. 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:

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

The framework also includes the possibility of using the external force signals as control inputs. This is visualized in the block diagram of Fig. 4.2 by the connection of the external forces and the controller block.

Another view on the control problem involved with a real-time hybrid test is given in Fig. 4.3: The parallel

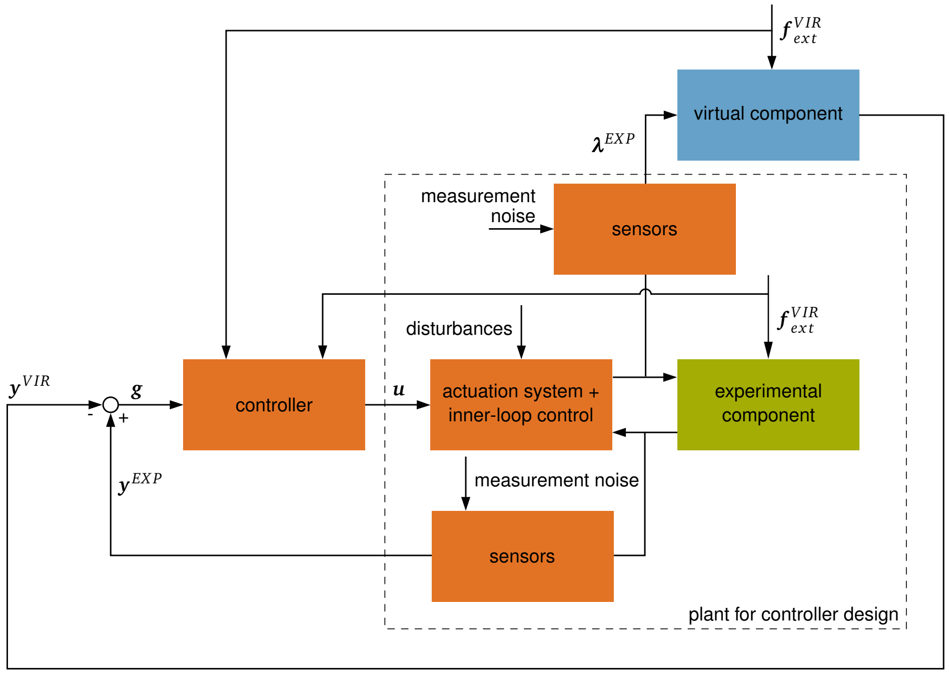


Figure 4.2: Control structure of a real-time hybrid test with displacement-controlled actuator

structure allows to consider the contribution of the virtual component in the controller design. The system structure is equivalent to the structure shown in Fig. 4.2.

Since sensors are usually selected such that they exhibit a bandwidth which is an order of magnitude wider than the actuators' bandwidth (see [79]), the sensor dynamics are omitted in the following considerations. However, as it can be seen in the block diagram of Fig. 4.2, the missing sensor dynamics do not alter the structure of the overall control system, and all approaches are equivalently valid if the sensor dynamics are considered. Because the external forces $\boldsymbol{f}_{ext}^{EXP}$ on the experimental component are usually not known, they are omitted as an input to the controller in the restructured model without loss of generality.

The open-loop state-space dynamics (see also Sec. 2.1) are given in Eq. (4.1),

$$\begin{aligned} \dot{\boldsymbol{x}} &= \underbrace{\begin{bmatrix} \boldsymbol{A}^{VIR} & \boldsymbol{B}_{\lambda}^{VIR}\boldsymbol{C}_{\lambda}^{TR} \\ \boldsymbol{0} & \boldsymbol{A}^{TR} \end{bmatrix}}_{\boldsymbol{A}} \boldsymbol{x} + \underbrace{\begin{bmatrix} \boldsymbol{B}_{\lambda}^{VIR}\boldsymbol{D}_{\lambda,u}^{TR} \\ \boldsymbol{B}_{u}^{TR} \end{bmatrix}}_{\boldsymbol{B}_u} \boldsymbol{u} + \underbrace{\begin{bmatrix} \boldsymbol{B}_{ext}^{VIR} & \boldsymbol{B}_{\lambda}^{VIR}\boldsymbol{D}_{\lambda,ext}^{TR} \\ \boldsymbol{0} & \boldsymbol{B}_{ext}^{TR} \end{bmatrix}}_{\boldsymbol{B}_{ext}} \boldsymbol{f}_{ext} \\ \boldsymbol{g} &= \underbrace{\begin{bmatrix} \boldsymbol{C}_{y}^{VIR} & -\boldsymbol{C}_{y}^{TR} \end{bmatrix}}_{\boldsymbol{C}} \boldsymbol{x} \quad \text{with} \quad \boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}^{VIR^T} & \boldsymbol{x}^{TR^T} \end{bmatrix}^T \quad \text{and} \quad \boldsymbol{f}_{ext} = \begin{bmatrix} \boldsymbol{f}_{ext}^{VIR^T} & \boldsymbol{f}_{ext}^{TR^T} \end{bmatrix}^T, \end{aligned} \tag{4.1}$$

where $\boldsymbol{x}$ is the state vector of the overall system, and the state space models of the subcomponents are used according to Sec. 2.1. The interface force vector $\boldsymbol{\lambda}^{VIR}$ in Eq. (2.10) is eliminated by applying the actuation forces $\boldsymbol{\lambda}^{VIR}$—an output of the actuation system—to the virtual component. In practice, applying the interface forces $\boldsymbol{\lambda}^{EXP}$ on the virtual component involves measuring the interface forces between the actuator system and the experimental component. In the same way, the interface displacements of the experimental component have to be measured. Actuator demand and external forces on both subcomponents are defined as system inputs, and the interface gap $\boldsymbol{g}$ is defined as system output. The interface gap $\boldsymbol{g}$ is the difference between the interface displacements of the virtual and the experimental component.

The control objective is to close the interface gap $\boldsymbol{g}$. The external forces can be seen as an external dis-

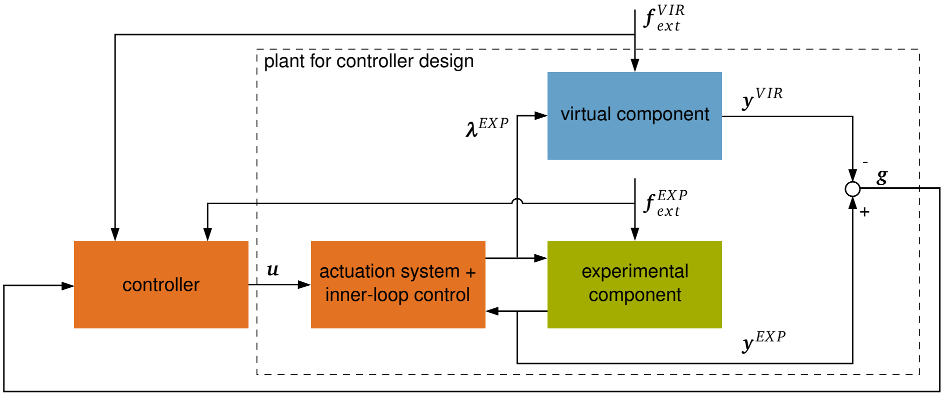


Figure 4.3: General control structure: Interface equilibrium is enforced by imposing the measured interface forces on the virtual component. A feedback controller reduces the interface gap g such that the system meets the compatibility constraints.

turbance to the open-loop system. The problem of rejecting the responses due to the disturbances can be addressed by applying a feedforward control law. In this way, the control structure of Fig. 4.3 is used in Chap. 5 where an adaptive feedforward control law replaces or complement the feedback controller.

Force-controlled actuators The control structure of a real-time hybrid test which is based on force-controlled actuators is given in Fig. 4.4. In this case, the measured displacements at the interface of the experimental component y^{EXP} are the input to the virtual component. The resulting interface forces λ^{VIR} acting on the virtual component serve as a demand signal for the actuation system. In other words, the controller shapes the actuator input such that the interface equilibrium constraint is met. The corresponding open loop state-space system is

$$\begin{gathered}
\dot{x} = \underbrace{\begin{bmatrix} A^{VIR} & B_y^{VIR} C_y^{TR} \\ 0 & A^{TR} \end{bmatrix}}_{A} x + \underbrace{\begin{bmatrix} 0 \\ B_u^{TR} \end{bmatrix}}_{B_u} u + \underbrace{\begin{bmatrix} B_{ext}^{VIR} & 0 \\ 0 & B_{ext}^{TR} \end{bmatrix}}_{B_{ext}} f_{ext} \\
g_\lambda = \underbrace{\begin{bmatrix} C_\lambda^{VIR} & -C_\lambda^{TR} \end{bmatrix}}_{C_\lambda} x + \underbrace{\begin{bmatrix} D_{\lambda,u}^{VIR} & -D_{\lambda,u}^{TR} \end{bmatrix}}_{D_{\lambda,u}} u + \underbrace{\begin{bmatrix} D_{\lambda,ext}^{VIR} & -D_{\lambda,ext}^{TR} \end{bmatrix}}_{D_{\lambda,ext}} f_{ext} \\
\text{with} \quad x = \begin{bmatrix} x^{VIR^T} & x^{EXP^T} \end{bmatrix}^T \quad \text{and} \quad f_{ext} = \begin{bmatrix} f_{ext}^{VIR^T} & f_{ext}^{EXP^T} \end{bmatrix}^T
\end{gathered} \tag{4.2}$$

Differences between the approach based on force-controlled actuators and the approach based on displacement-controlled actuators are discussed in the following section (Sec. 4.2).

4.2 Stability and Accuracy Issues

Stability issues which occur when coupling two numerical structures can be handled by choosing an appropriate time integration scheme. In contrast, real-time hybrid tests include inherent and unavoidable actuator dynamics and delays. Since during the tests forces and displacements are exchanged between the subcomponents, a closed-loop is established, and the delays can deteriorate the stability and accuracy of the tests. This relation between delay and stability is discussed in many control theory textbooks as e.g [72]. In the following section, the causes of the instabilities in real-time hybrid tests are analyzed. Approaches to assess the stability of real-time hybrid tests are discussed in the literature:

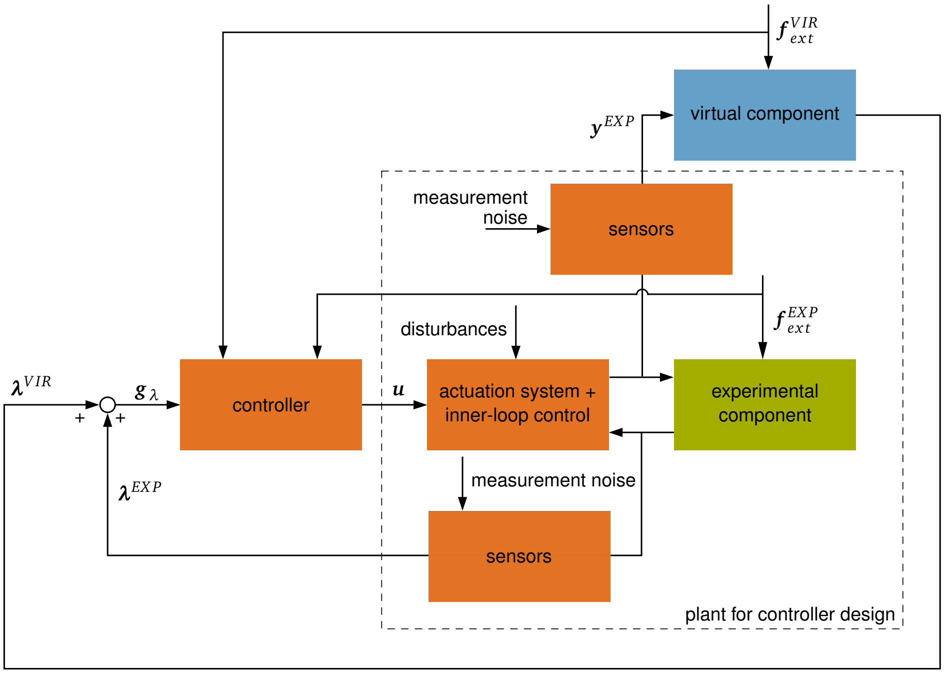


Figure 4.4: Control structure of a real-time hybrid test with force-controlled actuator

In [58], the stability analysis is performed on a system with a one-DOF oscillator and a pendulum system. [104] investigates the influence of the interface location on the stability properties of the coupled system. In Chap. 1 of this thesis, the relationship between interaction control in robotics and real-time hybrid testing has been mentioned. Stability analyses of simple lumped mass systems with the purpose of estimating the properties of robotic interaction controllers are presented in [44] and [60].

4.2.1 Sources of delay and time lags

The following processes can cause time delays and time lags in real-time hybrid tests:

Actuator Dynamics The dynamics of the actuators cause frequency-dependent time lags. Those time lags are discussed in detail in Chap. 3. Significant actuator dynamics occur in most real-time hybrid tests. They are the dominant source of delays in many test setups.

Time Integration The computation of the virtual component is discussed in Chap. 2. The computation process can introduce a constant time delay. Implicit time integration schemes inherently exhibit a time delay of one integration time step.

Communication Communication processes between different components of the test can introduce delays into the system. In [23], a test setup is described which makes use of network communication in order to exchange information between the simulation model, the control system, the actuators, and the sensors. Among others, the User Datagram Protocol (UDP) is applied. Such communication processes are not deterministic by nature and can go hand in hand with time-varying delays.

Signal Processing Signal processing and data acquisition can add significant delays. An example is given in [23], where computer vision is used to measure the system state of a floating wind turbine. The

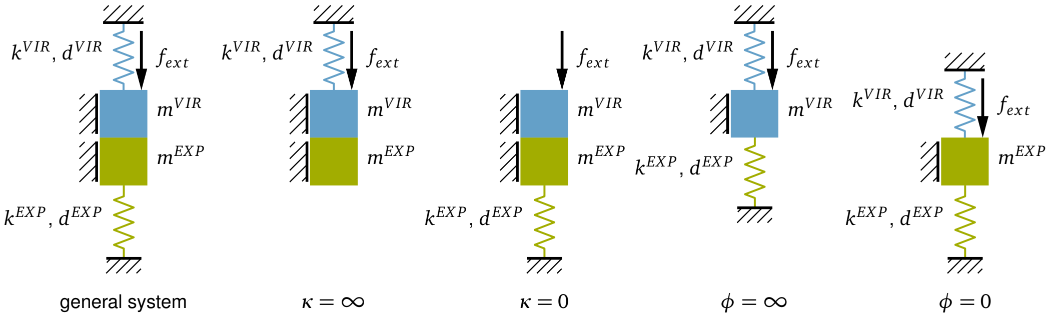


Figure 4.5: Investigated coupled system with special cases of mass ratios $\phi = {}^{m^{VIR}}/_{m^{EXP}}$ and stiffness ratios $\kappa = {}^{k^{VIR}}/_{k^{EXP}}$

system allows measuring position and orientation of the wind turbine structure. It consists of several cameras and markers which are attached to the structure. A computer system processes the camera data and sends the information about the system state to the control system. Obviously, the result of this process is a delay.

4.2.2 Effects of Delays

In the following section the influence of delay and other system parameters—as mass, damping and stiffness—on the behavior of real-time hybrid tests is investigated. For this purpose, two linear, single-DOF subcomponents are used. Fig. 4.5 shows the coupled system and some special cases of parameter combinations. Those systems serve as a model which helps to understand the basic relations between system parameters and stability properties. For most applications, those lumped-mass systems are rough simplifications: The actuators, in general, exhibit more complex dynamics which means that they have a frequency dependent transfer function. The same is true for the subcomponents which can be systems with a high number of DOFs or which can be continuous flexible systems. All components may exhibit a nonlinear behavior. The control system also may be more sophisticated in an application case. Nevertheless, the analysis is helpful to get a rough estimate of the stability properties of a system in the design phase of the test.

Non-Dimensional Form In order to be able to make general statements, the objective is to express the system dynamics in a non-dimensional form. The equations of motion of the two lumped-mass subcomponents from Fig. 4.5 read

$$m^{VIR}\ddot{x}^{VIR} + d^{VIR}\dot{x}^{VIR} + k^{VIR}x^{VIR} = -\lambda^{EXP} + f_{ext} \tag{4.3}$$

$$m^{EXP}\ddot{x}^{EXP} + d^{EXP}\dot{x}^{EXP} + k^{EXP}x^{EXP} = \lambda^{EXP}, \tag{4.4}$$

where x^* are the system coordinates, m^* are masses, d^* are the viscous damping constants, and k^* are the stiffness constants.The superscripts EXP and VIR denote the experimental and the virtual component. λ^{EXP} is the interface force and f_{ext} is the external force.

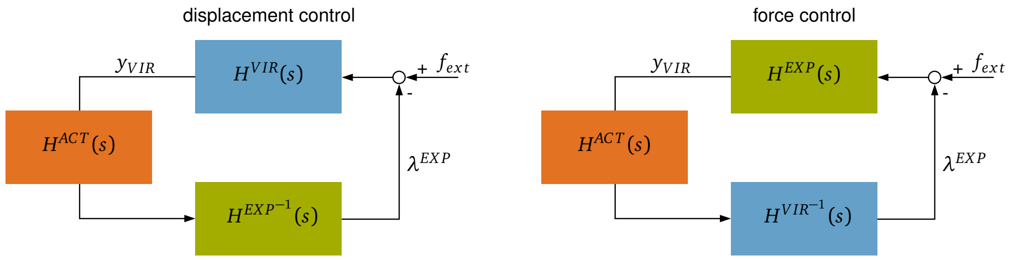


Figure 4.6: System with delay used for stability analysis

The corresponding Laplace domain transfer functions between forces and displacements are

$$\begin{aligned} H^{VIR}(s) &= \frac{\frac{1}{m^{VIR}}}{s^2 + 2\zeta^{VIR}\omega_0^{VIR}s + \omega_0^{VIR^2}} \quad \text{and} \quad H^{EXP}(s) = \frac{\frac{1}{m^{EXP}}}{s^2 + 2\zeta^{EXP}\omega_0^{EXP}s + \omega_0^{EXP^2}} \\ \text{with} \quad \zeta^{VIR} &= \frac{d^{EXP}}{2\sqrt{m^{VIR}k^{VIR}}}, \quad \zeta^{EXP} = \frac{d^{EXP}}{2\sqrt{m^{EXP}k^{EXP}}}, \\ \omega_0^{VIR} &= \sqrt{\frac{k^{VIR}}{m^{VIR}}} \quad \text{and} \quad \omega_0^{EXP} = \sqrt{\frac{k^{EXP}}{m^{EXP}}} \end{aligned} \tag{4.5}$$

The actuator dynamics are assumed to be pure delays T_d for the purpose of the analysis:

$$H^{ACT}(s) = \mathrm{e}^{-T_d s} \tag{4.6}$$

The transfer function of the coupled reference system is

$$H^{REF}(s) = \frac{1}{(m^{VIR} + m^{EXP})s^2 + (d^{VIR} + d^{EXP})s + k^{VIR} + k^{EXP}}. \tag{4.7}$$

This simplified structure of a real-time hybrid test is shown in Fig. 4.6. The analysis is first performed based on a displacement controlled experimental component. Because of the simple system structure, it is straightforward to apply the results from the displacement-control tests on tests with force-controlled actuators. This is discussed in Sec. 4.2.4. The open-loop transfer function $H_{OL}(s)$ reads

$$H_{OL}(s) = H^{EXP^{-1}}(s)H^{VIR}(s)H^{ACT}(s) = \frac{m^{EXP}}{m^{VIR}} \cdot \frac{s^2 + 2\zeta^{EXP}\omega_0^{EXP}s + \omega_0^{EXP^2}}{s^2 + 2\zeta^{VIR}\omega_0^{VIR}s + \omega_0^{VIR^2}} \cdot \mathrm{e}^{-T_d s}. \tag{4.8}$$

The inverse of the transfer function of the experimental component is applied here since the displacement output of the virtual component serves as an input to the experimental component and the interface forces are an output of the experimental component. The eigenfrequency of the experimental component appears as a zero in the nominator and as an anti-resonance peak in the open-loop transfer function.
The objective of the analysis is to be able to make statements on the properties of a coupled test independent of the absolute values of masses, damping constants and stiffness constants. The properties of a single DOF linear dynamic system can be described by its damping ratio and its eigenfrequency which are dimensionless quantities. Since in a coupled test it is necessary to account for the different contributions of the two subcomponents, the stiffness ratio κ and the mass ratio ϕ are used as additional dimensionless system parameters:

$$\kappa = \frac{k^{VIR}}{k^{EXP}} \quad \text{and} \quad \phi = \frac{m^{VIR}}{m^{EXP}} \tag{4.9}$$

The squared eigenfrequency of the coupled system ω_0^2 can be expressed as a function of the stiffness ratio κ, the mass ratio ϕ, and the squared eigenfrequency of the experimental component $\omega_0^{EXP^2}$.

Equivalently, it can be expressed as a function of κ, ϕ, and the squared eigenfrequency of the virtual component $\omega_0^{VIR^2}$.

$$
\begin{aligned}
\omega_0^2 &= \frac{k^{VIR}+k^{EXP}}{m^{VIR}+m^{EXP}} \\
&= \frac{(\kappa+1)k^{EXP}}{(\phi+1)m^{EXP}} = \frac{\kappa+1}{\phi+1}\omega_0^{EXP^2} \\
&= \frac{(1+\frac{1}{\kappa})k^{VIR}}{(1+\frac{1}{\phi})m^{VIR}} = \frac{1+\frac{1}{\kappa}}{1+\frac{1}{\phi}}\omega_0^{VIR^2}
\end{aligned}
\tag{4.10}
$$

Solving the expressions for $\omega_0^{EXP^2}$ and $\omega_0^{VIR^2}$ yields functions of the stiffness ratio κ, the mass ratio ϕ and the eigenfrequency of the coupled system ω_0^2:

$$
\omega_0^{EXP^2} = \frac{\phi+1}{\kappa+1}\omega_0^2 \quad \text{and} \quad \omega_0^{VIR^2} = \frac{1+\frac{1}{\phi}}{1+\frac{1}{\kappa}}\omega_0^2
\tag{4.11}
$$

Substituting $\omega_0^{EXP^2}$ and $\omega_0^{VIR^2}$ from Eqs. (4.11) in Eq. (4.8) yields

$$
H_{OL}(s) = \frac{1}{\phi} \cdot \frac{s^2 + 2\zeta^{EXP}\sqrt{\frac{\phi+1}{\kappa+1}}\omega_0 s + \frac{\phi+1}{\kappa+1}\omega_0^2}{s^2 + 2\zeta^{VIR}\sqrt{\frac{1+\frac{1}{\phi}}{1+\frac{1}{\kappa}}}\omega_0 s + \frac{1+\frac{1}{\phi}}{1+\frac{1}{\kappa}}\omega_0^2} \cdot \mathrm{e}^{-T_d s}
\tag{4.12}
$$

The complex frequency s can be substituted using the dimensionless complex frequency $\hat{s} = \frac{s}{\omega_0}$. The actuator time delay $T_d = \tau \cdot T_0 = \tau \cdot \frac{2\pi}{\omega_0}$ is expressed using the resonance period T_0 of the coupled system and the dimensionless delay variable τ. In other words, τ is the ratio between delay and resonance period. The fully dimensionless transfer function reads

$$
H_{OL}(\hat{s}) = \frac{1}{\phi} \cdot \frac{\hat{s}^2 + 2\zeta^{EXP}\sqrt{\frac{\phi+1}{\kappa+1}}\hat{s} + \frac{\phi+1}{\kappa+1}}{\hat{s}^2 + 2\zeta^{VIR}\sqrt{\frac{1+\frac{1}{\phi}}{1+\frac{1}{\kappa}}}\hat{s} + \frac{1+\frac{1}{\phi}}{1+\frac{1}{\kappa}}} \cdot \mathrm{e}^{-\tau 2\pi \hat{s}}.
\tag{4.13}
$$

This transfer function can be used to analyze the stability properties of coupled systems solely depending on the dimensionless system parameters κ, ϕ, ζ^{VIR}, ζ^{EXP} and τ.

Some special cases are shown in Fig. 4.5 and their open-loop transfer functions are given in Eqs. (4.15)-(4.17): For $\kappa = \infty$, the stiffness and the damping contributions of the experimental component disappear and the experimental component becomes a pure mass. For $\kappa = 0$ the virtual component becomes a pure mass. For $\phi = \infty$, the mass of the damping contributions of the experimental component disappear and the experimental component becomes a pure spring. And for $\phi = 0$, the virtual component becomes a pure spring.

$$
\kappa = \infty \qquad H_{OL}(\hat{s}) = \frac{1}{\phi} \cdot \frac{\hat{s}^2}{\hat{s}^2 + 2\zeta^{VIR}\sqrt{1+\frac{1}{\phi}}\hat{s} + 1 + \frac{1}{\phi}} \cdot \mathrm{e}^{-\tau 2\pi \hat{s}}
\tag{4.14}
$$

$$
\kappa = 0 \qquad H_{OL}(\hat{s}) = \frac{1}{\phi} \cdot \frac{\hat{s}^2 + 2\zeta^{EXP}\sqrt{\phi+1}\hat{s} + \phi + 1}{\hat{s}^2} \cdot \mathrm{e}^{-\tau 2\pi \hat{s}}
\tag{4.15}
$$

$$
\phi = \infty \qquad H_{OL}(\hat{s}) = \frac{\frac{1}{\kappa+1}}{\hat{s}^2 + 2\zeta^{VIR}\sqrt{\frac{1}{1+\frac{1}{\kappa}}}\hat{s} + \frac{1}{1+\frac{1}{\kappa}}} \cdot \mathrm{e}^{-\tau 2\pi \hat{s}}
\tag{4.16}
$$

$$
\phi = 0 \qquad H_{OL}(\hat{s}) = \frac{\hat{s}^2 + 2\zeta^{EXP}\sqrt{\frac{1}{\kappa+1}}\hat{s} + \frac{1}{\kappa+1}}{\frac{1}{1+\frac{1}{\kappa}}} \cdot \mathrm{e}^{-\tau 2\pi \hat{s}}
\tag{4.17}
$$

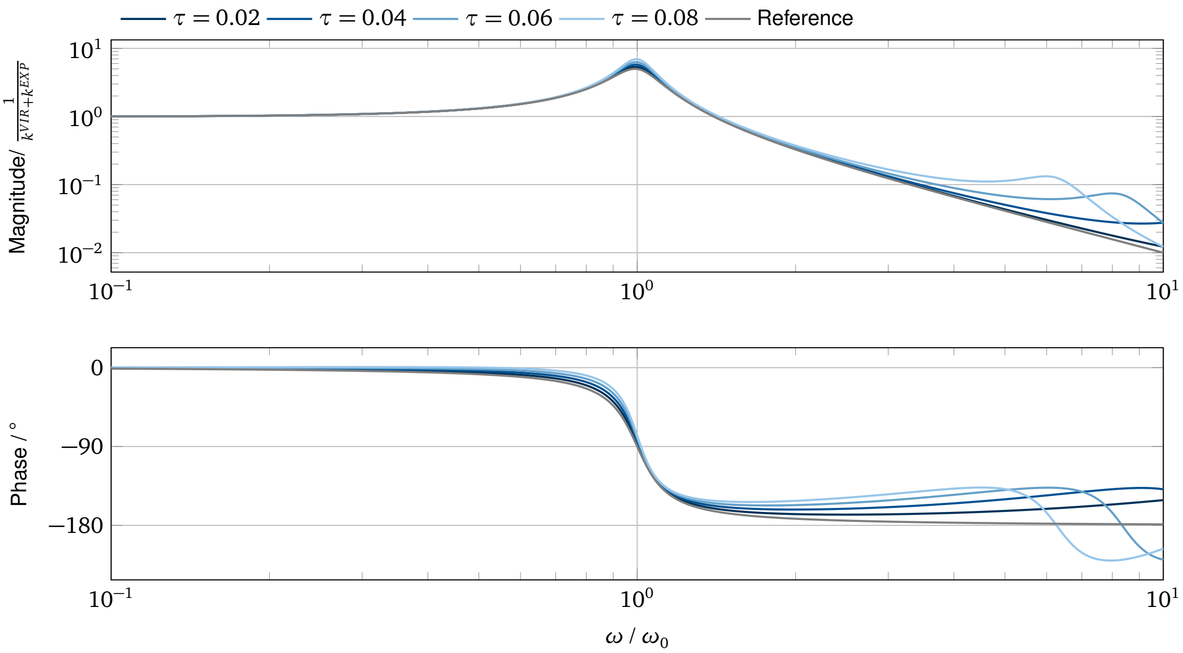


Figure 4.7: Bode plot of stable closed-loop transfer function $H_{CL}(\omega)$ with different delays τ (using $\kappa = 1$, $\phi = 1.5$, $\zeta^{EXP} = 0.1$ and $\zeta^{VIR} = 0.1$)

Closed-Loop Dynamics The closed-loop transfer function—which is used for assessing the stability of the system—reads

$$H_{CL}(\hat{s}) = \frac{H^{VIR}(\hat{s})}{1 + H_{OL}(\hat{s})} = \frac{H^{VIR}(\hat{s})}{1 + H^{EXP^{-1}}(\hat{s}) H^{VIR}(\hat{s}) H^{ACT}(\hat{s})}. \tag{4.18}$$

Fig. 4.7 shows the closed-loop transfer functions with different delays τ. All closed-loop transfer functions represent stable systems. In order to achieve this, the damping ratios are chosen as $\zeta^{EXP} = 0.1$ and $\zeta^{VIR} = 0.1$. The reference system dynamics are equivalent to a closed-loop system with delay $\tau = 0$:

$$\lim_{\tau \to 0} H_{CL}(s) = \frac{H^{VIR}(s) \cdot H^{EXP}(s)}{H^{VIR}(s) + H^{EXP}(s)} = H^{REF}(s) \tag{4.19}$$

Even for stable closed-loop systems the performance and accuracy of the test is deteriorated: a clear deviation of the phase and gain can be observed especially above the eigenfrequency. For the assessment of the closed-loop stability, the Pade approximation

$$\mathrm{e}^{-\tau 2 \pi \hat{s}} \approx \tilde{H}^{ACT}(\hat{s}) = \frac{1 - \pi \tau \hat{s}}{1 + \pi \tau \hat{s}} \tag{4.20}$$

is used. The system is assumed to be stable if all poles of $H_{CL}(\hat{s})$ have a negative real part. Fig. 4.8 shows the stability boundaries as a function of the mass ratio ϕ and the stiffness ratio κ. The contour lines indicate the maximum delay τ which allows stable coupling. The plot is given for four different damping values, which are applied to virtual and experimental component. This type of plot can be used for the rough classification of many real-time hybrid testing problems. The stable region has the smallest size for zero damping. With an increasing damping ratio, the size of the stable region increases. For values of $\phi < 1$—which means that the mass of the experimental component is greater than the mass of the virtual component—the test is always unstable. The physical behavior due to low values of ϕ is as follows: The high mass of the experimental component results in high forces which are applied on the virtual component. In turn, due to the lower mass of the virtual component, high forces cause larger

displacements which are fed back to the experimental component.
In the diagram for an undamped system with $\zeta^{EXP} = 0$ and $\zeta^{VIR} = 0$, a second condition for stability is $\kappa > \phi$. The physical behavior due to low values of κ is as follows: The high stiffness of the experimental component results in high forces which are applied on the virtual component. In turn, due to the lower stiffness of the virtual component, high forces cause larger displacements which are fed back to the experimental component. Higher damping ratios improve the stability properties of the tests.

Open-Loop Transfer Functions The reason of the unstable behavior of systems with $\phi < 1$ can be observed in the Bode plots of the open-loop transfer functions: The stability of a closed-loop system can be analyzed using its open-loop Nyquist curve in the complex plane according to the Nyquist criterion (see e.g the control theory textbook [34]). The simplified Nyquist criterion can be applied if

- the closed-loop system is $H_{CL} = \frac{1}{1+kH_{OL}(\hat{s})}$,
- the closed-loop system becomes unstable for an increased loop-gain k and
- the magnitude $|H_{CL}(\hat{s})|$ crosses 1 only once.

The simplified Nyquist criterion is defined by the following rule: The gain margin is defined as the separation of the gain $|H_{OL}(\hat{\omega})|$ from 1 at the frequency at which the phase angle $\angle H_{OL}(\hat{\omega})$ crosses the value $-180°$. If the gain is above 1 ($|H_{OL}(\hat{\omega})| > 0$) at this point, the system is unstable.
Even though the delay makes the Nyquist curves more complex and less straightforward to interpret, the phase decay caused by the delay can be observed in the bode plot: Fig. 4.9 shows the effect of different delays τ on the open-loop transfer function. It can be seen that the gain is not affected by the delay τ. However, the phase decays linearly with the frequency. The slope of the decay depends on the delay τ. For a zero delay $\tau = 0$, the phase remains at $\angle H_{OL} = 0$ as ω approaches infinity. For any delay $\tau > 0$, the phase falls below $-180°$ at one point. Since the limit of the gain of the open-loop transfer function is

$$\lim_{\omega\to\infty} |H_{OL}(\omega)| = \frac{1}{\phi}, \tag{4.21}$$

systems with $\phi < 1$ are never stable. A stable case ($\phi = 1.5$) as well as an unstable case $\phi = 0.5$ are shown in Fig. 4.9.
Fig. 4.10 shows the open-loop transfer functions for different locations in the stability diagram in Fig. 4.8. The reason for stability condition $\kappa > \phi$ for undamped systems can be gathered from the right diagram of Fig. 4.10. For values $\kappa < \phi$, the resonance appears at a lower frequency than the anti-resonance. The resonance introduces a negative phase shift of $-180°$ and the delay introduces an additional phase decay. Because the gain is above one in that frequency range, the closed-loop system is unstable. For values $\kappa > \phi$, the anti-resonance appears at a lower frequency than the resonance. The anti-resonance introduces a positive phase shift of $180°$ into the open-loop system and the closed-loop system is stable.

Damping Fig. 4.11 shows the effects of different damping ratios for both the experimental and the virtual component. Higher damping ratios smoothen the resonance peaks and the anti-resonances.
The resulting stability boundaries are shown in Fig. 4.12 and Fig. 4.13. The diagrams represent two slices—$\kappa = 1$ and $\phi = 1.5$—of a stability diagram as it is shown in Fig. 4.8. Fig. 4.12 shows the effects of variations in the damping ratio of the virtual component while the experimental component is kept undamped ($\zeta^{EXP} = 0$). The areas below the curves indicate delays τ which lead to a stable system. Because the increased damping pulls down the open-loop gain around the resonance, it improves the stability properties of the system. A jump can be observed at $\phi = 1$ for the diagram which shows curves with $\kappa = 1$ and at $\kappa = 1.5$ for the diagram which shows curves with $\phi = 1.5$. The reason is that the open-loop resonance and the open-loop anti-resonance overlap and change position in this point.

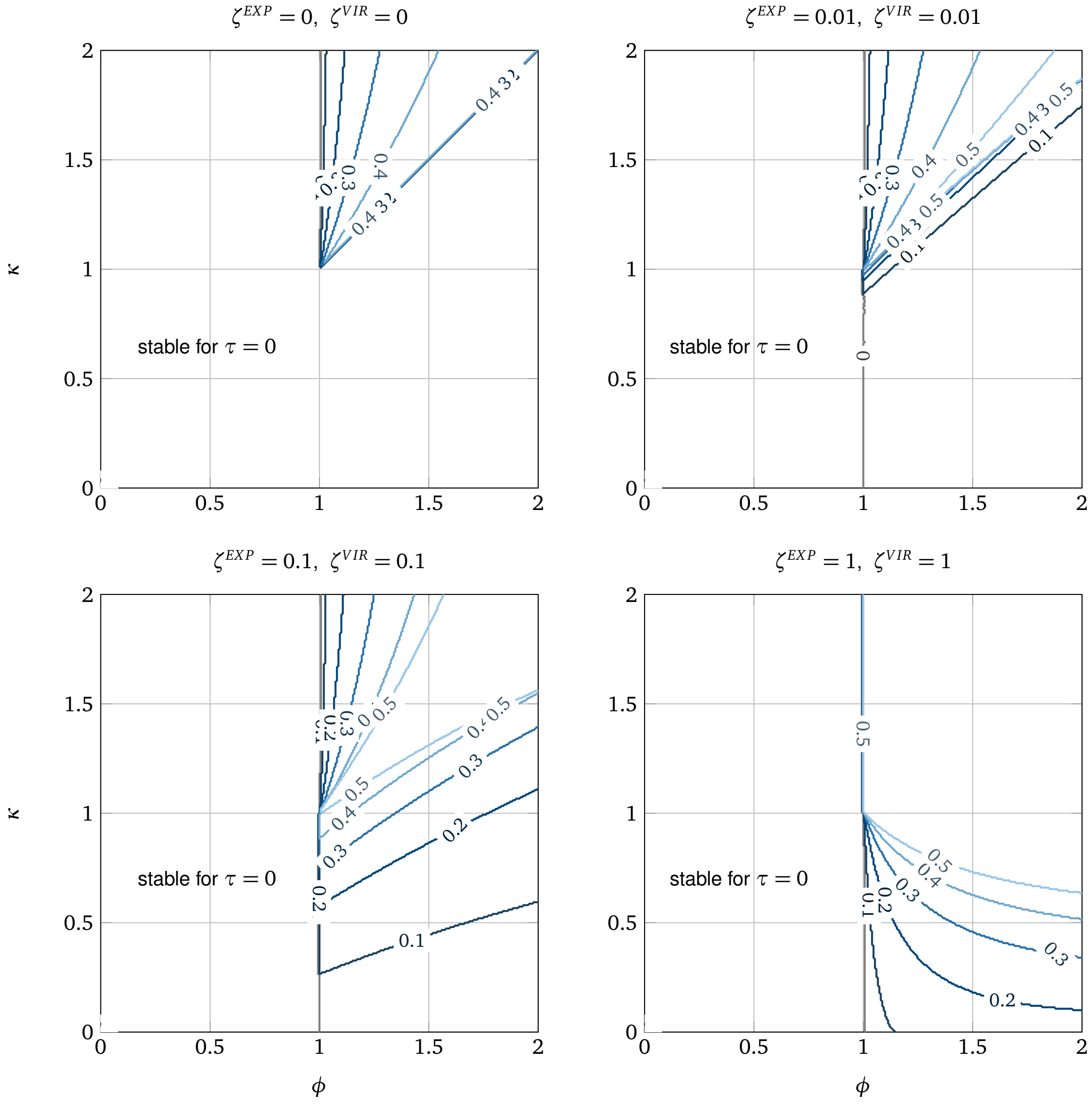


Figure 4.8: Displacement control: Stability boundaries as a function of the mass ratio ϕ and the stiffness ratio κ: the contour lines indicate the maximum delay τ which allows stable coupling. The curves were calculated numerically.

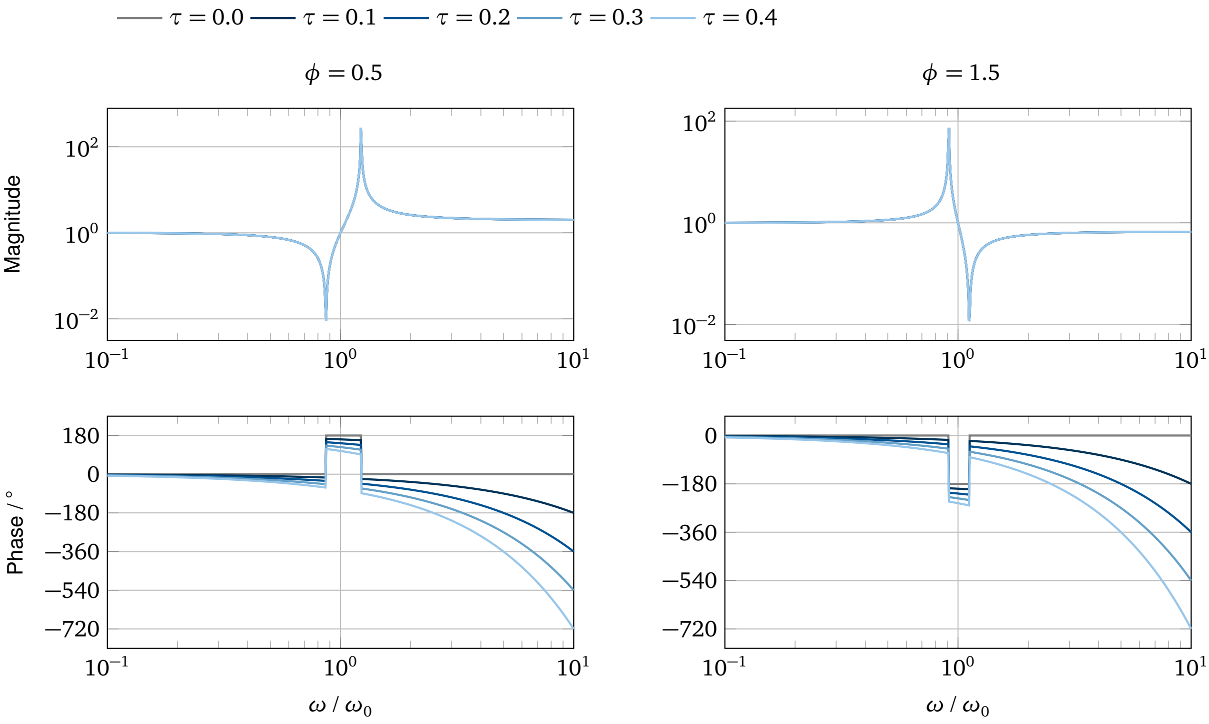


Figure 4.9: Bode plots open-loop transfer functions for different delays τ (using $\kappa = 1$, $\zeta^{EXP} = 0$ and $\zeta^{VIR} = 0$)

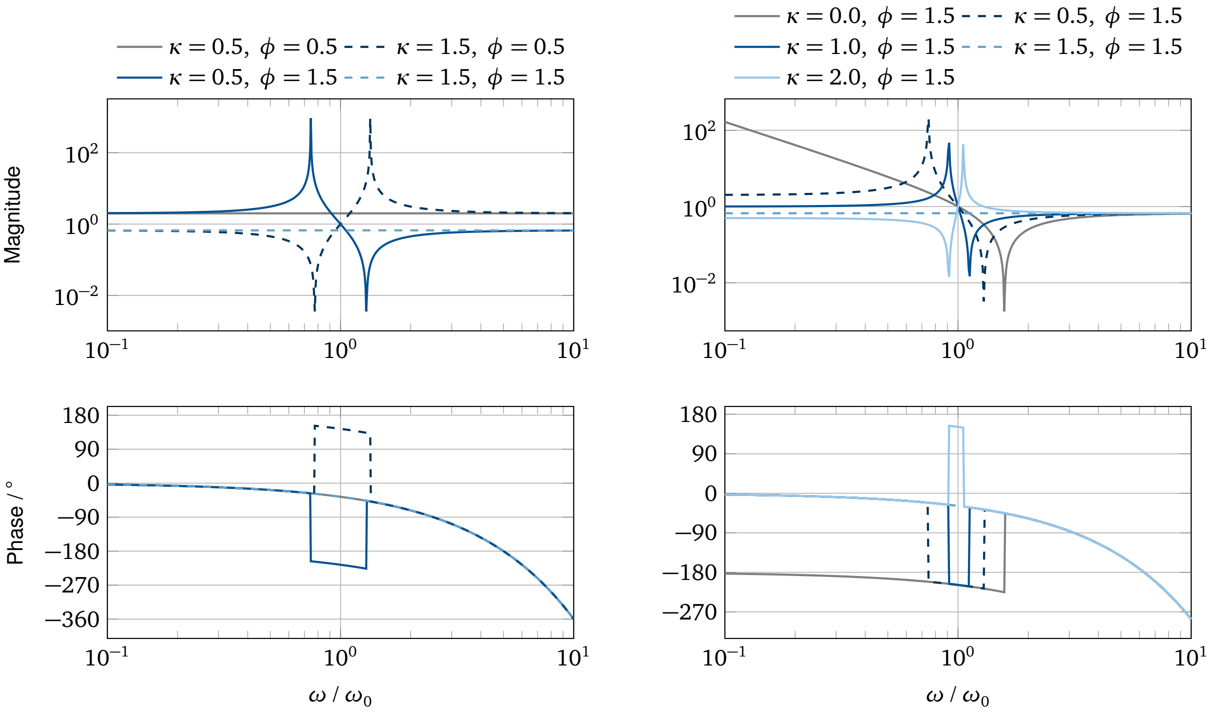


Figure 4.10: Bode plots of the open-loop transfer function $H_{OL}(\omega)$ for different combinations of stiffness ratio κ and mass ratio ϕ (using $\tau = 0.1$, $\zeta^{EXP} = 0$ and $\zeta^{VIR} = 0$)

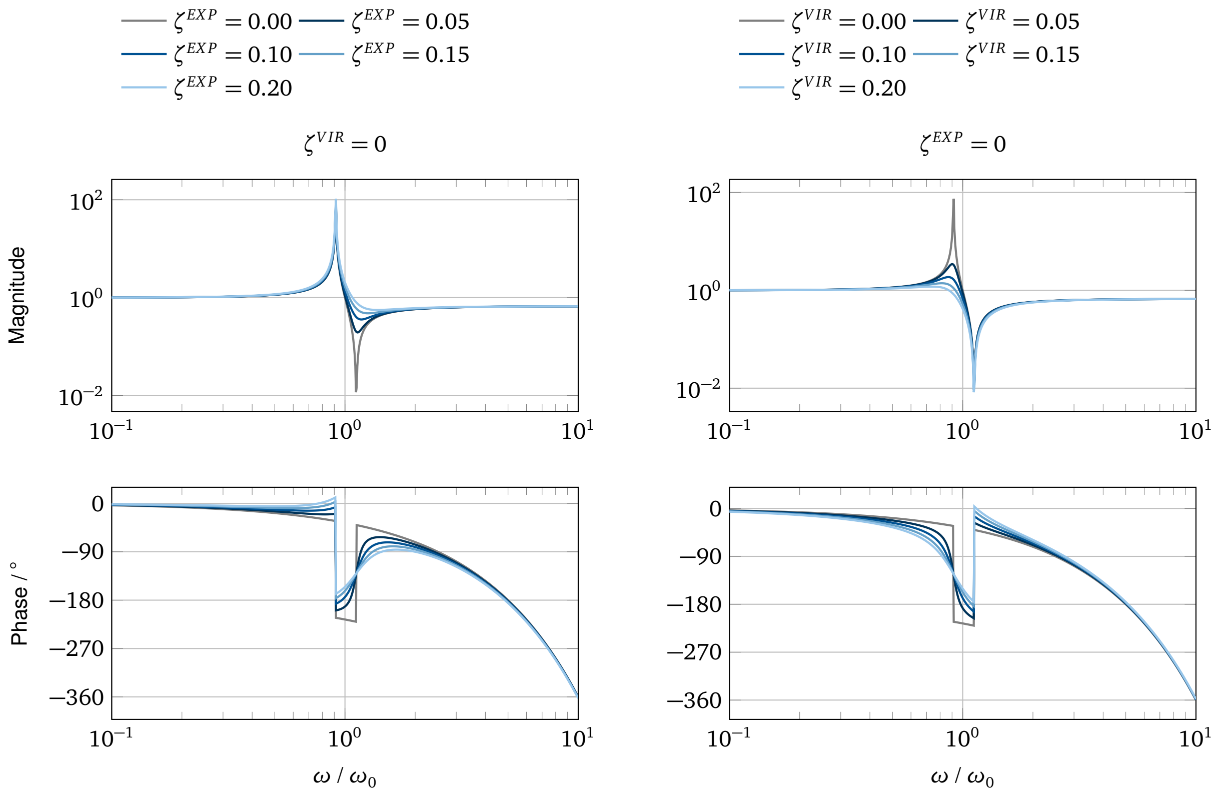


Figure 4.11: Bode plots of the open-loop transfer function $H_{OL}\omega$ for different damping ratios ζ^{EXP} and ζ^{VIR} of the virtual and the experimental component (using $\kappa = 1$, $\phi = 1.5$ and $\tau = 0.1$)

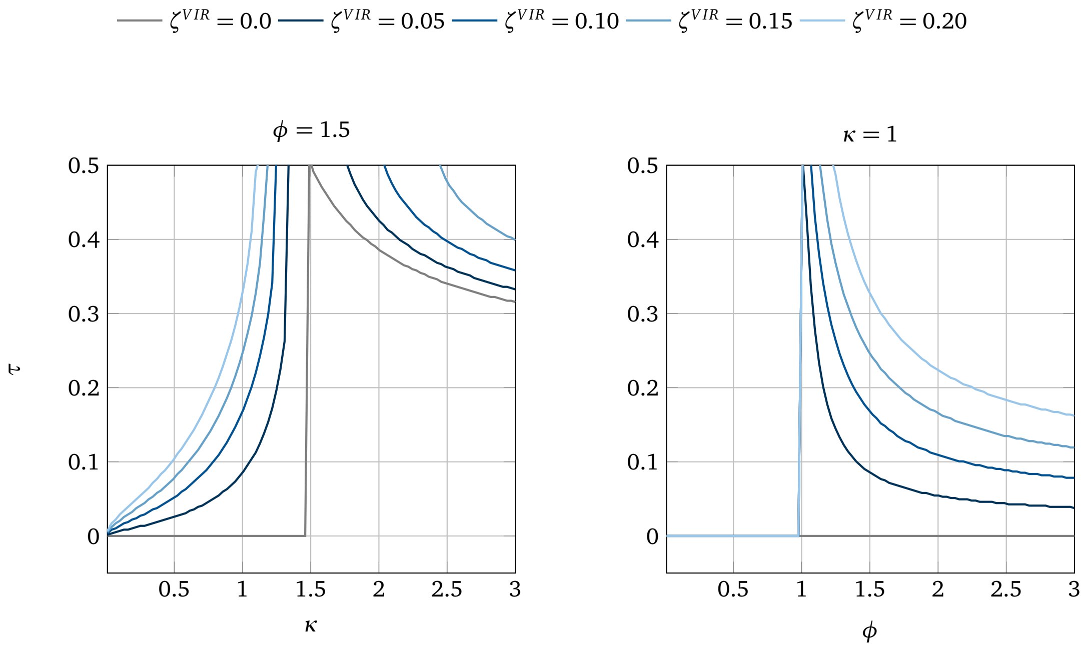


Figure 4.12: Stability boundaries for different damping ratios ζ^{VIR} of the virtual component (using $\zeta^{EXP} = 0$)

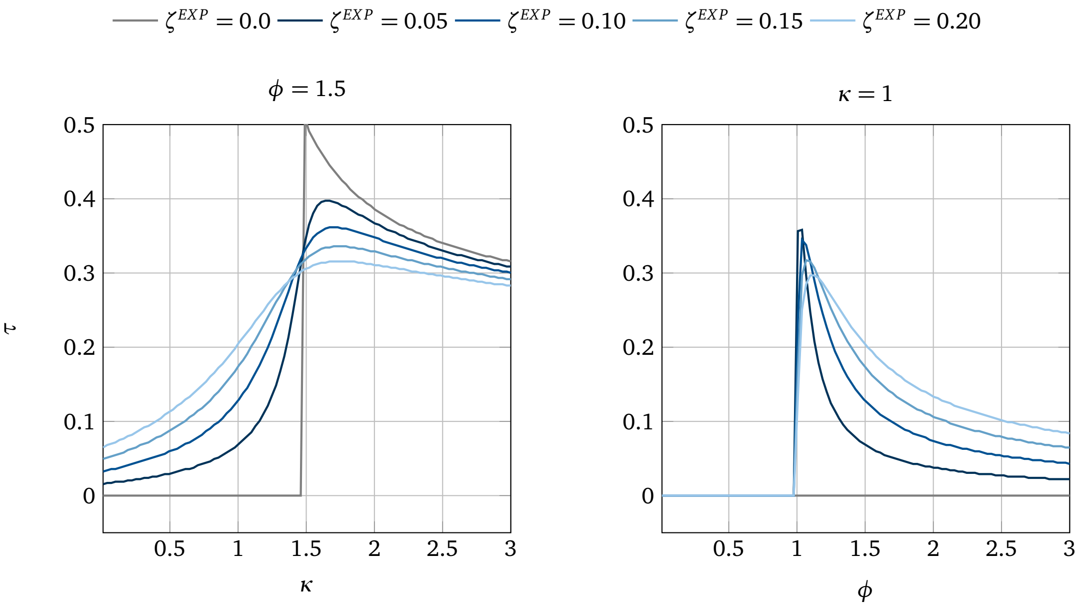


Figure 4.13: Stability boundaries for different damping ratios ζ^{EXP} of the experimental component (using $\zeta^{VIR} = 0$)

Fig. 4.13 shows the effects of variations in the damping ratio of the experimental component while the virtual component is kept undamped ($\zeta^{VIR} = 0$).

In general, additional damping has a positive effect on stability. The growth in size of the stable region is larger for variations of the damping ratio ζ^{VIR} of the virtual component than for variations of the damping ratio ζ^{EXP} of the experimental component. Increasing the damping ratio ζ^{EXP} has mainly a smoothening effect on the stability boundaries. Smooth stability boundaries can be helpful in practical applications to prevent unstable behavior when the system properties change during the test. This can be the case e.g. for systems with contacts.

4.2.3 Effect of Actuator Dynamics

The previous section describes the behavior of a simplified real-time hybrid test with pure delay. In reality, besides pure delay, the frequency-dependent actuator dynamics have a significant influence on the behavior of real-time hybrid tests. In this section, a system with an actuator as depicted in Fig. 1.2 is discussed. As in the previous section, the coupled reference system is a single-DOF oscillator and its transfer function reads

$$H^{REF}(s) = \frac{Y^{VIR}(s)}{F_{ext}(s)} = \frac{Y^{EXP}(s)}{F_{ext}(s)} = \frac{1}{(m^{VIR} + m^{EXP})s^2 + (d^{VIR} + d^{EXP})s + k^{VIR} + k^{EXP}} \tag{4.22}$$

with masses m^*, the viscous damping constants d^* and the stiffness constants k^*. The superscripts EXP and VIR denote the experimental and the virtual component and the superscript REF denotes the coupled reference system. $Y^{VIR}(s)$ and $Y^{EXP}(s)$ are the Laplace domain interface displacements and $F_{ext}(s)$ is the Laplace domain external force. A real-time hybrid test can be performed using the control structure which is described in Sec. 4.1. Fig. 4.14 shows a block diagram of this type of test with its specific transfer functions. The actuation system—a single-DOF oscillator—corresponds to the model of the voice-coil actuator in Chap. 3. The transfer function of the combined assembly of actuation system and experimental component reads

$$H^{TR}(s) = \frac{\Lambda^{EXP}(s)}{U(s)} = \frac{m^{EXP}s^2 + d^{EXP}s + k^{EXP}}{(m^{ACT} + m^{EXP})s^2 + (d^{ACT} + d^{EXP})s + k^{ACT} + k^{EXP}}, \tag{4.23}$$

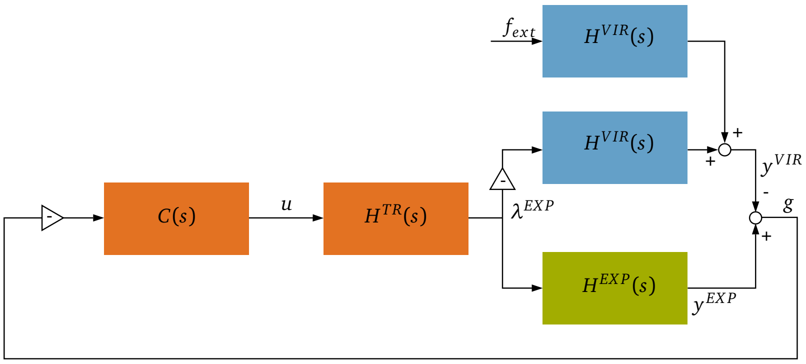


Figure 4.14: Block diagram of a system with actuator dynamics used for stability analysis.

where the superscript ACT denotes the actuation system and the superscript TR denotes the assembly of actuation system and experimental component. $\Lambda^{EXP}(s)$ is interface force and $U(s)$ is actuator input in Laplace domain.

The transfer function of the virtual component is

$$H^{VIR}(s) = \frac{Y^{VIR}(s)}{\Lambda^{EXP}(s)} = \frac{Y^{VIR}}{F_{ext}} = \frac{1}{m^{VIR}s^2 + d^{VIR}s + k^{VIR}}, \tag{4.24}$$

and the transfer function of the experimental component is

$$H^{EXP}(s) = \frac{Y^{EXP}(s)}{\Lambda^{EXP}(s)} = \frac{1}{m^{EXP}s^2 + d^{EXP}s + k^{EXP}}. \tag{4.25}$$

The controller $C(s)$ is a simple proportional gain and can be seen as a spring which connects the two subcomponents:

$$C(s) = k^P \tag{4.26}$$

Non-Dimensional Form According to the block diagram in Fig. 4.14, the open-loop transfer function is

$$H_{OL}(s) = C \cdot H^{TR}(s) \cdot \left(H^{VIR}(s) + H^{EXP}(s)\right). \tag{4.27}$$

The single transfer functions as they are listed in Eq. (4.23)- Eq. (4.26) depend on variables with units. However, the overall system properties can be described with the mass ratios ϕ^{ACT} and ϕ^{EXP}, the stiffness ratios κ^P, κ^{EXP} and κ^{ACT}, and the damping ratios ζ^{VIR}, ζ^{EXP} and ζ^{ACT}.

$$\begin{aligned} &\zeta^{VIR} = \frac{d^{VIR}}{2\sqrt{m^{VIR}k^{VIR}}}, \qquad \zeta^{ACT} = \frac{d^{ACT}}{2\sqrt{m^{ACT}k^{ACT}}}, \qquad \zeta^{EXP} = \frac{d^{EXP}}{2\sqrt{m^{EXP}k^{EXP}}}, \\ &\kappa^{EXP} = \frac{k^{EXP}}{k^{VIR}}, \qquad \kappa^{ACT} = \frac{k^{ACT}}{k^{VIR}}, \qquad \kappa^{P} = \frac{k^{P}}{k^{VIR}}, \\ &\phi^{EXP} = \frac{m^{EXP}}{m^{VIR}}, \qquad \phi^{ACT} = \frac{m^{ACT}}{m^{VIR}} \end{aligned} \tag{4.28}$$

Furthermore, the non-dimensional complex frequency $\hat{s} = \frac{s}{\omega_0}$ is used.

$$\hat{s} = \frac{s}{\omega_0} \qquad \text{with} \qquad \omega_0 = \sqrt{\frac{k^{EXP} + k^{VIR}}{m^{EXP} + m^{VIR}}} \tag{4.29}$$

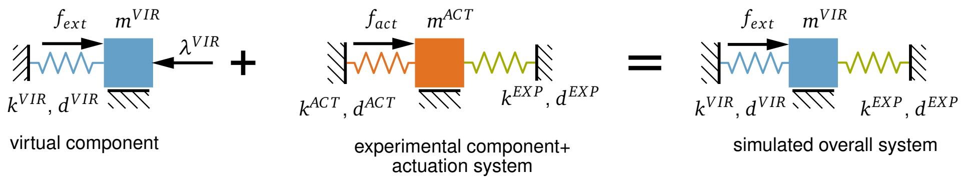


Figure 4.15: Investigated coupled system with actuator dynamics.

Recasting and substitution—which is described in Appendix C.1—results in the non-dimensional open-loop transfer function of Eq (4.30):

$$H_{OL}(\hat{s}) = \left(\phi^{EXP} \cdot \frac{\hat{s}^2 + 2\zeta^{EXP}\sqrt{\frac{\kappa^{EXP}}{\phi^{EXP}} \cdot \frac{\phi^{EXP}+1}{\kappa^{EXP}+1}} \cdot \hat{s} + \frac{\kappa^{EXP}}{\phi^{EXP}} \cdot \frac{\phi^{EXP}+1}{\kappa^{EXP}+1}}{\hat{s}^2 + 2\zeta^{VIR}\sqrt{\frac{\phi^{EXP}+1}{\kappa^{EXP}+1}}\hat{s} + \frac{\phi^{EXP}+1}{\kappa^{EXP}+1}} + 1 \right) \cdot \\ \cdot \frac{\frac{\kappa^{P}}{\phi^{ACT}+\phi^{EXP}} \cdot \frac{\phi^{EXP}+1}{\kappa^{EXP}+1}}{\hat{s}^2 + 2\left(\frac{\zeta^{ACT}\sqrt{\kappa^{ACT}\phi^{ACT}}}{\phi^{ACT}+\phi^{EXP}} + \frac{\zeta^{EXP}\sqrt{\kappa^{EXP}\phi^{EXP}}}{\phi^{ACT}+\phi^{EXP}} \right) \cdot \sqrt{\frac{\phi^{EXP}+1}{\kappa^{EXP}+1}} \cdot \hat{s} + \frac{\kappa^{ACT}+\kappa^{EXP}}{\phi^{ACT}+\phi^{EXP}} \cdot \frac{\phi^{EXP}+1}{\kappa^{EXP}+1}} \tag{4.30}$$

The following analysis is performed on a system with $\phi^{EXP} = 0$. This corresponds to the coupling of a virtual single-DOF oscillator with an experimental massless spring. A visualization of the system is shown in Fig. 4.15. Its open-loop transfer function reads

$$H_{OL}(\hat{s}) = \left(\frac{\frac{\kappa^{EXP}}{\kappa^{EXP}+1}}{\hat{s}^2 + 2\zeta^{VIR}\sqrt{\frac{1}{\kappa^{EXP}+1}} \cdot \hat{s} + \frac{1}{\kappa^{EXP}+1}} + 1 \right) \cdot \\ \cdot \frac{\frac{\kappa^{P}}{\phi^{ACT}} \cdot \frac{1}{\kappa^{EXP}+1}}{\hat{s}^2 + 2 \cdot \zeta^{ACT}\sqrt{\frac{\kappa^{ACT}}{\phi^{ACT}} \cdot \frac{1}{\kappa^{EXP}+1}} \cdot \hat{s} + \frac{\kappa^{ACT}+\kappa^{EXP}}{\phi^{ACT}} \cdot \frac{1}{\kappa^{EXP}+1}} \tag{4.31}$$

Closed-Loop Dynamics The closed-loop transfer function, which results from the structure given in Fig. 4.14, reads

$$H_{CL}(s) = \frac{Y^{VIR}(s)}{F_{ext}(s)} = \frac{H^{VIR}(s) \cdot \left(C(s) \cdot H^{TR}(s) \cdot H^{EXP}(s) + 1 \right)}{C(s) \cdot H^{TR}(s) \cdot (H^{EXP}(s) + H^{VIR}(s)) + 1}. \tag{4.32}$$

Its derivation is explained in Appendix C.2. Fig. 4.16 shows closed-loop bode plots of transfer functions for a range of values of κ^P. Starting from Eq. (4.32) and using $C(s) = \kappa^P k^{VIR}$, it is straightforward to show that as the loop-gain ratio κ^P approaches infinity, the closed-loop transfer behavior approaches the dynamics of the reference system:

$$\lim_{\kappa^P \to \infty} H_{CL}(s) = \frac{H^{VIR}(s) \cdot H^{EXP}(s)}{H^{VIR}(s) + H^{EXP}(s)} = H^{REF}(s) \tag{4.33}$$

The insight from Eq. (4.33) means that the closed-loop system is stable for $\kappa_P \to \infty$. A high control-gain in combination with a slight delay, however, will cause an unstable system behavior. This can be observed in Fig. 4.17 with the Nyquist criterion in mind: It shows the effects of a change in the control gain on an exemplary open-loop transfer function. The phase angle $\angle H_{OL}(\omega)$ for high frequencies and delay would fall below $-180°$ at one point. If the gain $|H_{OL}(\omega)|$ was above 1 at this point—due to the loop-gain κ^P—the system would be unstable.

Lower control gains, in turn, can cause an unstable behavior even without any delay. The root-locus

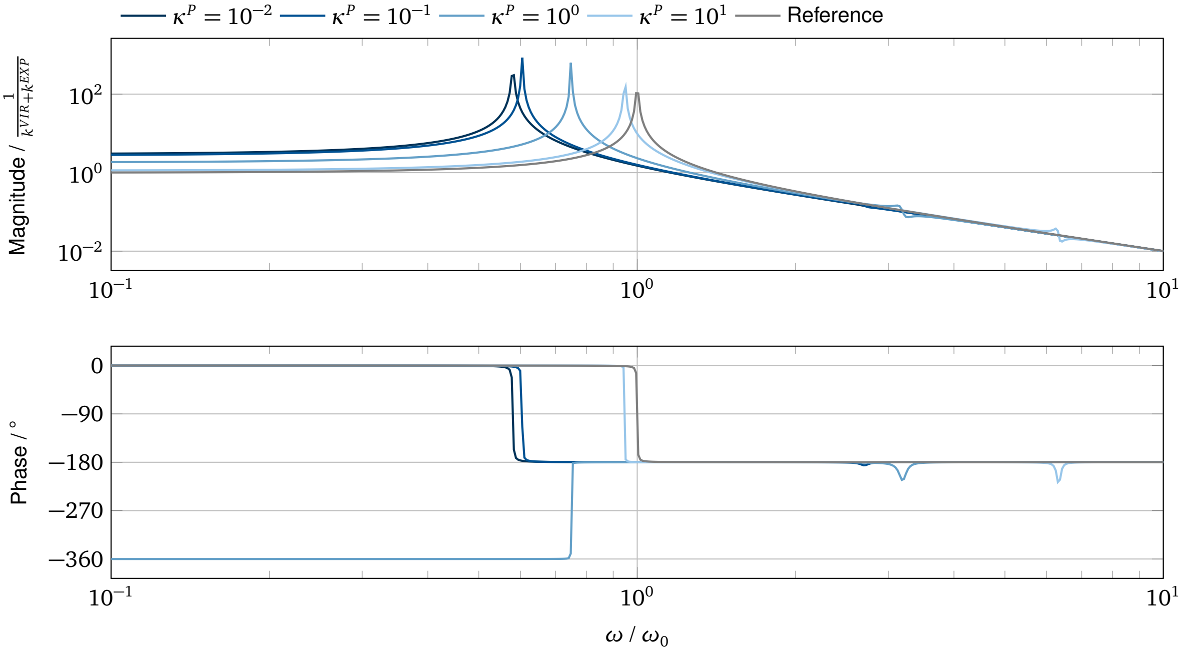


Figure 4.16: Bode plots of the closed-loop transfer function $H_{CL}(\omega)$ for different control gain ratios κ^P(using $\zeta^{VIR} = 2 \cdot 10^{-3}$, $\zeta^{ACT} = 0.1$, $\kappa^{EXP} = 2$, $\kappa^{ACT} = 0.1$ and $\phi^{ACT} = 0.1$)

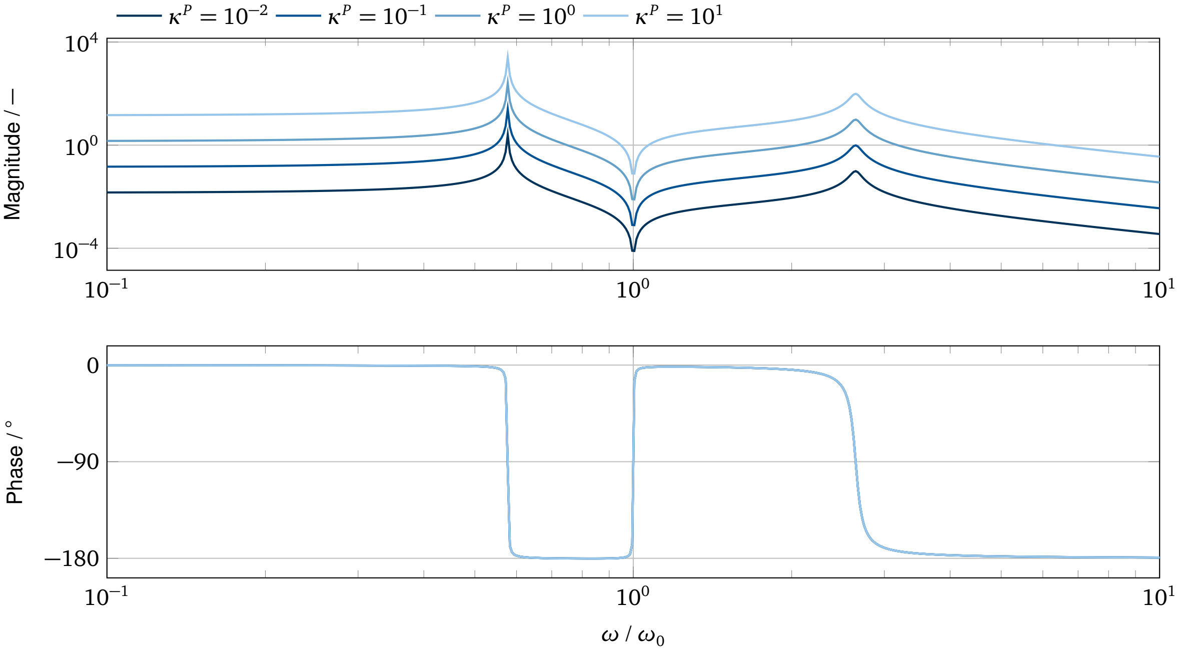


Figure 4.17: Bode plots of the open-loop transfer function $H_{OL}(\omega)$ for different control gain ratios κ^P (using $\zeta^{VIR} = 2 \cdot 10^{-3}$, $\zeta^{ACT} = 0.1$, $\kappa^{EXP} = 2$, $\kappa^{ACT} = 0.1$ and $\phi^{ACT} = 0.1$)

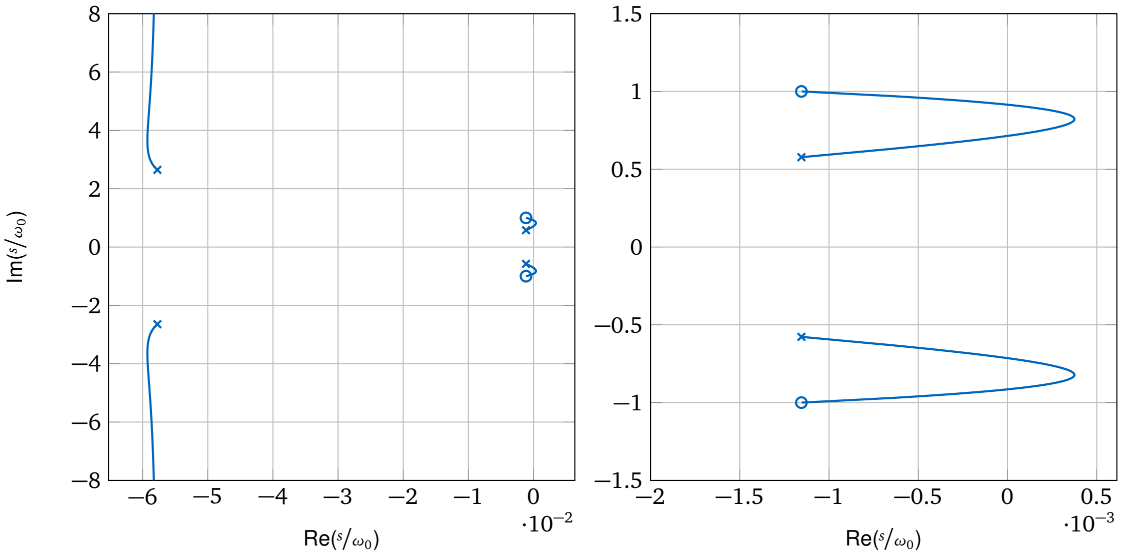


Figure 4.18: Root-locus plot for the closed-loop transfer function $H_{CL}(\omega)$ (using $\zeta^{VIR} = 2 \cdot 10^{-3}$, $\zeta^{ACT} = 0.1$, $\kappa^{EXP} = 2$, $\kappa^{ACT} = 0.1$ and $\phi^{ACT} = 0.1$). The curves show the trajectories of the poles as κ_P is varied from 0 to ∞. The open loop poles ($\kappa_P = 0$) are marked with "x" and the poles for $\kappa_P = \infty$ are marked with "o". The two subplots show different details of the same data.

plot of Fig. 4.18 shows the curves of the closed-loop poles describing a variation of the loop gain κ^P from 0 to ∞. The case shown in Fig. 4.18 illustrates that κ^P can take on values which make the system unstable i.e. when the real parts of the poles become positive and the poles move into the right half plane.

The stability of the closed-loop system can be assessed without explicitly calculating the poles of the system by applying the Routh–Hurwitz criterion. The denominator of the closed-loop system $H_{OL}(s)+1$ is a 4th-order polynomial of the form

$$P(s) = s^4 + a_1 s^3 + a_2 s^2 + a_3 s + a_4. \tag{4.34}$$

According to the Routh–Hurwitz criterion, all roots of $P(s)$ have only negative real parts if all coefficients a_1, a_2, a_3 and a_4 are positive and if

$$-a_1^2 a_4 - a_3^2 a_0 + a_1 a_2 a_3 > 0 \tag{4.35}$$

holds. Hence, the limit case

$$-a_1^2 a_4 - a_3^2 a_0 + a_1 a_2 a_3 = 0 \tag{4.36}$$

gives a non-linear equation for the stability boundary, which can be solved e.g. for the damping ratio of the virtual component ζ^{VIR} or for the actuator damping ratio ζ^{ACT}.

The effects of the loop gain ratio κ^P on the stability boundaries are shown in Fig. 4.19. The figure shows contour lines of the minimum damping ratio ζ^{VIR}, which leads to a stable system. As discussed above, the size of the stable region does not increase monotonically with the loop gain ratio κ^P. On the contrary, the minimum necessary damping ratio falls with increasing κ^P—after a rise for small values of κ^P.

Fig. 4.20 shows the stability boundaries as a function of the stiffness ratio κ^{ACT} and the stiffness ratio κ^{EXP} as well as the stability boundaries as a function of the damping ratio ζ^{ACT} and the stiffness ratio κ^{EXP}. Again, the contour lines indicate the minimum damping ratio ζ^{VIR}, which leads to a stable closed-loop system. The diagram shows the negative influence of actuator stiffness and damping on the stability properties of the system.

Fig. 4.21 shows a stability diagram which relates to the actuator dynamics. The stability boundaries are

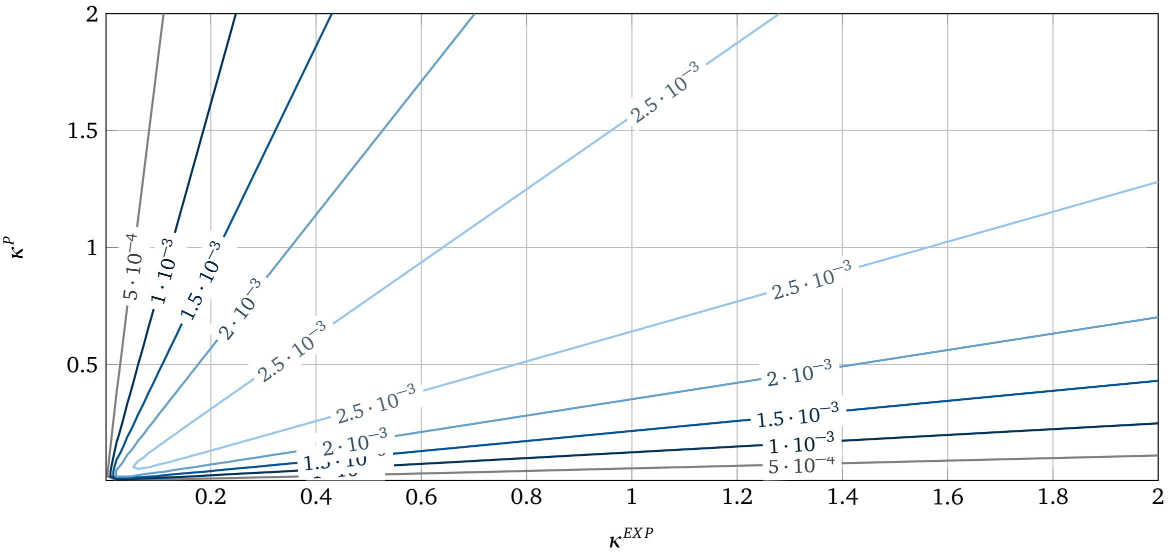


Figure 4.19: Stability boundaries as a function of the loop-gain ratio κ^P and the stiffness ratio κ^{EXP}. The contour lines indicate the minimum damping ratio ζ^{VIR} which leads to a stable system $H_{CL}(s)$ (using the constant values $\kappa^{ACT} = 0.1$, $\zeta^{ACT} = 10$ and $\phi^{ACT} = 0.1$).

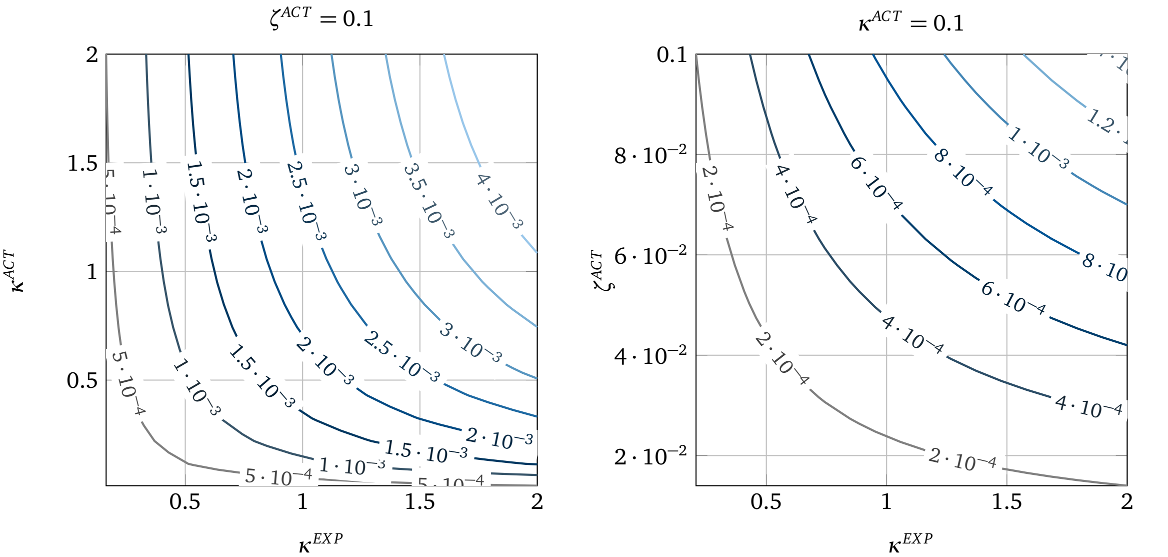


Figure 4.20: Stability boundaries as a function of the stiffness ratio κ^{ACT}, ζ^{ACT} and the stiffness ratio κ^{EXP}. The contour lines indicate the minimum damping ratio ζ^{VIR} which leads to a stable system $H_{CL}(s)$ (using the constant values $\zeta^{ACT} = 0.1$, $\kappa^P = 10$ and $\phi^{ACT} = 0.1$).

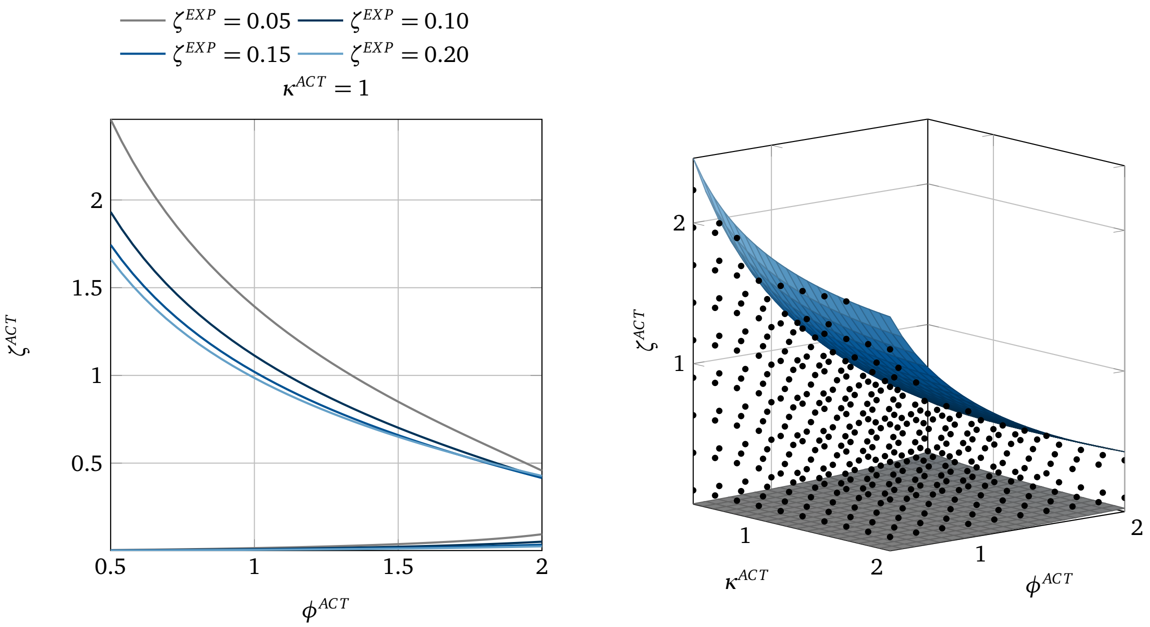


Figure 4.21: Stability boundaries as a function of the stiffness ratio κ^{ACT} and the stiffness ratio ϕ^{ACT}. The contour lines indicate the minimum damping ratio ζ^{ACT} which leads to a stable system $H_{CL}(s)$ (using the constant values $\zeta^{VIR} = 0.1$, $\kappa^{P} = 10$ and $\kappa^{EXP} = 1$). The dots mark the stable region.

shown as a function of the stiffness ratio κ^{ACT} and the mass ratio ϕ^{ACT}. Both, high actuator damping ratios ζ^{ACT} and high actuator mass ratios ϕ^{ACT} lead to an unstable system behavior. The reason is that additional actuator damping and masses make the dynamic responses of the test system slower. For all cases shown in Fig. 4.19 - Fig. 4.21, the stability properties improve with increasing stiffness ratio κ^{EXP}, which is a measure of the stiffness of the experimental component.

4.2.4 Types of inner-loop reference signals

In this section, the influence of different reference signal types for the inner-loop control on stability and accuracy of the tests is discussed. The discussion is based on the model with pure delay of Sec. 4.2.2.

Force and Displacement Control In Sec. 4.2.2, the effects of delays on tests with displacement control are analyzed. Fig. 4.6 shows the control structure for displacement control and force control of the experimental component. If force control is used instead of displacement control, nominator and denominator of the open-loop transfer function are exchanged, and the new open-loop transfer function reads

$$H_{OL}(s) = H^{EXP}(s)H^{VIR^{-1}}(s)H^{ACT}(s) = \phi \cdot \frac{\hat{s}^2 + 2\zeta^{VIR}\sqrt{\frac{1+\frac{1}{\phi}}{1+\frac{1}{\kappa}}}\hat{s} + \frac{1+\frac{1}{\phi}}{1+\frac{1}{\kappa}}}{\hat{s}^2 + 2\zeta^{EXP}\sqrt{\frac{\phi+1}{\kappa+1}}\hat{s} + \frac{\phi+1}{\kappa+1}} \cdot \mathrm{e}^{-\tau 2\pi\hat{s}} \tag{4.37}$$

The stability diagram of the force-controlled experimental component is given in Fig. 4.22. Compared to the stability diagram for displacement controlled experimental components in Fig. 4.22, the position of the stable regions are mirrored: Systems with $\phi > 1$—which means that the mass of the virtual component is higher than the mass of the experimental component—are always unstable. Systems with $\phi < 1$, in turn, can exhibit stable behavior. This is contrary to the findings relating to the displacement controlled systems. The reason for this condition is that the magnitude of the open-loop transfer function at high

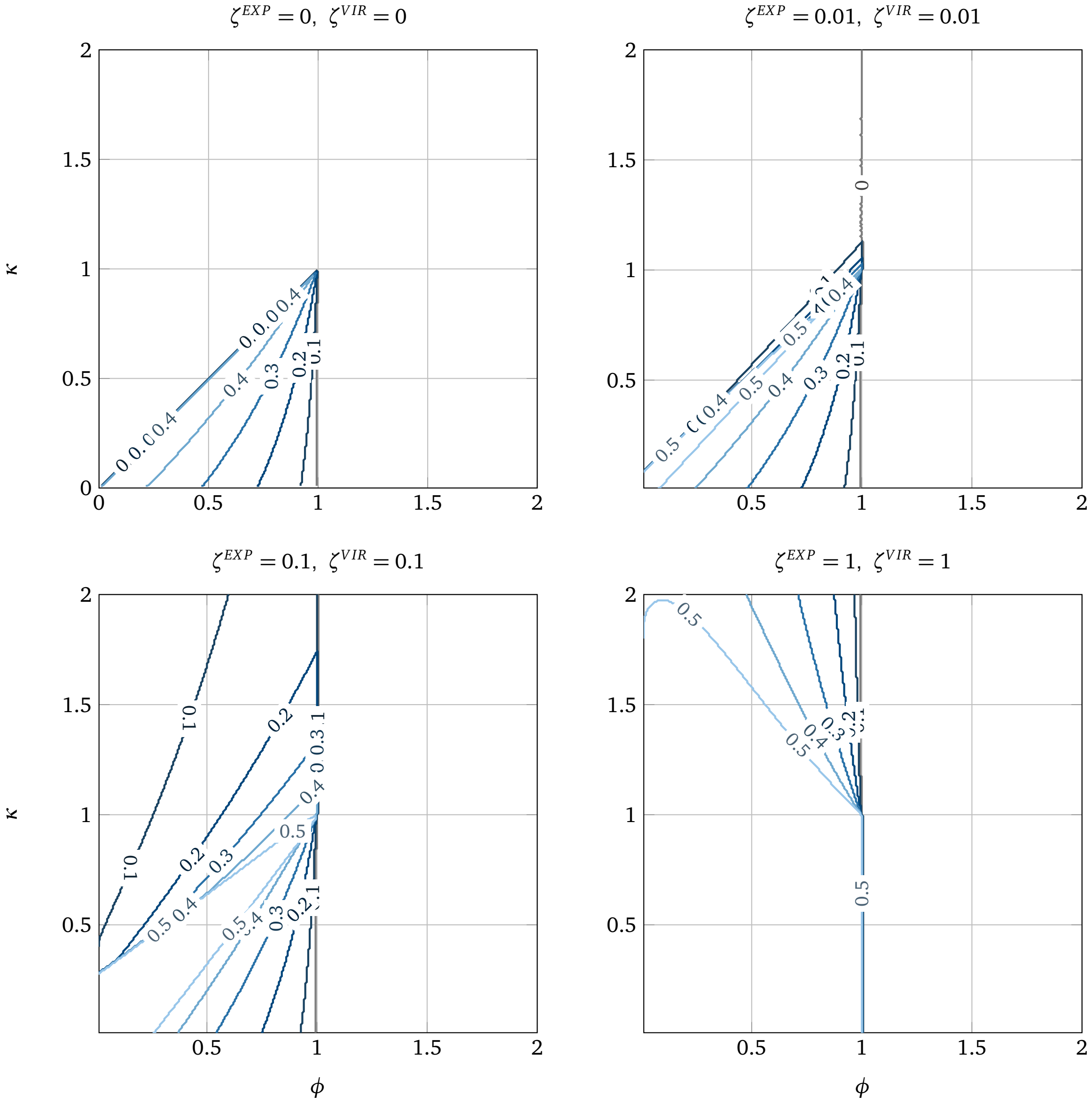


Figure 4.22: Force control: Stability boundaries as a function of the mass ratio ϕ and the stiffness ratio κ: the contour lines indicate the maximum delay τ which allows stable coupling.

frequencies is ϕ. Any delay will lead to a violation of the Nyquist criterion if $\phi > 1$. The physical behavior due to high values of ϕ is as follows: The high mass of the virtual component results in high forces which are applied to the experimental component. In turn, due to the lower mass of the experimental component, high forces cause larger displacements which are fed back to the virtual component.

An additional stability condition for systems without any damping is $\phi > \kappa$ since it leads to another violation of the Nyquist criterion. The order of pole and zero, which lead to the condition $\phi < \kappa$ for the displacement control in Sec. 4.2.2, is inverted. The physical behavior due to high values of κ is as follows: The high stiffness of the virtual component results in high forces which are applied to the experimental component. In turn, due to the lower stiffness of the experimental component, high forces cause larger displacements which are fed back to the virtual component.

A consequence of this observation is, that for systems where the mass of the experimental component is higher than the mass of the virtual component, it may be useful to apply force control.

Acceleration and Displacement Control In modal analysis, the common approach to measure the transfer behavior of structures is the use of acceleration sensors. The idea of applying those sensors in real-time hybrid tests leads to the need for an analysis of the effects of enforcing the compatibility constraint on acceleration level. The second-order time derivative is equivalent to multiplying the term s^2 with the transfer functions of the components. The resulting transfer function of the virtual component is

$$H^{VIR}(s) = \frac{Y^{VIR}(s)}{\Lambda^{EXP}(s)} = \frac{Y^{VIR}}{F_{ext}} = \frac{s^2}{m^{VIR}s^2 + d^{VIR}s + k^{VIR}}, \tag{4.38}$$

and the resulting transfer function of the experimental component is

$$H^{EXP}(s) = \frac{Y^{EXP}(s)}{\Lambda^{EXP}(s)} = \frac{s^2}{m^{EXP}s^2 + d^{EXP}s + k^{EXP}}. \tag{4.39}$$

Obviously in the pure delay model of Sec. 4.2.2, a change between displacement and acceleration control would make no difference, since the term s^2 simply cancels out. However, if actuator dynamics are present as described in Sec. 4.2.3, the use of acceleration signals for coupling the components changes the open loop transfer function. The new open-loop transfer function of a system with acceleration control reads

$$H_{OL}(\hat{s}) = \left(\frac{\frac{\kappa^{EXP}}{\kappa^{EXP}+1}}{\hat{s}^2 + 2\zeta^{VIR}\sqrt{\frac{1}{\kappa^{EXP}+1}} \cdot \omega_0 \cdot \hat{s} + \frac{1}{\kappa^{EXP}+1}} + 1 \right) \cdot \tag{4.40}$$

$$\cdot \frac{\frac{\kappa^{P}}{\phi^{ACT}} \cdot \frac{1}{\kappa^{EXP}+1} \cdot \hat{s}^2}{\hat{s}^2 + 2 \cdot \zeta^{ACT}\sqrt{\frac{\kappa^{ACT}}{\phi^{ACT}} \cdot \frac{1}{\kappa^{EXP}+1}} \cdot \hat{s} + \frac{\kappa^{ACT}+\kappa^{EXP}}{\phi^{ACT}} \cdot \frac{1}{\kappa^{EXP}+1}} \tag{4.41}$$

Note that the loop-gain is $k_P = \omega_0 \kappa_P k^{VIR}$ as consequence. Fig. 4.23 shows the Bode plot of $H_{OL}(\hat{s})$. The squared differentiator s^2 causes a positive 180° phase shift. As a result, the phase margin is higher at all frequencies. However, the magnitude of the open-loop transfer function stays constant as the frequency approaches infinity. This can lead to problems if a delay is involved in the test. On the other hand, due to the high magnitude of the open-loop transfer function, the accuracy of the closed-loop transfer function is improved at high frequencies. This can be observed in Fig. 4.24.

The magnitude of the open-loop transfer function goes to zero as the frequency approaches zero. This results in an uncompensated residual steady-state error. In other words, the use of acceleration signals for coupling can cause drift between the components. The closed-loop transfer function in Fig. 4.24 shows high deviations from the reference transfer function at low frequencies.

4.2.5 Summary

To summarize, the discussion on the simple lumped-mass systems leads to the following findings:

- Low mass ratios and stiffness ratios between virtual and experimental component can be problematic when using displacement control. Force control makes sense if the mass of the experimental component is higher than the mass of the virtual component.

- High mass ratios and stiffness ratios between virtual and experimental component can be problematic when using force control. Displacement control makes sense if the mass of the virtual component is higher than the mass of the experimental component.

- Damping in the experimental and in the virtual component improves the stability properties of a test.

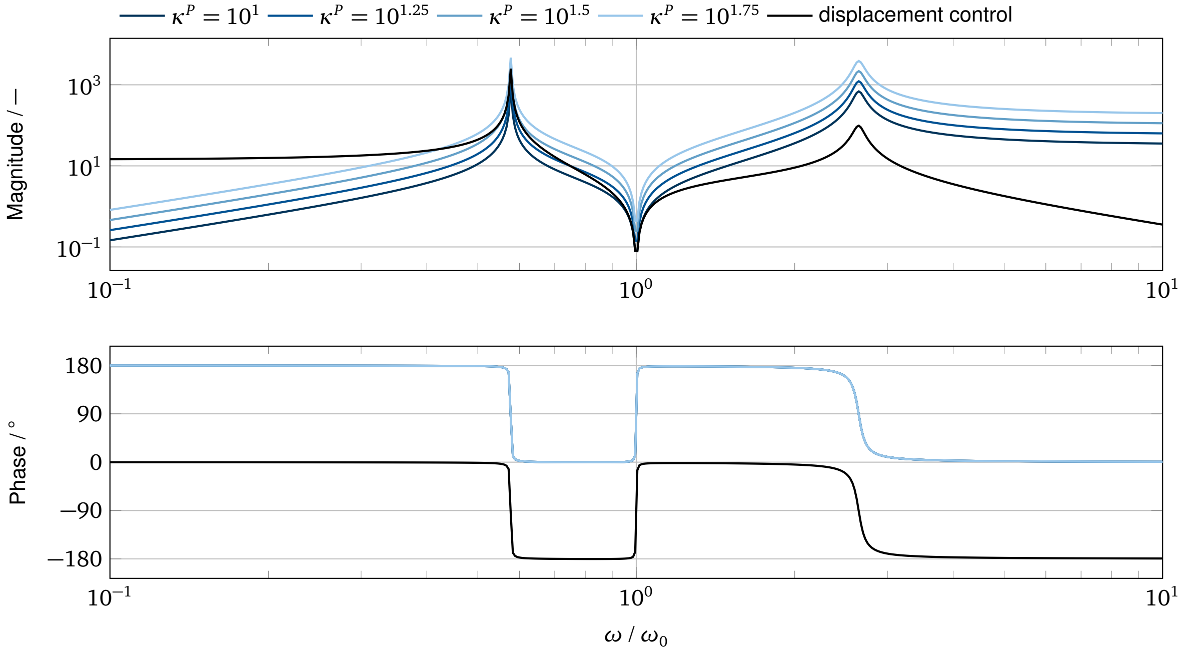


Figure 4.23: Acceleration control: Bode plots of the open-loop transfer function $H_{OL}(\omega)$ for different control gain ratios κ^P (using $\zeta^{VIR} = 2 \cdot 10^{-3}$, $\zeta^{ACT} = 0.1$, $\kappa^{EXP} = 2$, $\kappa^{ACT} = 0.1$ and $\phi^{ACT} = 0.1$)

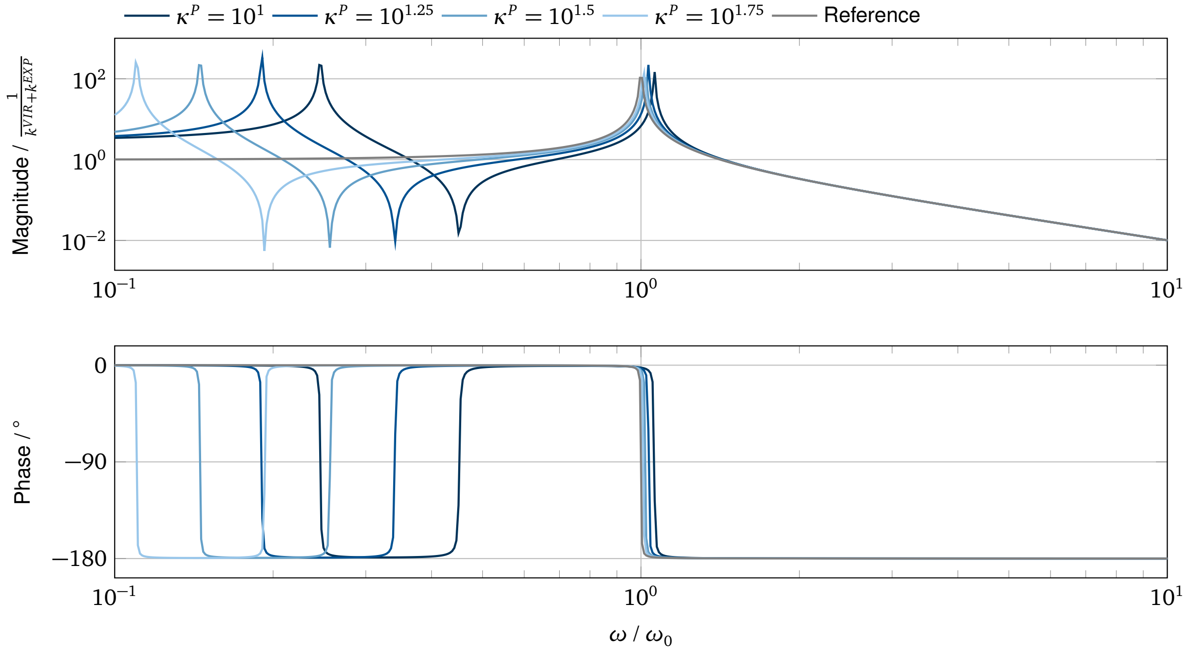


Figure 4.24: Acceleration control: Bode plots of the closed-loop transfer function $H_{CL}(\omega)$ for different control gain ratios κ^P(using $\zeta^{VIR} = 2 \cdot 10^{-3}$, $\zeta^{ACT} = 0.1$, $\kappa^{EXP} = 2$, $\kappa^{ACT} = 0.1$ and $\phi^{ACT} = 0.1$)

- Damping, stiffness and mass in the actuator deteriorate the stability properties of a test.
- Acceleration control causes drift effects and makes sense if coupling at high frequencies is desired.

A generalization to more complex systems is not straightforward. Nevertheless, the results can be used for a rough categorization of a test in the early design phase. Complex systems have to be evaluated based on their specific dynamics.

The analysis in this section was constrained to systems with solely a proportional gain. The performance can be improved by adding integral and derivative terms to the controller. Based on additional system knowledge, even more sophisticated control laws are possible. However, the design of the control using accurate system knowledge contradicts the objective of hybrid tests that is being able to test structures with unknown dynamics. Moreover, the performance of a specifically designed controller with a high bandwidth may be deteriorated by modeling inaccuracies.

A solution to the problem is the use of adaptive control strategies as they are discussed in Sec. 4.3, Sec. 4.4 and Chap. 5. The use of feedforward controllers—as proposed in Chap. 5—can help to emphasize robustness and stability in the design of the feedback controller.

4.3 Polynomial forward prediction

A method that is frequently used for compensating the actuator delay in real-time hybrid testing is polynomial forward prediction. The following section gives a short overview of the method. [45] has first proposed a polynomial forward prediction procedure for real-time hybrid testing. [109] describes the general form: The basic idea is to compensate for the delay of the actuator system by generating an actuator input u based on the current and previous values of the demand signal from the virtual component y^{VIR}. A block diagram of this approach is given in Fig. 4.25. For predicting future values of the demand signal y^{VIR}, an Nth-order polynomial

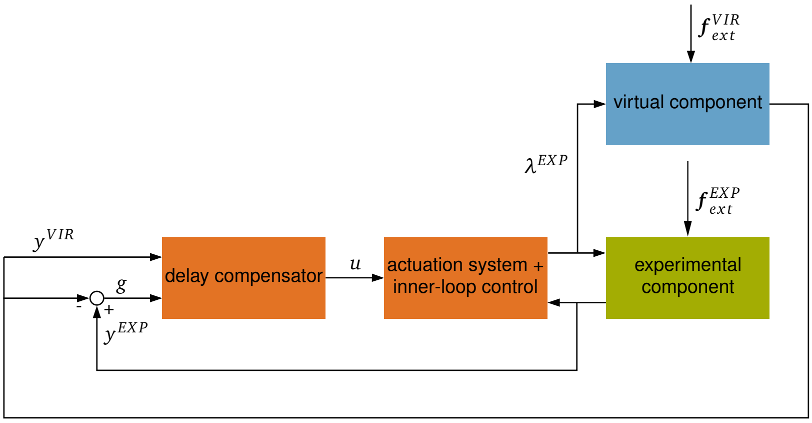


Figure 4.25: Block diagram of coupled system

$$y^{VIR}(n\Delta t + \tau) = a_0 + a_1\tau + \ldots + a_N\tau^N \tag{4.42}$$

is used, where τ is the deviation from the time at the current time step n and Δt is the time step width. The coefficients a_i with $i \in [0, 1, \ldots, N]$ can be obtained using y^{VIR} from M previous time steps. The

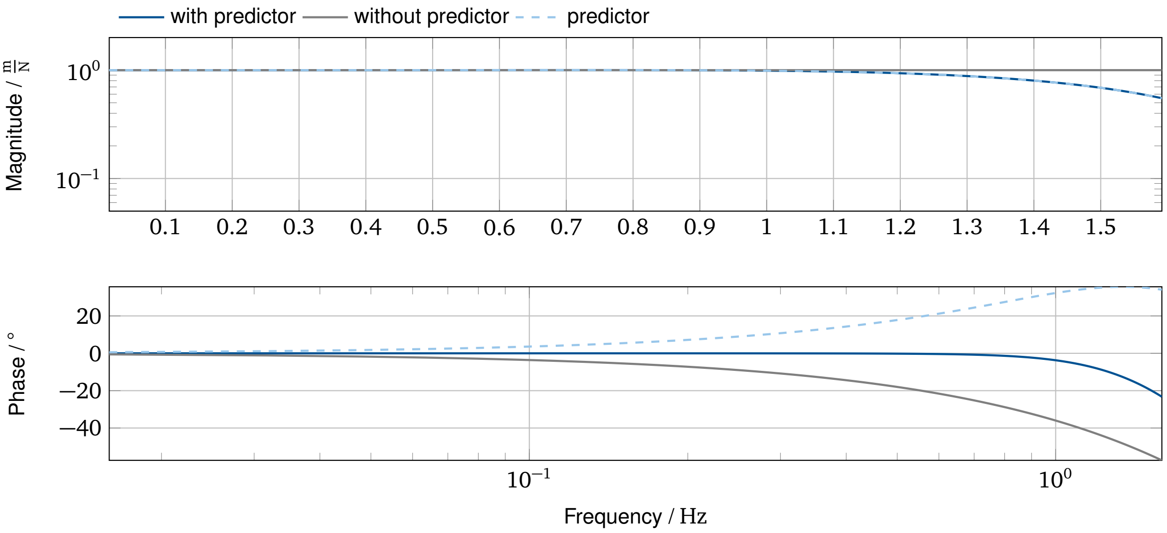


Figure 4.26: Exemplary Bode diagram of a pure delay and its compensation using polynomial forward prediction. A polynomial of grade $N = 5$ and a delay of $T_d = 0.1$ s used for the plot.

corresponding polynomials in matrix-vector notation read

$$\underbrace{\begin{bmatrix} y^{VIR}[n] \\ y^{VIR}[n-1] \\ \vdots \\ y^{VIR}[n-M] \end{bmatrix}}_{\boldsymbol{b}} = \underbrace{\begin{bmatrix} 1 & 0 & \dots & 0 \\ 1 & -\Delta t & \dots & (\Delta t)^N \\ \vdots & \vdots & \ddots & \vdots \\ 1 & -M\Delta t & \dots & (-M\Delta t)^N \end{bmatrix}}_{\boldsymbol{X}} \underbrace{\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_M \end{bmatrix}}_{\boldsymbol{a}}. \tag{4.43}$$

Consequently, the coefficient vector $\boldsymbol{a}$ can be estimated using a least-squares fit:

$$\boldsymbol{a} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{b} \tag{4.44}$$

At time $\tau = P_{AFP}\Delta t$, the extrapolated value of y^{VIR} is

$$y^{VIR}(n\Delta t + P_{AFP}\Delta t) = \underbrace{\begin{bmatrix} 1 & P_{AFP}\Delta t & \dots & (P_{AFP}\Delta t)^N \end{bmatrix}}_{\boldsymbol{X}_P} \boldsymbol{a} \tag{4.45}$$

P_{AFP} defines the length of the prediction horizon. The extrapolation can be used to generate an actuator input $u[n]$ which compensates for the actuator delay. Using the Moore–Penrose pseudo-inverse from (4.44), the actuator input reads

$$u[n] = c_{AFP}\, y^{VIR}(n\Delta t + P_{AFP}\Delta t) = c_{AFP}\boldsymbol{X}_P(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{b}, \tag{4.46}$$

where the factor c_{AFP} is used to compensate for errors in the magnitude of the oscillation. The expression $\boldsymbol{X}_P(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T$ is a vector of length $N+1$, which depends only on the prediction horizon P_{AFP} and the polynomial of order N. E.g., for the polynomial order $N = 2$, $M = 2$ and the prediction horizon $P_{AFP} = 1$ the actuator input reads

$$u[n] = c_{AFP}(3y^{VIR}[n] - 3y^{VIR}[n-1] + y^{VIR}[n-2]), \tag{4.47}$$

where c_{AFP} is a parameter which can be used for compensating error in the signal magnitude. The characteristics of polynomial forward prediction correspond to a lead compensator. This fact can be observed in the Bode diagram of Fig. 4.26.

The delay in a test can vary because it depends on test conditions and test dynamics. This fact makes

trigger condition	slope condition	update
$y^{VIR} = 0$	$\dot{y}^{VIR} > 0$	$\rho_{AFP} = \rho_{AFP} + \alpha_{AFP} g^{\gamma_{AFP}}$
$y^{VIR} = 0$	$\dot{y}^{VIR} > 0$	$\rho_{AFP} = \rho_{AFP} - \alpha_{AFP} g^{\gamma_{AFP}}$
$\dot{y}^{VIR} = 0$	$\ddot{y}^{VIR} > 0$	$\sigma_{AFP} = \sigma_{AFP} + \beta_{AFP} g^{\gamma_{AFP}}$
$\dot{y}^{VIR} = 0$	$\ddot{y}^{VIR} < 0$	$\sigma_{AFP} = \sigma_{AFP} - \beta_{AFP} g^{\gamma_{AFP}}$

Table 4.1: Triggers and update laws for adaptive polynomial forward prediction.

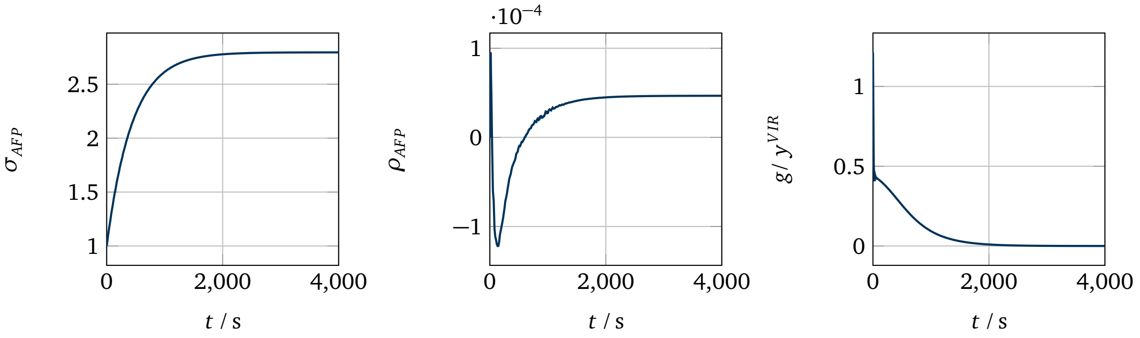


Figure 4.27: Exemplary adaptation process shown with the help of σ_{AFP}, ρ_{AFP} and the envelope of the interface gap g: The open loop system corresponds to the system from Sec. 4.2.3 using the parameters $\kappa^{EXP} = 2$, $\kappa^{ACT} = 0.1$, $\kappa^{P} = 10$, $\phi^{ACT} = 0.1$, $\zeta^{VIR} = 0.2$ and $\zeta^{VIR} = 0.1$ with an additional delay of 0.01 s

an on-line delay estimation algorithm necessary. [29] proposes a technique, which uses a proportional feedback system for delay estimation. [109] introduces an on-line delay estimation technique based on zero crossings of the signal: The length of the prediction horizon P_{AFP} as well as the magnitude compensation parameter c_{AFP} consist of a constant part ($P_{0,AFP}$ and $c_{0,AFP}$) and a variable part (ρ_{AFP} and σ_{AFP}):

$$P_{AFP} = P_{0,AFP} + \rho_{AFP} \tag{4.48}$$

$$c_{AFP} = c_{0,AFP} + \sigma_{AFP} \tag{4.49}$$

Updates of the variable parts are triggered by sign changes of y^{VIR} and $\dot{y}^{VIR}$ according to Tab. 4.1. An exemplary adaptation process is shown in Fig. 4.27.

4.4 Passivity Based Control

Another method which enables robust and stable coupling of subcomponents in real-time hybrid tests is proposed in [78]. The authors apply a passivity-based control scheme in order to improve the stability and accuracy of real-time hybrid tests.

In control theory, a system is called passive if it does not generate energy. The use of the passivity property is appealing in the context of complex control systems due to two reasons according to [40]: Firstly, all passive systems are stable. Hence, the prove of passivity is a sufficient condition for the proof of stability. Secondly, the passivity of a system which is built up from various components can be assessed based on the passivity of the single components. If each component of the system is passive, the overall system behaves passively. This also applies to non-linear systems. In the case of real-time hybrid testing, it is necessary that the actuator system behaves passively, i.e. it does not generate energy. The unstable behavior of a real-time hybrid test can be seen as an energy injection to

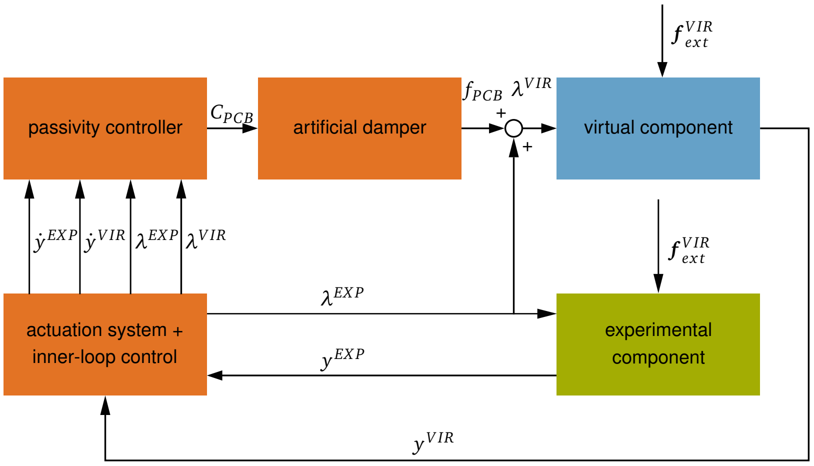


Figure 4.28: Block diagram of a real-time hybrid test with a passivity based controller (according to [78]).

the system, which is caused by the actuator and control system.

Passivity-based control approaches have been successfully used in teleoperation, robotic impedance control and for haptic interfaces (see e.g. [71], [1] and [40]). As mentioned above, one way to apply a passivity-based control scheme to real-time hybrid testing is suggested by [78]. A dissipative element with a variable viscous damping rate, which depends on the net power-outflow from the actuation system, is introduced. In this way, additional damping is injected at the interface in order to restore passivity of an initially non-passive actuator system. A block diagram of the control scheme is shown in Fig. 4.28. In the depicted structure an actuation system with inner-loop control serves as the basis for the real-time-hybrid test. The passivity-based control scheme is augmented to this structure.

The passivity based control scheme works as follows: The net power-outflow from the actuation system is a measure for the passivity violation caused by the actuation system. For a real-time hybrid test with interface forces λ^{EXP} and λ^{VIR}, and interface velocities $\dot{y}^{EXP}$ and $\dot{y}^{VIR}$ it reads

$$P_{out}^{ACT} = \lambda^{EXP}\dot{y}^{EXP} - \lambda^{VIR}\dot{y}^{VIR} \tag{4.50}$$

Normalization of the power-outflow makes tuning of the passivity based control independent of the magnitude of the excitation forces on the system. The normalization is performed using the total power flow P_{tot} as a reference. It is defined as

$$P_{tot} = \lambda^{EXP}\dot{y}^{EXP} + \lambda^{VIR}\dot{y}^{VIR}. \tag{4.51}$$

The damping coefficient C_{PCB} is a function of the power-outflow from the actuation system. In order to prevent nonlinear distortions, it is desirable to reduce fast changes in C_{PCB}. This is achieved using first-order low pass filters. Low pass filtering of P_{out}^{ACT} and P_{tot} yields the filtered power signals $\tilde{P}_{out}^{ACT}$ and $\tilde{P}_{tot}$. Design rules for the choice of the cut-off frequencies of the low pass filters are given in [78]. The variable damping coefficient is proportional to the normalized net power-outflow according to Eq. (4.52). Negative power-outflows do not require additional damping and the damping constant is set to zero:

$$C_{PCB} = \begin{cases} 0 & \text{if } \tilde{P}_{out}^{ACT}/|\tilde{P}_{tot}| \leq 0 \\ G_{PCB}\frac{\tilde{P}_{out}^{ACT}}{|\tilde{P}_{tot}|} & \text{if } \tilde{P}_{out}^{ACT}/|\tilde{P}_{tot}| > 0 \end{cases} \tag{4.52}$$

G_{PCB} is a gain which is used to tune the performance of the passivity based controller. The additional damping force f_{PCB} at the interface is calculated according to Eq. (4.53):

$$f_{PCB} = C_{PCB}\dot{y}^{VIR} \tag{4.53}$$

Finally, the interface force with augmented damping forces reads

$$\lambda^{VIR} = \lambda^{EXP} + f_{PCB}. \tag{4.54}$$

The augmentation of a damping force to the interface forces can lead to a deteriorated test performance since the equilibrium constraint is not met exactly. However, the control law is not active in the case of a passive actuation system and prevents potentially damaging behavior in the case of a passivity violation while continuing to run the test.

Chapter 5

Feedforward Based Coupling

Parts of this chapter have been published in [3], and submitted for publication in [7] and in [9].

As discussed in Chap. 4, feedback loops in combination with actuator dynamics and delays can cause an inaccurate and unstable behavior of real-time hybrid tests. If the exact model of the full system is known, it is possible to use feedforward filters to couple the subcomponents. The corresponding control structure is given in Fig. 5.1. It shows a block diagram of a hybrid test with pure feedforward control. It is a simplified version of the general control scheme from Fig. 4.3. According to the scheme, the external excitations serve as inputs to the feedforward filter. The feedforward filter generates an actuator input which enforces compatibility between the interfaces. In other words, it closes the interface gap. Equilibrium is enforced by applying the interface forces directly on the virtual component. The feedforward filter can be directly calculated using the compatibility constraint $\boldsymbol{g} = \boldsymbol{0}$. In Laplace domain, after substituting the transfer functions defined in Chap. 2, the constraint reads:

$$\begin{aligned} \boldsymbol{G}(s) =& \boldsymbol{Y}^{EXP}(s) - \boldsymbol{Y}^{VIR}(s) \\ =& \underbrace{\left(\boldsymbol{H}^{TR}_{y,u}(s) + \boldsymbol{H}^{VIR}_{y,\lambda}(s)\boldsymbol{H}^{TR}_{\lambda,u}(s)\right)}_{\boldsymbol{H}_{g,u}(s)} \boldsymbol{U}(s) + \boldsymbol{H}^{VIR}_{ext}(s)\boldsymbol{F}^{VIR}_{g,ext}(s) + \boldsymbol{H}^{EXP}_{ext}(s)\boldsymbol{F}^{EXP}_{g,ext}(s) \overset{!}{=} \boldsymbol{0} \end{aligned} \tag{5.1}$$

Remember that the superscript TR denotes the assembly of actuator system and experimental component. $\boldsymbol{H}_{g,u}(\omega)$ is the transfer function between actuator input and interface gap. $\boldsymbol{H}^{*}_{g,ext}(\omega)$ is the transfer function between external excitation and interface gap. Solving Eq. (5.1) for the actuator input yields

$$\boldsymbol{U}(s) = \underbrace{-\boldsymbol{H}_{g,u}(s)^{-1}\boldsymbol{H}^{VIR}_{g,ext}(s)}_{\boldsymbol{H}^{VIR}_{FF}(s)} \boldsymbol{F}^{VIR}_{ext}(s) \underbrace{-\boldsymbol{H}_{g,u}(s)^{-1}\boldsymbol{H}^{EXP}_{g,ext}(s)}_{\boldsymbol{H}^{EXP}_{FF}(s)} \boldsymbol{F}^{EXP}_{ext}(s), \tag{5.2}$$

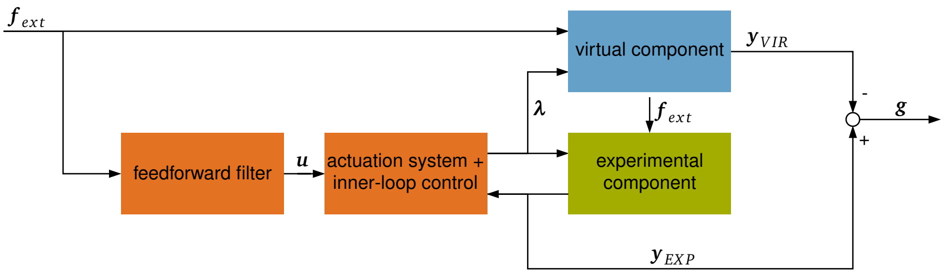


Figure 5.1: Block diagram of a purely feedforward controlled hybrid test. For readability reasons and without loss of generality the excitation on the experimental component is omitted in the diagram.

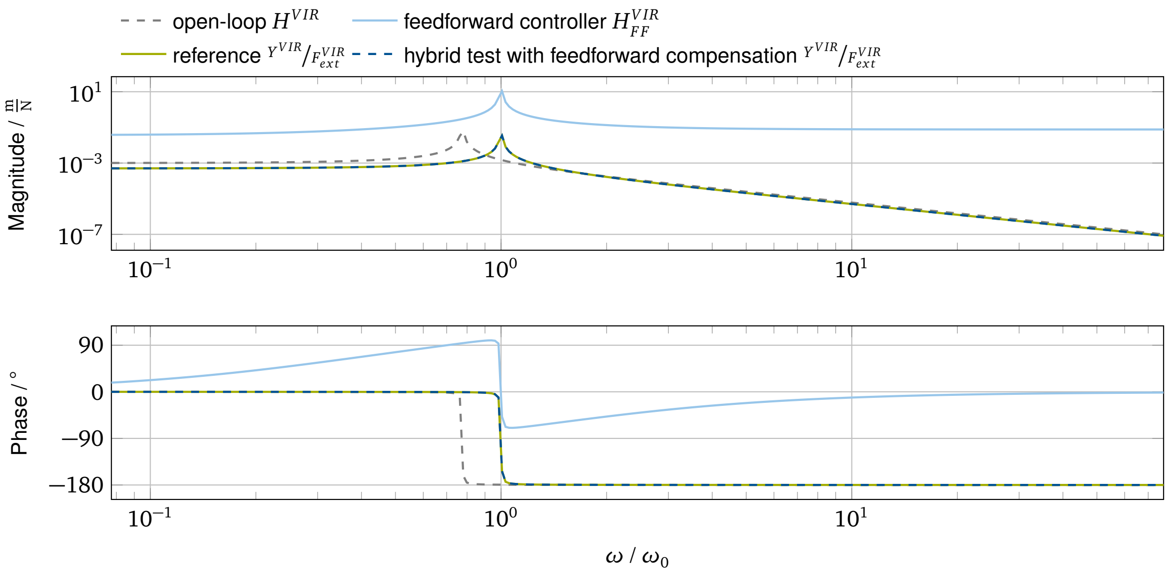


Figure 5.2: Bode diagrams of various transfer functions related to a real-time hybrid test using a pure feedforward filter.

where $\boldsymbol{H}_{FF}^{VIR}(s)$ and $\boldsymbol{H}_{FF}^{EXP}(s)$ are the transfer functions of the feedforward filters which enforce compatibility of the interface displacements with the given system.

The fact that this system correctly emulates the reference dynamics can be observed using the transfer functions from Sec. 4.2:

$$\boldsymbol{H}_{y,u}^{TR}(s) = H^{TR}(s)H^{EXP}(s) \tag{5.3}$$

$$\boldsymbol{H}_{y,\lambda}^{VIR}(s) = H^{TR}(s) \tag{5.4}$$

$$\boldsymbol{H}_{y,\lambda}^{VIR}(s) = \boldsymbol{H}_{ext}^{VIR}(s) = H^{VIR}(s) \tag{5.5}$$

$$\boldsymbol{F}_{ext}^{EXP}(s) = 0 \tag{5.6}$$

The displacement output at the interface of the virtual component in Laplace domain reads

$$\boldsymbol{Y}^{VIR}(s) = \boldsymbol{H}_{\lambda}^{VIR}(s)\boldsymbol{H}_{\lambda,u}^{TR}(s)\boldsymbol{U}(s) + \boldsymbol{H}_{ext}^{VIR}(s)\boldsymbol{F}_{ext}^{VIR}(s). \tag{5.7}$$

Substitution yields the transfer function from external excitations to interface displacements:

$$\frac{Y^{VIR}(s)}{F_{ext}(s)} = -\frac{H^{VIR}(s)H^{VIR}(s)H^{TR}(s)}{H^{TR}(s)H^{VIR}(s) + H^{TR}(s)H^{EXP}(s)} + H^{VIR}(s) \tag{5.8}$$

$$= \frac{H^{VIR}(s)H^{EXP}(s)}{H^{VIR}(s) + H^{EXP}(s)} = H^{REF}(s) \tag{5.9}$$

This type of hybrid test with a pure feedforward filter exactly emulates the reference system. The corresponding transfer functions are shown in the plot of Fig. 5.2. The control structure shown above includes no feedback loop. Hence the system is inherently stable. It is noteworthy that the pole of the feedforward filter is placed at the frequency of its counterpart from the coupled system. Hence, knowledge of the full system is necessary in order to design the filter. The knowledge of the experimental component, however, is not exact—if it is present at all—according to the paradigm of hybrid testing. Furthermore, if the feedforward filter is not exactly appropriate for the given system, the test results will be highly deteriorated.

Adaptive feedforward filters can overcome these shortcomings. They are computed during the test and continuously adapting to new situations or slight changes in the system dynamics. The corresponding

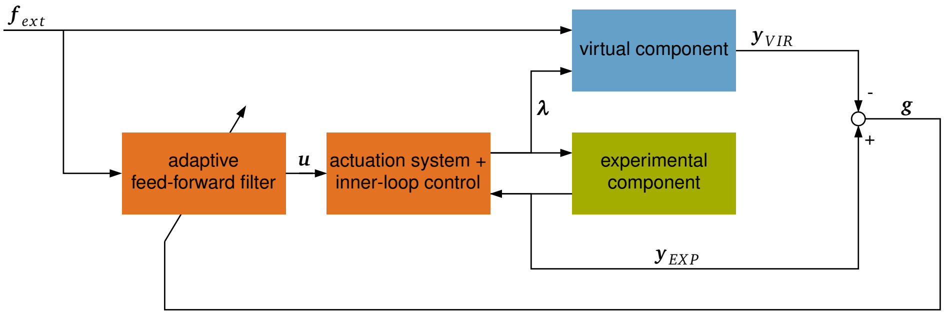


Figure 5.3: Block diagram of hybrid test with adaptive feedforward filter. For readability reasons and without loss of generality the excitation on the experimental component is omitted in the diagram.

structure of the control system is shown in Fig. 5.3. Adaptive feedforward filters have been used for testing applications in the well-established Time Waveform Replication (TWR) method. As described in [27] and [32], the technique allows replication of reference signals in vibration tests, e.g. road simulations. In the TWR method, the actuators usually work under real-time control while the feedforward filters generate the input signal. In an outer loop, the feedforward 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 industrial applications. [20] proposes the use of adaptive feedforward 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 [53]. The interface synchronization problem can be formulated as a disturbance rejection problem. This structure is used by [61] with Neural Networks as feedforward filters. A similar approach was investigated in the student thesis of [46].

In this chapter, methods are discussed which allow applying adaptive feedforward filters to real-time hybrid testing. Different types of adaptive feedforward filters are analyzed. Sec. 5.1 describes harmonic basis functions which are helpful to build efficient adaptive filters. Two types of adaptation laws are shown in Sec. 5.2 and Sec. 5.4 while Sec. 5.3 addresses the automated choice of adaptation parameters. Sec. 5.5 considers adaptive algorithms based on finite impulse response filters. Finally, the combination of adaptive feedforward filters with feedback based coupling is introduced in Sec. 5.6.

5.1 Harmonic Basis Functions

The objective of this section is to reformulate the compatibility constraint such that a feedforward 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 adaptation 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 $\mathbf{0}$.

$$\boldsymbol{G}(\omega) = \boldsymbol{H}_{g,u}(\omega)\boldsymbol{U}(\omega) + \boldsymbol{H}_{g,ext}(\omega)\boldsymbol{F}_{ext}(\omega) \overset{!}{=} \mathbf{0} \tag{5.10}$$

An example of an excitation force vector with n_Ω harmonics and n_{int} interface DOFs is given in Eq. (5.11).

$$\boldsymbol{f}_{ext}(t) = \mathbf{1}\sum_{i=1}^{n_\Omega}\cos(\Omega_i t) \tag{5.11}$$

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

$$\mathcal{F}\left(\boldsymbol{f}_{ext}\right) = \boldsymbol{F}_{ext}(\omega) = \mathbf{1}\sqrt{2\pi}\sum_{i=1}^{n_\Omega}\left(\frac{\delta(\omega-\Omega_i)}{2} + \frac{\delta(\omega+\Omega_i)}{2}\right).$$

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. Rearranging Eq. (5.10) yields the actuator input in frequency domain which is given in Eq. (5.12).

$$\boldsymbol{U}(\omega) = -\boldsymbol{H}_{g,u}(\omega)^{-1}\boldsymbol{H}_{g,ext}(\omega)\boldsymbol{F}_{ext}(\omega) = \sqrt{2\pi}\sum_{i=1}^{n_\Omega}\boldsymbol{\psi}(\Omega_i)\left(\frac{\delta(\omega-\Omega_i)}{2} + \frac{\delta(\omega+\Omega_i)}{2}\right) \tag{5.12}$$

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 DOFs is equal to the number of actuator DOFs. Transforming $\boldsymbol{U}(\omega)$ back into the time domain and performing some elementary complex number operations—using the complex conjugate $\bar{\boldsymbol{\psi}}(\Omega_i)$ of $\boldsymbol{\psi}(\Omega_i)$—yields the matrix-vector form of Eq. (5.13).

$$\begin{aligned}
\boldsymbol{u}(t) &= \mathcal{F}^{-1}(\boldsymbol{U}(\omega)) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\boldsymbol{U}(\omega)\mathrm{e}^{j\omega t}\mathrm{d}\omega = \frac{1}{2}\sum_{i=1}^{n_\Omega}\left(\bar{\boldsymbol{\psi}}(\Omega_i)\mathrm{e}^{-j\Omega_i t} + \boldsymbol{\psi}(\Omega_i)\mathrm{e}^{j\Omega_i t}\right)\\
&= \sum_{i=1}^{n_\Omega}\mathrm{Re}\left(\boldsymbol{\psi}(\Omega_i)\mathrm{e}^{j\Omega_i t}\right) = \sum_{i=1}^{n_\Omega}\mathrm{Re}\left(\mathrm{e}^{j\Omega_i t}\right)\mathrm{Re}\left(\boldsymbol{\psi}(\Omega_i)\right) - \mathrm{Im}\left(\mathrm{e}^{j\Omega_i t}\right)\mathrm{Im}\left(\boldsymbol{\psi}(\Omega_i)\right)\\
&= \underbrace{\begin{bmatrix} \boldsymbol{I}\cos(\Omega_1 t)\\ -\boldsymbol{I}\sin(\Omega_1 t)\\ \vdots\\ \boldsymbol{I}\cos\left(\Omega_{n_\Omega} t\right)\\ -\boldsymbol{I}\sin\left(\Omega_{n_\Omega} t\right)\end{bmatrix}^T}_{\boldsymbol{W}(t)}\underbrace{\begin{bmatrix}\mathrm{Re}(\boldsymbol{\psi}(\Omega_1))\\ \mathrm{Im}(\boldsymbol{\psi}(\Omega_1))\\ \vdots\\ \mathrm{Re}\left(\boldsymbol{\psi}(\Omega_{n_\Omega})\right)\\ \mathrm{Im}\left(\boldsymbol{\psi}(\Omega_{n_\Omega})\right)\end{bmatrix}}_{\boldsymbol{\theta}} = \boldsymbol{W}(t)\boldsymbol{\theta}
\end{aligned} \tag{5.13}$$

The time-dependent harmonic basis functions can be written in a compact way by 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 complex number operations from (5.13), the interface gap $\boldsymbol{g}(t)$ is rearranged in matrix-vector form as can be seen in Eq. (5.14).

$$\begin{aligned}
\boldsymbol{g}(t) =& \frac{1}{2\pi}\int_{-\infty}^{\infty}\boldsymbol{H}_{g,u}(\omega)\boldsymbol{U}(\omega)\mathrm{e}^{j\omega t}\mathrm{d}\omega + \frac{1}{2\pi}\int_{-\infty}^{\infty}\boldsymbol{H}_{g,ext}(\omega)\boldsymbol{F}_{ext}(\omega)\mathrm{e}^{j\omega t}\mathrm{d}\omega\\
=&\boldsymbol{W}(t)\begin{bmatrix}\mathrm{Re}\left(\boldsymbol{H}_{g,u}(\Omega_1)\boldsymbol{\psi}(\Omega_1)\right)\\ \mathrm{Im}\left(\boldsymbol{H}_{g,u}(\Omega_1)\boldsymbol{\psi}(\Omega_1)\right)\\ \vdots\\ \mathrm{Re}\left(\boldsymbol{H}_{g,u}(\Omega_{n_\Omega})\boldsymbol{\psi}(\Omega_{n_\Omega})\right)\\ \mathrm{Im}\left(\boldsymbol{H}_{g,u}(\Omega_{n_\Omega})\boldsymbol{\psi}(\Omega_{n_\Omega})\right)\end{bmatrix} + \boldsymbol{g}_{ext}\\
=&\boldsymbol{W}(t)\boldsymbol{P}_{g,u}\boldsymbol{\theta} + \boldsymbol{g}_{ext}
\end{aligned}$$

with

$$\boldsymbol{P}_{g,u} = \begin{bmatrix} \mathrm{Re}\big(\boldsymbol{H}_{g,u}(\Omega_1)\big) & -\mathrm{Im}\big(\boldsymbol{H}_{g,u}(\Omega_1)\big) & & & \\ \mathrm{Im}\big(\boldsymbol{H}_{g,u}(\Omega_1)\big) & \mathrm{Re}\big(\boldsymbol{H}_{g,u}(\Omega_1)\big) & & & \\ & & \ddots & & \\ & & & \mathrm{Re}\big(\boldsymbol{H}_{g,u}(\Omega_{n_\Omega})\big) & -\mathrm{Im}\big(\boldsymbol{H}_{g,u}(\Omega_{n_\Omega})\big) \\ & & & \mathrm{Im}\big(\boldsymbol{H}_{g,u}(\Omega_{n_\Omega})\big) & \mathrm{Re}\big(\boldsymbol{H}_{g,u}(\Omega_{n_\Omega})\big) \end{bmatrix} \tag{5.14}$$

The interface gap $\boldsymbol{g}(t)$ can be 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. The symbol $\boldsymbol{g}_{ext}(t)$ represents the contribution of the excitation forces to the interface gap. This form can be used to derive the adaptation law for $\boldsymbol{\theta}$. A perfect $\boldsymbol{\theta}$ would yield an interface gap $\boldsymbol{g}$ that is zero. The matrix-vector formulation of Sec. 5.1 serves as a framework and allows various adaptation laws to be applied. Two adaptation laws which proved to work in the context of coupling structures are applied in Sec. 5.2 and Sec. 5.4: a Least-Mean-Squares adaptation law, which is based on stochastic gradient descent and a Recursive-Least-Squares adaptation law, which is based on a deterministic recursive formulation.
Substituting time step size Δt and time step number k into Eq. (5.14) yields the time discrete form given in Eq. (5.15).

$$\boldsymbol{g}[k] = \boldsymbol{W}[k]\boldsymbol{P}_{g,u}\boldsymbol{\theta} + \boldsymbol{g}_{ext}[k] \tag{5.15}$$

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

5.2 Least-Mean-Squares Based Adaptation Law

Least-Mean-Squares-type algorithms have been successfully applied to active noise cancellation and disturbance rejection (see e.g. [57]). The properties of these algorithms include simple implementation and low computational costs. The coupling problem can be formulated in the form of a disturbance rejection problem (Fig. 5.4). Hence, it is possible to apply LMS algorithms to hybrid testing. The adaptation law proposed in the following section can be seen as a narrow-band version of the fx-LMS algorithm (see e.g. [49]).

5.2.1 Adaption

The objective of the adaptation 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 adaptation law makes use of the cost function of Eq. (5.16), which is the expected value of the squared interface gap $\boldsymbol{g}$.

$$\begin{aligned} J[k] &= \mathrm{E}\left\{\boldsymbol{g}^T[k]\boldsymbol{g}[k] + \gamma_{LMS}\boldsymbol{\theta}[k]^T\boldsymbol{\theta}[k]\right\} \\ &= \mathrm{E}\left\{\left(\boldsymbol{W}[k]\boldsymbol{P}_{g,u}[k]\boldsymbol{\theta}[k] + \boldsymbol{g}_{ext}[k]\right)^T\left(\boldsymbol{W}[k]\boldsymbol{P}_{g,u}[k]\boldsymbol{\theta}[k] + \boldsymbol{g}_{ext}[k]\right) + \gamma_{LMS}\boldsymbol{\theta}[k]^T\boldsymbol{\theta}[k]\right\} \end{aligned} \tag{5.16}$$

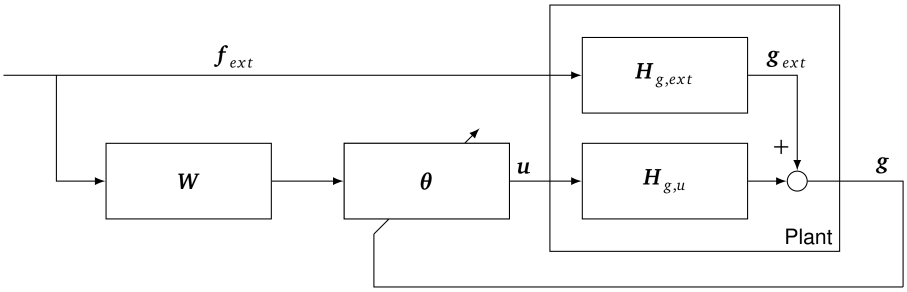


Figure 5.4: Block diagram for adapting θ: The system dynamics seen in Fig. 5.1 can be rearranged as a superposition of interface force contribution and external force contribution. Interface synchronization is realized via an LMS-type adaptive feedforward filter.

γ_{LMS} is the regularization factor. 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. (5.17) contains the gradient of the given cost function with respect to parameter vector $\boldsymbol{\theta}$.

$$\nabla J[k] = \frac{\partial J[k]}{\partial \boldsymbol{\theta}[k]} = 2\mathrm{E}\left\{\boldsymbol{P}_{g,u}[k]^T \boldsymbol{W}[k]^T \boldsymbol{g}[k] + \gamma_{LMS}\boldsymbol{\theta}[k]\right\} \tag{5.17}$$

This expression for the gradient still depends on the expectation function, which cannot be calculated directly. For the on-line adaptation, the expectation function is approximated using the last sample:

$$\mathrm{E}\left\{\boldsymbol{P}_{g,u}[k]^T \boldsymbol{W}[k]^T \boldsymbol{g}[k] + \gamma_{LMS}\boldsymbol{\theta}[k]\right\} \simeq \boldsymbol{P}^{est}_{g,u}[k]^T \boldsymbol{W}[k]^T \boldsymbol{g}[k] + \gamma_{LMS}\boldsymbol{\theta}[k]$$

The adaptation law (5.19) ultimately results from the gradient of the cost function.

$$\boldsymbol{\theta}[k+1] = \boldsymbol{\theta}[k] - \bar{\mu}_{LMS}\nabla J[k] = \nu_{LMS}\boldsymbol{\theta}[k] - \bar{\mu}_{LMS}\boldsymbol{P}^{est}_{g,u}[k]^T \boldsymbol{W}[k]^T \boldsymbol{g}[k] \tag{5.18}$$

$$\text{with} \qquad \nu_{LMS} = 1 - \bar{\mu}_{LMS}\gamma_{LMS} \tag{5.19}$$

The adaptation gain $\bar{\mu}_{LMS}$ defines the step size of the gradient descent. The leakage factor ν_{LMS} originates from the regularization term in the objective function. Note that instead of the true interface transfer matrix $\boldsymbol{P}_{g,u}$, the estimated matrix $\boldsymbol{P}^{est}_{g,u}$ has to be used. Sec. 5.2.2 shows an identification procedure to build $\boldsymbol{P}^{est}_{g,u}$. In order to make the choice of the adaptation gain more practicable, a normalized adaptation gain μ_{LMS} according to Eq. (5.20) is used.

$$\bar{\mu}_{LMS} = \mu_{LMS}\frac{1}{\lambda_{max} + \gamma_{LMS}} \tag{5.20}$$

The normalization makes use of the maximum eigenvalue λ_{max} of the matrix $\boldsymbol{P}^T_{g,u}\boldsymbol{P}_{g,u}$. See [57] for a derivation of the expressions. In theory, a normalized adaptation gain of $\mu_{LMS} = 1$ results in the fastest possible convergence. Changes in the system dynamics and inaccuracy in the identification process of $\boldsymbol{P}^{est}_{g,u}$ may bring the maximum adaptation gain down to a lower value. Fig. 5.4 shows the resulting block diagram for the adaptation 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 based on the interface gap signal $\boldsymbol{g}$.

5.2.2 Identification

As shown in the previous section, it is necessary to estimate the interface transfer matrix $\boldsymbol{P}_{g,u}$ before performing the adaptation of $\boldsymbol{\theta}$. An LMS-based approach—similar to the one used for the adaptation

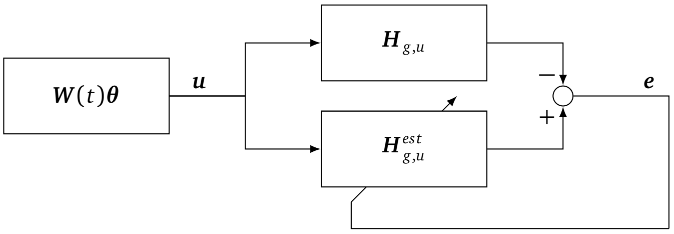


Figure 5.5: Block diagram for identifying $\boldsymbol{P}_{g,u}$: During the identification process, excitation forces $\boldsymbol{f}_{ext}$ seen in Fig. 5.4 are set to zero. The dynamics are exposed through excitations with a random vector $\boldsymbol{\theta}_i d$ at the actuator inputs.

of $\boldsymbol{\theta}$—is used to identify $\boldsymbol{P}_{g,u}$. The basis for deriving an identification algorithm is the system identification error $\boldsymbol{e}[k]$.

$$\boldsymbol{e}[k] = \boldsymbol{W}[k]\boldsymbol{P}^{est}_{g,u}[k]\boldsymbol{\theta}[k] + \boldsymbol{g}_{ext}[k] - \boldsymbol{g}[k] \tag{5.21}$$

Since $\boldsymbol{g}_{ext}[k]$ may be unknown, it is removed during the identification process by setting $\boldsymbol{f}^{VIR}_{ext}(t) = \boldsymbol{0}$ and $\boldsymbol{f}^{EXP}_{ext}(t) = \boldsymbol{0}$. To make the same derivations as in the previous section, the values in the matrix $\boldsymbol{P}^{est}_{g,u}$ have to be rearranged in a vector according to the example given in Appendix D.1. This vector is referred to as $\tilde{\boldsymbol{P}}^{est}_{g,u}$. The resulting rearranged form in Eq. (5.22) contains the matrix $\tilde{\boldsymbol{\theta}}[k]$, which is a result of rearranging the expressions from Eq. (5.14).

$$\boldsymbol{e}[k] = \boldsymbol{W}[k]\tilde{\boldsymbol{\theta}}[k]\tilde{\boldsymbol{P}}^{est}_{g,u}[k] - \boldsymbol{g}[k] \quad \text{with} \quad \tilde{\boldsymbol{\theta}} \in \mathbb{R}^{2n_{int}^2 n_\Omega \times 2n_{int}^2 n_\Omega} \quad \text{and} \quad \tilde{\boldsymbol{P}}_{g,u} \in \mathbb{R}^{2n_{int}^2 n_\Omega} \tag{5.22}$$

The cost function and the resulting adaptation law are given in Eqs. (5.23) and (5.24). The identification process can be initialized with $\tilde{\boldsymbol{P}}^{est}_{g,u} = \boldsymbol{0}$.

$$J[k] = \mathrm{E}\left\{\boldsymbol{e}^T[k]\boldsymbol{e}[k]\right\} \tag{5.23}$$

$$\tilde{\boldsymbol{P}}^{est}_{g,u}[k+1] = \tilde{\boldsymbol{P}}^{est}_{g,u}[k] + \mu_{id,LMS}\tilde{\boldsymbol{\theta}}[k]^T\boldsymbol{W}[k]^T\boldsymbol{g}[k] \tag{5.24}$$

Similar to Sec. 5.2.1, $\mu_{id,LMS}$ is the adaptation gain.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. 5.5.

5.2.3 Testing Procedure

Alg. 1 shows a pseudocode 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 $\boldsymbol{P}^{est}_{g,u}$, the vector $\boldsymbol{\theta}$ is rearranged into the matrix $\tilde{\boldsymbol{\theta}}$ according to Appendix D.1 after each change in $\boldsymbol{\theta}$. The other way round, the matrix $\boldsymbol{P}^{est}_{g,u}$, which is used during the adaptation process, is retrieved by rearrangement of the vector $\tilde{\boldsymbol{P}}^{est}_{g,u}$. After the identification process is finished the adaptation process starts. The parameter vector can be initialized with $\boldsymbol{\theta} = \boldsymbol{0}$. 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 adaptation process. The time length of the adaptation phase is defined by t_{ad}.

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**
```

5.2.4 Numerical experiment: Dynamics of the coupled system

In the following section, the algorithm described above is applied to the lumped mass system of Chap. 1. The system is harmonically excited with the frequencies $\omega_k \in [0.25\omega_0, 0.5\omega_0, \ldots, 2.5\omega_0]$. Fig. 5.6 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. The fact that the dynamics of the proposed LMS controller can be expressed as a linear system is used here. See [4] for the proof and details. The plots show that the closed-loop dynamics match with the reference dynamics in phases and amplitudes exactly at the excitation frequencies. At all other points, the closed-loop transfer function stays on the curve of the virtual component. Fig. 5.7 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 $0.1\omega_0$. The system which is 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.

5.2.5 Numerical experiment: Effects of identification errors

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, the mass m_{EXP} was modified in the numerical example. The LQR in Fig. 5.8 is tuned to the original value of m_{EXP}, while the value

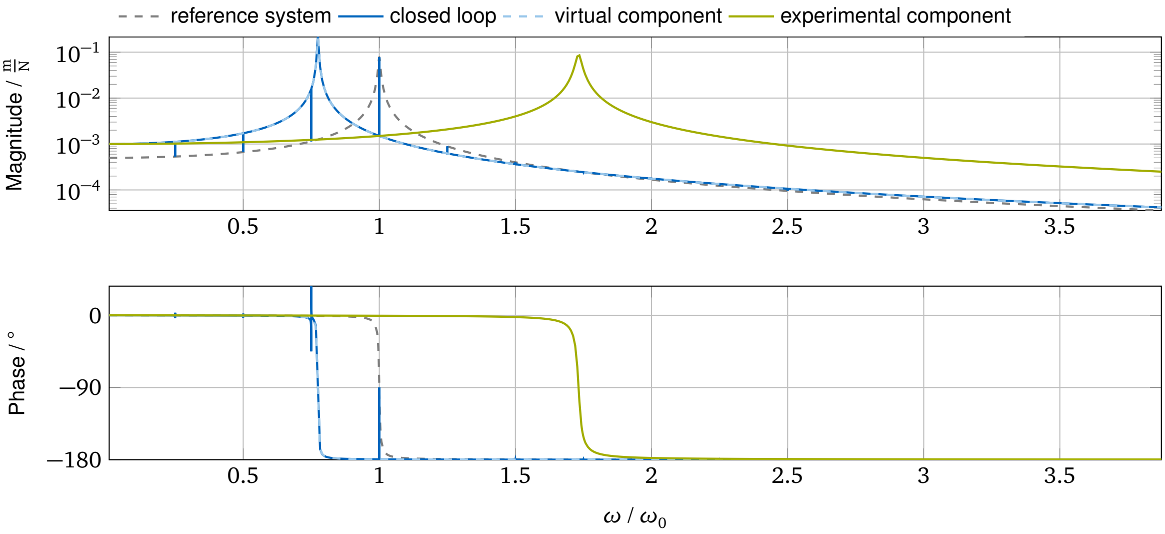


Figure 5.6: Coupling of two mass-spring-damper systems with an adaptive feedforward filter. The transfer functions shows that the steady-state closed-loop dynamics match with reference dynamics at the excitation frequencies $f_k \in [5\,\mathrm{Hz}, 10\,\mathrm{Hz}, \ldots, 50\,\mathrm{Hz}]$.

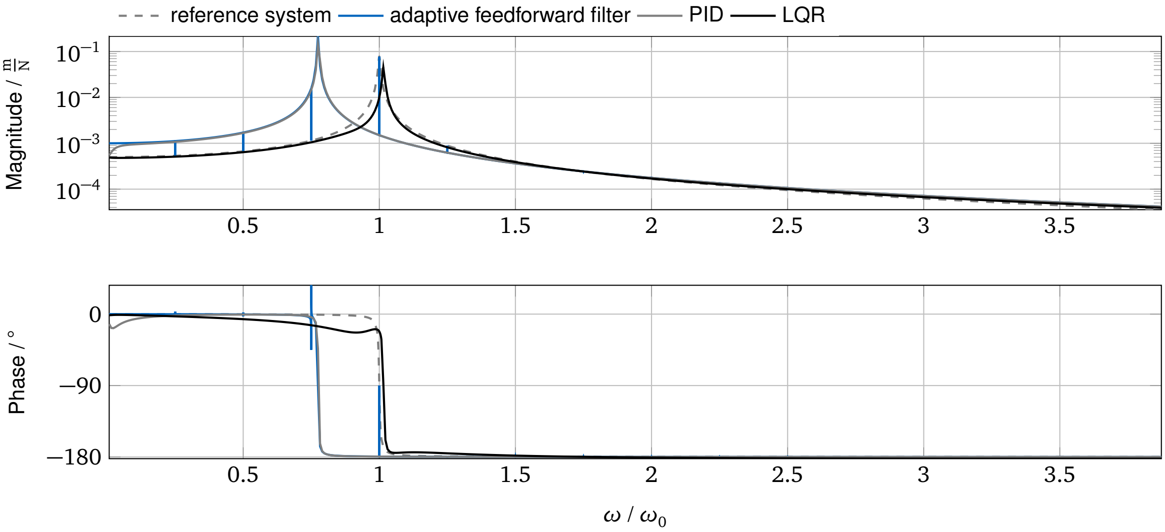


Figure 5.7: Comparing the closed-loop transfer functions for LMS filter-based algorithms with automatically generated controllers.

of m_{EXP} in the system under control is varied. More specifically, the mass of the system under test is set to values of $m^*_{EXP} \in [0.001\,\mathrm{kg}, 0.002\,\mathrm{kg}, \dots, 0.03\,\mathrm{kg}]$. 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. 5.9, the same procedure is repeated and applied to the adaptive feedforward filter. Again, the controller is tuned to the original value of m_{EXP}, while the system under control is varied. In other words, in the case of the adaptive feedforward filter, identifying $\boldsymbol{P}^{est}_{g,u}$ is performed on the system with the original mass.

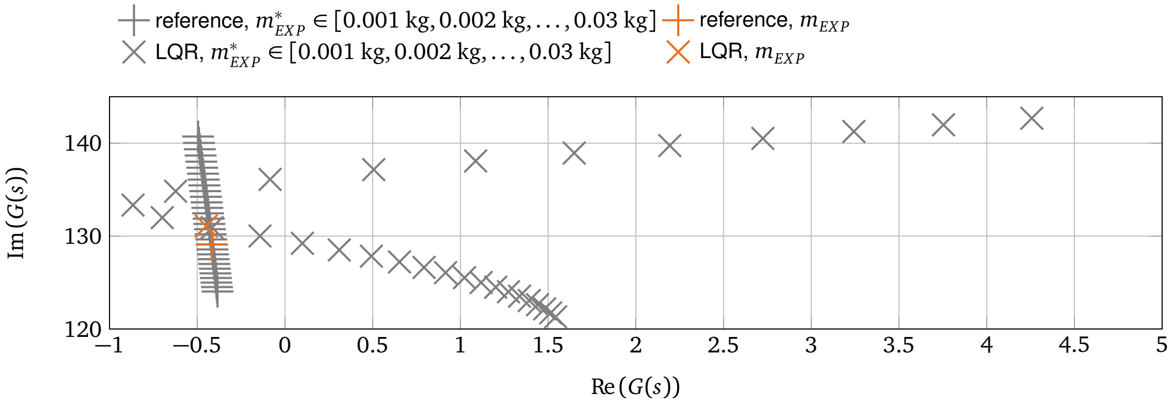


Figure 5.8: Detail from pole plot for closed-loop transfer functions $G(s)$: The LQR is tuned to the original value of m_{EXP}, whereas the value of $m^*_{EXP} \in [0.001\,\mathrm{kg}, 0.002\,\mathrm{kg}, \dots, 0.03\,\mathrm{kg}]$ in the system under test is varied.

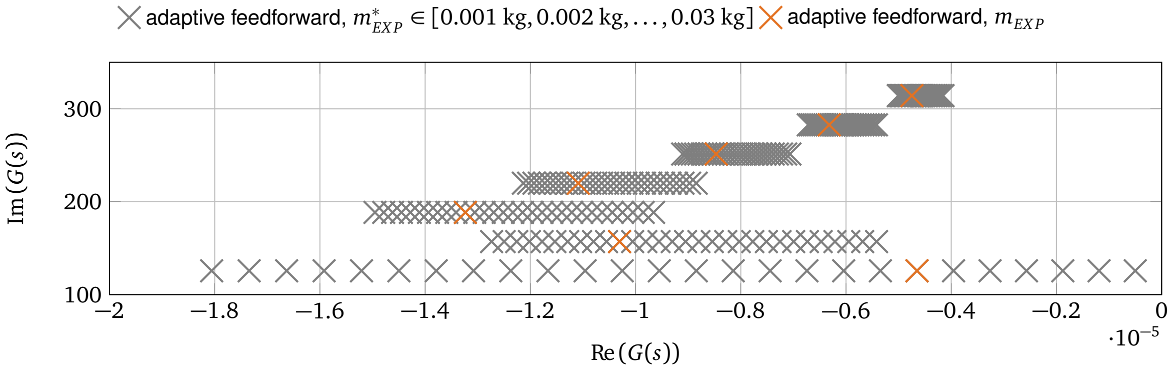


Figure 5.9: Detail from pole plot for closed-loop transfer functions $G(s)$: The interface transfer matrix $\boldsymbol{P}_{g,u}$ is identified with the original value of m_{EXP}, whereas the value of $m^*_{EXP} \in [0.001\,\mathrm{kg}, 0.002\,\mathrm{kg}, \dots, 0.03\,\mathrm{kg}]$ in the system under test is varied.

Another perspective on the robustness of the approach of using the adaptive feedforward filter is given in Fig. 5.10. 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 $\boldsymbol{H}^{est}_{g,u} = H^{est}_{g,u}$ is a complex scalar for $n_\Omega = 1$ and $n_{int} = 1$. 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 adaptation gain $\mu_{LMS} = 1$. The maps are shown for the excitation frequencies $\Omega_0 = 0.4\omega_0$, $\Omega_0 = 0.8\omega_0$, which are close to the resonance of the virtual component and for $\Omega_0 = \omega_0$ which is at the resonance of the coupled system. According to Fig. 5.10, 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 adaptation gain. Even though this seems robust, for complex interfaces with multiple DOF and

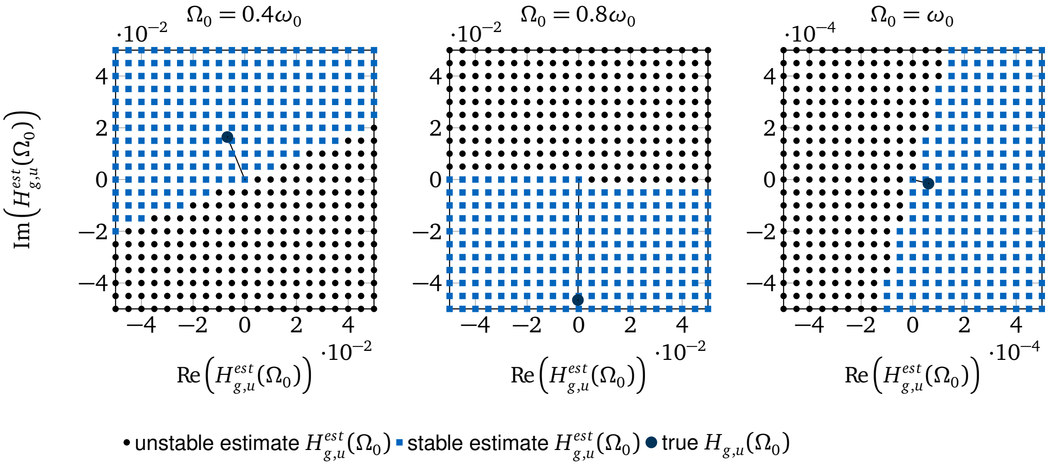


Figure 5.10: Stability map for a range of identified values $H_{g,u}^{est}(\Omega_0)$: The three plots show stable regions for excitation frequencies of $\Omega_0 = 0.4\omega_0$, $\Omega_0 = 0.8\omega_0$ and $\Omega_0 = \omega_0$. The true value $H_{g,u}(\Omega_0)$ lies within the stable region.

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 Sec. 5.4.

5.3 Passivity Preserving Adaptive Feedforward Filters

High adaptation gains, insufficient plant identification or changes in the system dynamics during the test can lead to unstable filter dynamics. In the following section, a method is presented which helps to overcome the stability issues: A supervisor is introduced which reduces the adaptation gain μ_{LMS} whenever necessary. To do so, the power-flows between the subcomponents are analyzed. Those power-flows are closely linked to passivity properties of the hybrid test. Passive systems are defined as systems which consume energy but do not produce energy. Coupling two arbitrary passive systems results in a passive overall system. One can think of the control system and actuation system of a hybrid test as an interconnection device between the virtual and the experimental component. This combination of the control system and the actuation system is referred to as the "transfer system" in this section. If the transfer system and the subcomponents are passive, the test setup is guaranteed to be passive. This implies that energy is only injected through external forcing on the virtual or the experimental component but not through the transfer system. Fig. 5.11 shows the power-flows in a hybrid test using adaptive feedforward filters. The transfer system is referred to as "passive" if the power-inflow into the system is always larger than the power-outflow. The power-inflow P_{in}^{ACT} to the transfer system is the sum of the power-inflow from the experimental component and the power-inflow from the virtual component. Both are the product of collocated interface forces and interface velocities. As a result, the power-inflow to the transfer system in Eq. (5.25) is the product of interface forces and the time derivative of the interface gap.

$$P_{in}^{ACT} = -\boldsymbol{\lambda}^{VIR}\dot{\boldsymbol{y}}_{VIR} + \boldsymbol{\lambda}^{VIR}\dot{\boldsymbol{y}}_{EXP} = \boldsymbol{\lambda}^{VIR}\dot{\boldsymbol{g}} \qquad (5.25)$$

A negative power-inflow into the transfer system—or in other words a power-outflow from the transfer system—implies undesirable energy injection into the hybrid test. In order to constrain the power-outflow of the transfer system which deteriorates its passive nature, the power-outflow is limited to P_{lim}. The limit $P_{lim} < 0$ is a negative value specific to the required power-outflow limit of the test. The objective of

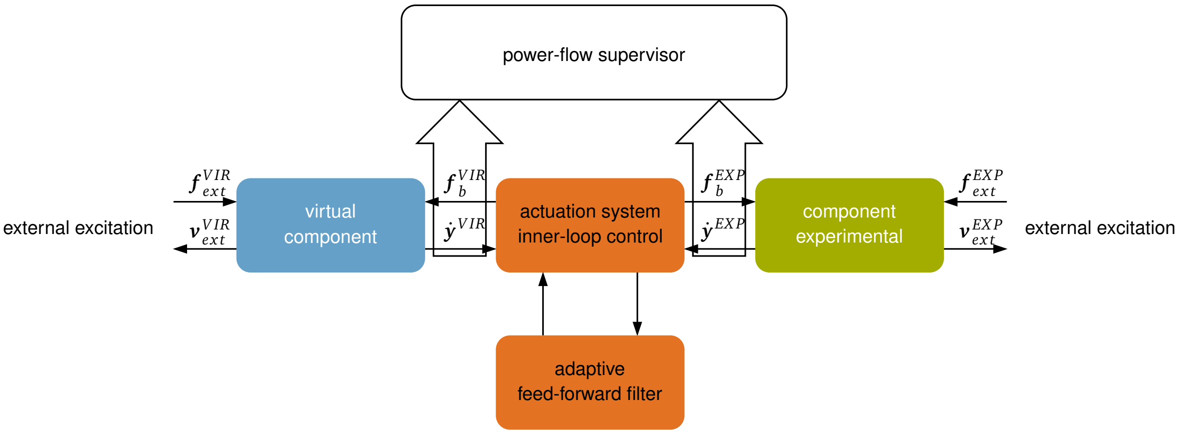


Figure 5.11: Power-flow diagram and power-flow supervisor

the power-flow supervision is to constrain the power-flow according to Eq. (5.26).

$$\lim_{t\to\infty} P_{in}^{ACT} \geq P_{lim} \tag{5.26}$$

Note that the energy-inflow is defined as the time integral of the power-inflow:

$$E_{in}^{ACT} = \int_0^t P_{in}^{ACT}(\tau)\mathrm{d}\tau$$

If the passivity constraint is violated, the two parameters μ_{LMS} and γ_{LMS} of the adaptation algorithm are adjusted following a simple heuristic approach. The objective is to reduce the actuator amplitudes, to restore passivity and to enable fast adaptation. Approaches where the leakage factor ν_{LMS} is adjusted according to the algorithm's performance have been proposed in literature such as [55]. The stability and the convergence of the LMS algorithm is controlled by the adaptation gain μ_{LMS}: A high adaptation gain μ_{LMS} leads to a fast convergence of the filter coefficients, but high values can lead to the unstable behavior of the algorithm. In contrast, lower values of μ_{LMS} cause a slower convergence, but stability is ensured if the value falls below the stability threshold. As a consequence, the adaptation gain μ_{LMS} is reduced using the exponential function in Eq. (5.27) with the initial adaptation gain μ_{init}, the variable a_μ and the user-defined exponent $b_\mu \geq 1$. The reasons for using an exponential function are to enable a faster drop of the adaptation gain μ_{LMS} in the initial phase, to ensure a fast restoration of passivity and a slower change in μ_{LMS} if it is closer to its optimal value. Fig. 5.12 shows the functions for some exemplary values b_μ.

$$\mu_{LMS} = \mu_{init} \cdot a_\mu^{b_\mu} \qquad \text{with} \qquad \mu_{LMS} \in [0,\, \mu_{init}] \;\; \forall \;\; a_\mu \in [0,\, 1] \tag{5.27}$$

The initial value is $a_\mu = 1$. If the power-inflow to the actuation system falls below the threshold P_{lim}—meaning that the system is not passive—the variable a_μ is reduced by the user-defined step-size parameter $\Delta_\downarrow a_\mu$. The lower bound for the variable a_μ is zero. As a result, the adaptation gain is bound by zero and the initial adaptation gain μ_{init}. The leakage factor ν_{LMS} results from the regularization factor γ_{LMS} according to Eq. (5.19). A low leakage factor—or equivalently, a high regularization factor—enforces lower filter coefficients. It is desirable to reduce the filter coefficients after a violation of the passivity constraint is detected. After the passive state is restored by the drop in the adaptation gain μ_{LMS}, the regularization should be reduced to ensure that the filter coefficients are adapted accurately to their optimal values. The regularization parameter γ_{LMS} is calculated using the exponential function in Eq. (5.28) with the user-defined maximum regularization factor γ_{max}, the variable a_γ and the user-defined exponent $b_\gamma \geq 1$. Fig. 5.12 shows the functions for some exemplary values b_γ. The nature of the exponential

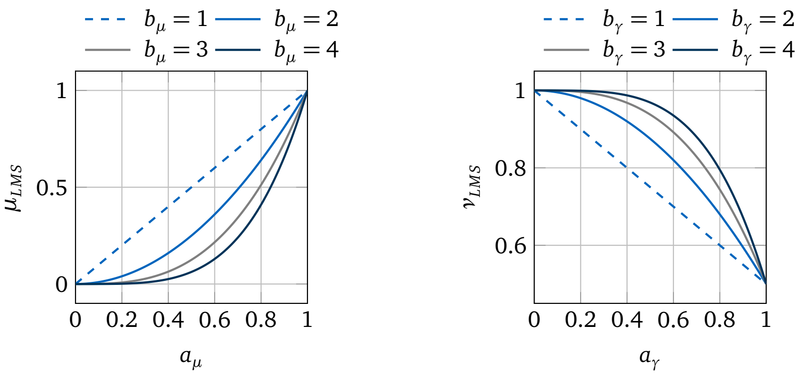


Figure 5.12: Parameter adjustment functions

function leads to a progressive behavior of the leakage: In cases of severe power-outflow, the filter coefficients fall faster.

$$\begin{aligned} &\gamma_{LMS} = \gamma_{max} \cdot a_\gamma^{b_\gamma} && \text{and} \\ &\nu_{LMS} = 1 - \bar{\mu}_{LMS}\gamma_{LMS} && \text{with} \qquad \nu_{LMS} \in [1 - \bar{\mu}_{LMS}\gamma_{max},\ 1] \quad \forall \quad a_\gamma \in [0,\ 1] \end{aligned} \tag{5.28}$$

The initial value is $a_\gamma = 0$. If the power-inflow to the actuation system P_{in}^{ACT} falls below the threshold P_{lim}—meaning that the system is not passive—the variable a_γ is increased by the user-defined step-size parameter $\Delta_\uparrow a_\gamma$. If the power-inflow to the actuation system P_{in}^{ACT} rises above the threshold P_{lim}—meaning that the system is assumed to be passive—the variable a_γ is reduced by the user-defined step-size parameter $\Delta_\downarrow a_\gamma$. The lower bound for the variable a_γ is zero and the upper bound is 1. As a result, the adaptation gain is bound by one and $1 - \bar{\mu}_{LMS}\gamma_{max}$. The complete procedure including the adaptation of μ_{LMS} and ν_{LMS} is summarized in the pseudo-code of Alg. 2. The simple

Algorithm 2 Power supervision for adaptive feedforward filters in hybrid testing

```
Initialize $a_\gamma := 0$ and $a_\mu := 1$
while adaptation is running do
    if $P_{in}^{ACT} < P_{lim}$ then
        Set $a_\gamma := \min(a_\gamma + \Delta_\uparrow a_\gamma, 1)$
        Set $a_\mu := \max(a_\mu - \Delta_\downarrow a_\mu, 0)$
    end if
    if $P_{in}^{ACT} > P_{lim}$ then
        $a_\gamma := \max(a_\gamma - \Delta_\downarrow a_\gamma, 0)$
    end if
    Set $\gamma_{LMS} := \gamma_{max} \cdot a_\gamma^{b_\gamma}$
    Set $\mu_{LMS} := \mu_{init} \cdot a_\mu^{b_\mu}$
end while
```

numerical test case of Chap. 1 helps to demonstrate the effects of changes in the adaptation parameters a_μ and b_μ as well as the power-inflow limit P_{lim}. The overall system emulated in the test is a lumped mass-spring-damper system as shown in Fig. 1.2. The system is split into a virtual component and an experimental component. In Fig. 1.2, the virtual component is depicted in blue, and the experimental component is depicted in green. The experimental component is controlled via an actuation system, which is depicted in orange in Fig. 1.2. Since the adaptive feedforward filter acts only at the excitation frequencies, measurement noise can cause a drift of the interface gap. A peak filter using the excitation frequency was applied at the actuator input in order to prevent those drift effects. The properties of the subsystems are listed in Tab. B.1 in Appendix B.

5.3.1 Adaptation gain step-size

The objective of the first numerical experiment is to investigate the influence of the changes in $\Delta_{\downarrow}a_{\mu}$. To do so, the parameter $\Delta_{\downarrow}a_{\mu}$ was varied while keeping the step-sizes $\Delta_{\uparrow}a_{\gamma}$ and $\Delta_{\downarrow}a_{\gamma}$, the exponents b_{γ} and b_{μ}, the initial adaptation gain μ_{init}, the power-outflow P_{lim}, the excitation amplitude A_{ext} as well as the excitation frequency f_{ext} constant. Tab. B.3 gives an overview of the parameters. Note that the initial adaptation gain $\mu_{init} = 10$ is a high value which causes an unstable system behavior without the passivity-preserving mechanism. As mentioned earlier, such instabilities can be caused in the same way by an insufficient system identification process as well as by changes in the system parameters. The values of the power-inflow P_{in}^{ACT}, the energy-inflow E_{in}^{ACT}, the leakage factor ν_{LMS}, the adaptation gain μ_{LMS}, and the interface gap g are recorded during the simulation. The simulation includes a 50 s identification phase. The identification was performed according to the procedure which is described in Sec. 5.2. The long duration of the identification phase is chosen to rule out all possible influences of identification errors on the simulation. Fig. 5.13 shows the development of the adaptation gain μ_{LMS} and the leakage factor ν_{LMS} during the simulation. As expected, the decay rate of the adaptation gain is higher for higher step-sizes $\Delta_{\downarrow}a_{\mu}$. The adaptation gain reduction is activated when the passivity constraint is violated. Depending on the reduction step-size $\Delta_{\downarrow}a_{\mu}$, the adaptation gain μ_{LMS} may overstep the optimal adaptation gain or reach it gradually. The resulting adaptation gains for the passive state vary: Higher step-sizes $\Delta_{\downarrow}a_{\mu}$ result in a lower end value. Lower step-sizes $\Delta_{\downarrow}a_{\mu}$ exhibit a slower decay of the adaptation gain but result in μ_{LMS} being closer to the optimal value. The behavior of the leakage factor ν_{LMS} with respect to the step-size $\Delta_{\downarrow}a_{\mu}$ follows the development of the adaptation gain μ_{LMS}: For a fast decaying adaptation gain μ_{LMS}, the drop in the leakage factor is lower since passivity is restored faster. On the other hand, a slow decay in μ_{LMS} makes a larger drop in ν_{LMS} necessary.
Fig. 5.14 shows the power-inflow P_{in}^{ACT} to the actuator system, the energy-inflow E_{in}^{ACT} to the actuator system and the envelope of the interface gap g. Note that the power-inflow is normalized with the peak values of the power-inflow into the reference system $P_{in,max}^{REF}$, the energy-inflow is normalized with the peak values of the energy in the reference system E_{max}^{REF} and the interface gap is normalized with the amplitude of displacement of the reference system y_{amp}^{REF}. The duration of the power-outflow of the actuator system is longer for lower step-size values $\Delta_{\downarrow}a_{\mu}$. For higher step-size values $\Delta_{\downarrow}a_{\mu}$, the duration gradually decreases. Accordingly, further energy injection is stopped after the settling of the algorithm. This fact can be seen in the diagram as the curves flatten after the settling time of the algorithm. It is noteworthy that the total energy-outflow is highest for the lowest step-size values $\Delta_{\downarrow}a_{\mu}$. The interface gap is a measure for the synchronization of the interface between the virtual component and the experimental component. The step-size $\Delta_{\downarrow}a_{\mu} = 0.0001$ shows a high peak value of the interface gap, while with increasing values of the step-size $\Delta_{\downarrow}a_{\mu}$, the peak gradually decrease. The duration which is needed to reach synchronization improves in the same way for higher step-size values. The reason is that lower choices for the step-size $\Delta_{\downarrow}a_{\mu}$ result in higher amplitudes at the start of the adaptation phase. The higher resulting adaptation gain μ_{LMS}, however, may result in a faster convergence later in the test. To conclude, there is a trade-off between the higher adaptation gain and a longer duration of the passivity violation on the one hand, and lower adaptation gains and shorter durations of the passivity violation, on the other hand. Choosing $\Delta_{\downarrow}a_{\mu}$ determines the quality of the resulting adaptation gain and the duration of passivity violations.

5.3.2 Leakage factor step-size

The objective of the second numerical experiment is to investigate the influence of the changes to $\Delta_{\uparrow}a_{\gamma}$. The parameter $\Delta_{\uparrow}a_{\gamma}$ was varied while the step-sizes $\Delta_{\downarrow}a_{\mu}$ and $\Delta_{\downarrow}a_{\gamma}$, the exponents b_{γ} and b_{μ}, the initial adaptation gain μ_{init}, power-outflow limit P_{lim}, the excitation amplitude A_{ext} as well as the excitation frequency f_{ext} remained constant. Tab. B.4 gives an overview of the parameters. The values of the power-inflow P_{in}^{ACT}, the energy-inflow E_{in}^{ACT}, the leakage factor ν_{LMS}, the adaptation gain μ_{LMS}, and

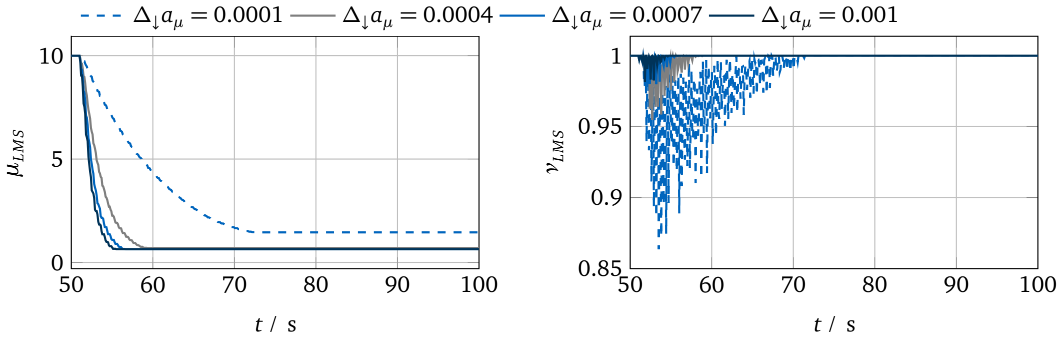


Figure 5.13: Adaptation gain and leakage factor for different adaptation gain step-sizes $\Delta_{\downarrow}a_{\mu}$

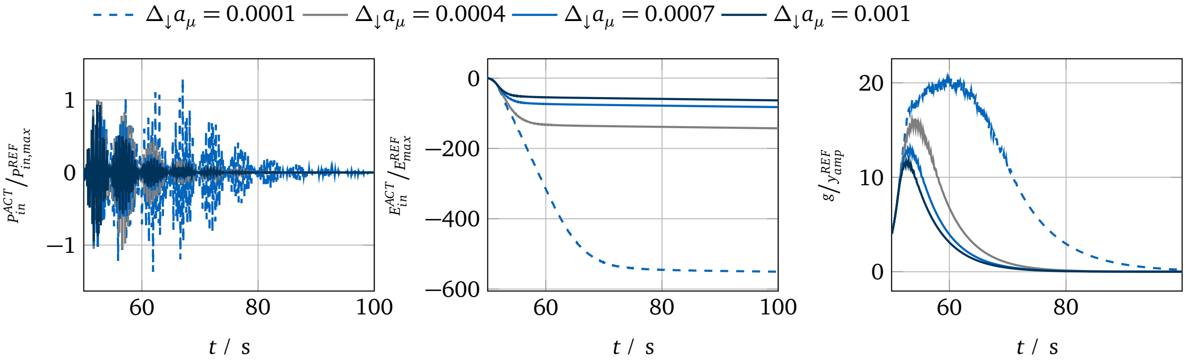


Figure 5.14: Power/energy-inflow to actuator system and learning curve for varying adaptation gain step-sizes $\Delta_{\downarrow}a_{\mu}$

the interface gap g are recorded during the simulation. Just like in the simulation described earlier in the text, a 50 s identification phase was applied.

Fig. 5.15 shows the effects of changes in the leakage factor step-size $\Delta_{\uparrow}a_{\gamma}$, on the leakage factor and the adaptation gain. As expected, higher step-sizes result in a larger drop in the leakage factor. Since the lower leakage factors reduce further power-outflow, the drop in the adaptation gain μ_{LMS} is steeper, and it settles faster for lower step-sizes $\Delta_{\uparrow}a_{\gamma}$. As a consequence, the duration of the reduction in the leakage factor is higher for high values of $\Delta_{\uparrow}a_{\gamma}$.

Fig. 5.16 shows the power and energy-inflow P_{in}^{ACT} and E_{in}^{ACT} as well as the envelope of the interface gap. The maximum power-outflows are close to the trigger value P_{lim} for higher step-size values $\Delta_{\uparrow}a_{\gamma}$. The reason is that lower leakage factors allow the filter coefficient—and as a consequence, the amplitudes—to drop faster. However, the increasing settling time prolongs the duration of the power-outflow. Equivalently, the curves for the energy-inflow flatten later when using higher values of $\Delta_{\uparrow}a_{\gamma}$. The total energy-outflow is lowest for low values of $\Delta_{\uparrow}a_{\gamma}$. Similar effects can be observed for the interface gap: High values of $\Delta_{\uparrow}a_{\gamma}$ help to reduce the peak values of the interface gap but prolong the settling time to full synchronization.

To conclude, high values of $\Delta_{\uparrow}a_{\gamma}$ reduce the peak values of the power-outflow and the interface gap but increase the settling time. Consequently, a trade-off between settling time and peak values has to be made.

5.3.3 Power-outflow limit

The objective of the third numerical experiment is to investigate the influence of the changes in the power-inflow limit P_{lim} which triggers the algorithm. As the trigger value P_{lim} is applied to the power-

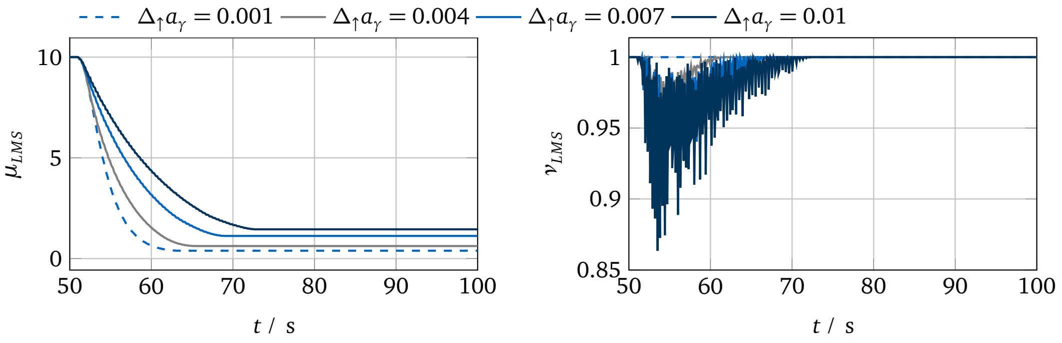


Figure 5.15: Adaptation gain and leakage factor for varying leakage factor step-sizes $\Delta_{\uparrow}a_{\gamma}$

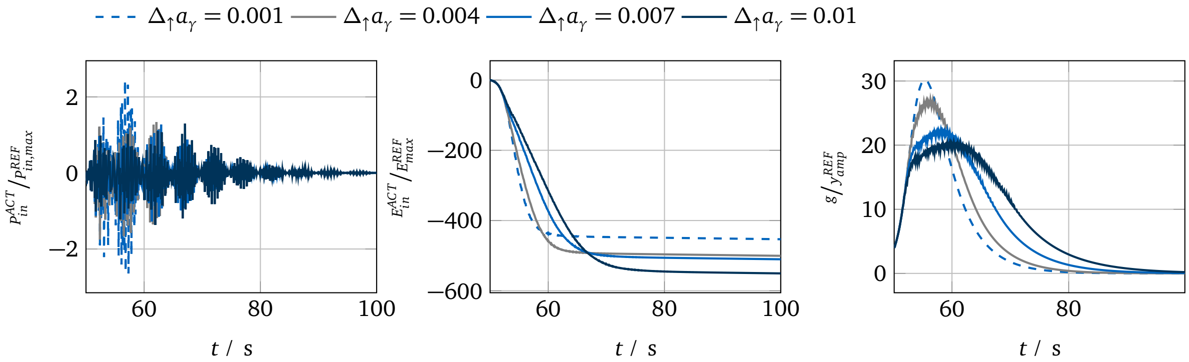


Figure 5.16: Power/energy-inflow to actuator system and learning curve for varying leakage factor step-sizes $\Delta_{\uparrow}a_{\gamma}$

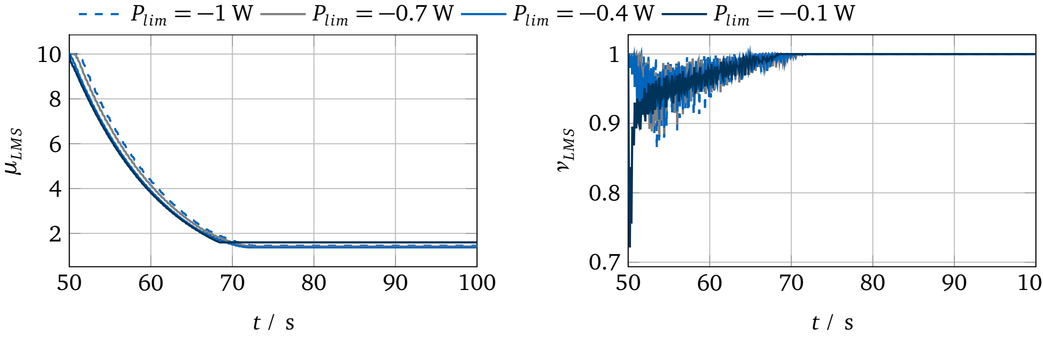


Figure 5.17: Adaptation gain and leakage factor for varying power-outflow limits P_{lim}

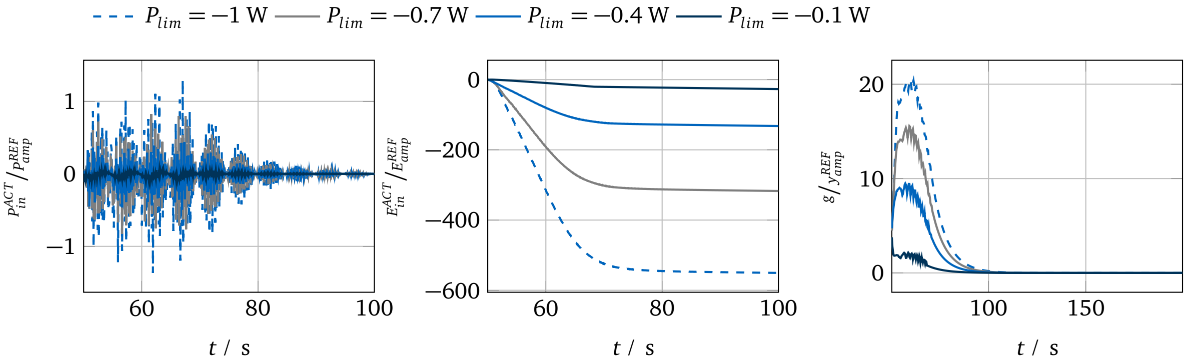


Figure 5.18: Power/energy-inflow to actuator system and learning curve for varying power-outflow limits P_{lim}

inflow values, it takes on negative values. During the numerical experiment, the step-sizes $\Delta_{\downarrow}a_{\mu}$, $\Delta_{\uparrow}a_{\gamma}$ and $\Delta_{\downarrow}a_{\gamma}$, the exponents b_{γ} and b_{μ}, the initial adaptation gain μ_{init}, the excitation amplitude A_{ext} as well as the excitation frequency f_{ext} remained constant. Tab. B.5 gives an overview of the parameters. The values of the power-inflow P_{in}^{ACT}, the energy-inflow E_{in}^{ACT}, the leakage factor ν_{LMS}, the adaptation gain μ_{LMS}, and the interface gap g are recorded during the simulation. Just as in the simulation described in Sec. 5.3.1 and Sec. 5.3.2, a 50 s identification phase was applied. Fig. 5.17 shows the development of the adaptation gain μ_{LMS} and the leakage factors ν_{LMS} for a range of values of P_{lim}. The curves for μ_{LMS} are shifted in time since the different values of P_{lim} trigger the reduction of the adaptation gain at different points in time. Slopes and final values of μ_{LMS} are not affected significantly by the choice of P_{lim}. The duration of the drops in the leakage factor ν_{LMS} are similar for all tested values of P_{lim}. However, the initial drop of ν_{LMS} is largest for the lowest trigger level $P_{lim} = 0.1$ W. The reason is that transient dynamics trigger the initial drop with amplitudes in the ranges of the trigger level $P_{lim} = 0.1$ W. Fig. 5.18 shows the power-inflow, energy-inflow and interface gap over time. As expected, different trigger values of P_{lim} result in different maximum power-outflow values. The duration of the outflow is similar for all values. As a consequence, the peak values of the interface gap gradually decrease for tighter power-outflow limits. Independently of the power-outflow limit the interface synchronization can get perfect. In summary, the power-outflow limit—even though it has only a minor influence on the final value of the adaptation gain—influences the expected peak amplitudes of the test.

5.3.4 Conclusion

The study leads to the following conclusions:

- Step size $\Delta_{\downarrow}a_{\mu}$ defines the convergence speed of the adaptation gain μ_{LMS}. High values can lead to lower values of μ_{LMS}.

- Step size $\Delta_{\uparrow} a_{\gamma}$ defines the convergence speed of the leakage factor ν_{LMS}. Higher values can reduce the peak values of power-outflow but increase the settling time.

- The power-outflow limit P_{lim} has only a slight influence on the final value of the adaptation gain, but higher values reduce the peak interface gap.

5.4 Recursive-Least-Squares Based Adaptation Law

The LMS-based algorithm from Sec. 5.2 may face problems where systems with multiple DOFs interfaces are coupled. The reason is that the performance of the algorithm is determined by the eigenvalue spread of the matrix $\boldsymbol{P}^T_{g,u}\boldsymbol{P}_{g,u}$. The presence of additional interface DoFs has a negative effect on this eigenvalue spread. The algorithm proposed in this section makes use of a Recursive-Least-Squares (RLS) adaptation law. RLS-based adaptation laws have been successfully applied to Active-Noise-Cancellation as shown e.g. in [41] or [49]. 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 the convergence of the RLS-based algorithm faster since in each time step the exact solution to the sum of squares problem is obtained. Furthermore, the error of the RLS algorithm approaches zero while—due to its stochastic nature—the LMS algorithm exhibits a remaining small error.

5.4.1 Adaption

The cost function (5.29) 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 adaptation while choosing a high value yields a slow adaptation. Ξ is a positive-definite matrix, which is a measure of confidence at the starting value $\boldsymbol{\theta}[0] = \boldsymbol{0}$. It can be set to $\Xi = \gamma_{RLS}\boldsymbol{I}$, with scalar γ_{RLS} and unity matrix $\boldsymbol{I}$. High values of γ_{RLS} will help to prevent jumps in $\boldsymbol{\theta}$ at the start of the adaptation phase, while low values will allow a faster adaptation to the optimal $\boldsymbol{\theta}^o$.

$$\begin{aligned} J[k] &= \sum_{i=0}^{k} \mu_{RLS}^{k-i} \boldsymbol{g}^T[i]\boldsymbol{g}[i] + \mu_{RLS}^{k+1}\boldsymbol{\theta}^T[k]\Xi\boldsymbol{\theta}[k] \\ &= \sum_{i=0}^{k} \mu_{RLS}^{k-i} (\boldsymbol{W}[i]\boldsymbol{P}_{g,u}[i]\boldsymbol{\theta}[k] + \boldsymbol{g}_{ext}[i])^T \cdot \\ &\quad \cdot (\boldsymbol{W}[i]\boldsymbol{P}_{g,u}[i]\boldsymbol{\theta}[k] + \boldsymbol{g}_{ext}[i]) + \mu_{RLS}^{k+1}\boldsymbol{\theta}^T[k]\Xi\boldsymbol{\theta}[k] \end{aligned} \tag{5.29}$$

Setting the gradient of the cost function (Eq. (5.31)) 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

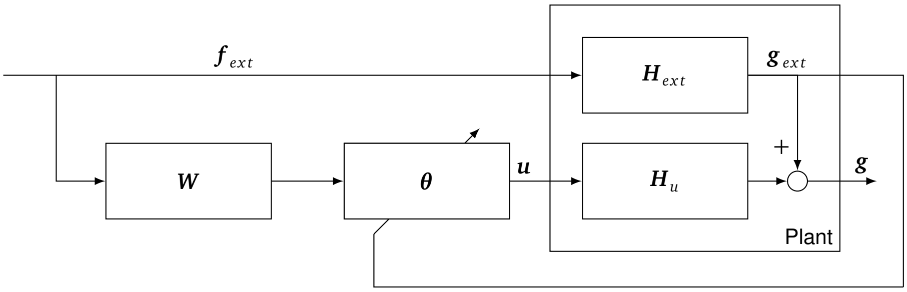


Figure 5.19: Block diagram for the RLS-based adaptation process.

available.

$$\nabla J[k] = \frac{\partial J[k]}{\partial \boldsymbol{\theta}[k]} \tag{5.30}$$

$$= \sum_{i=0}^{k} 2\mu_{RLS}^{k-i} \left(\boldsymbol{P}_{g,u}^T[i]\boldsymbol{W}^T[i]\boldsymbol{W}[i]\boldsymbol{P}_{g,u}[i]\boldsymbol{\theta}[k] + \boldsymbol{P}_{g,u}^T[i]\boldsymbol{W}^T[i]\boldsymbol{g}_{ext}[i] \right) + 2\mu_{RLS}^{k+1}\boldsymbol{\Xi}\boldsymbol{\theta}[k] \tag{5.31}$$

Eq. (5.32) is a rearranged form of the above-mentioned condition $\nabla J[k] = \boldsymbol{0}$. For convenience, the symbol $\boldsymbol{\Phi}$ is introduced for the terms related $\boldsymbol{\theta}$ and the symbol $\boldsymbol{s}$ is introduced for the terms depending on $\boldsymbol{g}_{ext}$:

$$\begin{aligned}
&\boldsymbol{\Phi}[k]\boldsymbol{\theta}[k] = \boldsymbol{s}[k] \\
&\text{with} \quad \boldsymbol{\Phi}[k] = \sum_{i=0}^{k} \mu_{RLS}^{k-i}\boldsymbol{P}_{g,u}^T[i]\boldsymbol{W}^T[i]\boldsymbol{W}[i]\boldsymbol{P}_{g,u}[i] + \mu_{RLS}^{k+1}\boldsymbol{\Xi} \\
&\text{and} \quad \boldsymbol{s}[k] = \sum_{i=0}^{k} -\mu_{RLS}^{k-i}\boldsymbol{P}_{g,u}^T[i]\boldsymbol{W}^T[i]\boldsymbol{g}_{ext}[i]
\end{aligned} \tag{5.32}$$

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. (5.32) 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. (5.32).

$$\begin{aligned}
\boldsymbol{\Phi}[k+1] &= \mu_{RLS}\boldsymbol{\Phi}[k] + \boldsymbol{P}_{g,u}^T[k+1]\boldsymbol{W}^T[k+1]\boldsymbol{W}[k+1]\boldsymbol{P}_{g,u}[k+1] \\
\boldsymbol{s}[k+1] &= \mu_{RLS}\boldsymbol{s}[k] - \boldsymbol{P}_{g,u}^T[k+1]\boldsymbol{W}^T[k+1]\boldsymbol{g}_{ext}[k+1]
\end{aligned} \tag{5.33}$$

A numerically stable and efficient way to perform the update of $\boldsymbol{\theta}[k]$ is the so-called QR-RLS algorithm. Alg. 3 describes the full update cycle. The derivation of the algorithm is described in Appendix D.2. $\boldsymbol{\Phi}^{\frac{1}{2}}[0]$ is initialized as $\sqrt{\gamma_{RLS}}\boldsymbol{I}$. The parameter vector is initialized with $\boldsymbol{\theta}[0] = \boldsymbol{0}$. For the correct adaptation of $\boldsymbol{\theta}$, an estimate of the interface transfer matrix $\boldsymbol{P}_{g,u}^{est}$ is necessary. The identification procedure for this estimate is described in the following section. Note that in contrast to the LMS-based algorithm, the contribution $\boldsymbol{g}_{ext}$ of the external forces to the interface gap $\boldsymbol{g}$ is used. The interface gap itself is not explicitly calculated during the adaptation process. A block diagram of the described system is shown in Fig. 5.19.

5.4.2 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. An RLS-based approach can be applied to identify $\boldsymbol{P}_{u,g}$. The squared identification

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

```
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
```

error $\boldsymbol{e}[i]$ serves as a cost function. 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 D.1. Correspondingly, vector $\boldsymbol{\theta}$ is rearranged in the matrix $\tilde{\boldsymbol{\theta}}$ according to Appendix D.1.

$$
\begin{aligned}
J[k] &= \sum_{i=0}^{k} \mu_{id,RLS}^{k-i}\left(\boldsymbol{e}^{T}[i]\boldsymbol{e}[i]\right) + \mu_{id,RLS}^{k+1}\tilde{\boldsymbol{P}}_{u,g}^{est,T}[k]\Xi\tilde{\boldsymbol{P}}_{u,g}^{est}[k] \\
&= \sum_{i=0}^{k} \mu_{id,RLS}^{k-i}\left(\boldsymbol{W}[i]\tilde{\boldsymbol{\theta}}[i]\tilde{\boldsymbol{P}}_{u,g}^{est}[k] - \boldsymbol{g}[i] + \boldsymbol{g}_{ext}[i]\right)^{T}\left(\boldsymbol{W}[i]\tilde{\boldsymbol{\theta}}[i]\tilde{\boldsymbol{P}}_{u,g}^{est}[k] - \boldsymbol{g}[i] + \boldsymbol{g}_{ext}[i]\right) \\
&\quad + \mu_{id,RLS}^{k+1}\tilde{\boldsymbol{P}}_{u,g}^{est,T}[k]\Xi\tilde{\boldsymbol{P}}_{u,g}^{est}[k]
\end{aligned}
\tag{5.34}
$$

Alg. 4 results from the recursive formulation based on the cost function in Eq. (5.34). The derivation is analogous to the one in the previous section: $\boldsymbol{P}_{g,u}^{est}$ is replaced by $\tilde{\boldsymbol{\theta}}$ and $\boldsymbol{\theta}$ is replaced by $\tilde{\boldsymbol{P}}_{g,u}^{est}$. The input signal $\boldsymbol{u}$ is produced by setting $\boldsymbol{\theta}$ to random values in time intervals of length t_{rand}.

5.4.3 Simultaneous Identification and Adaption

In the sequential adaptation and identification procedures described in the previous sections, a new identification step is necessary every time the dynamics of the system change. The solution is a simultaneous adaptation and identification procedure. The outline of the algorithm is given in Alg. 5. An identification phase precedes the simultaneous identification and adaptation process. This procedure makes it possible to initialize the interface transfer vector $\tilde{\boldsymbol{P}}_{g,u}^{est}[0]$ 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 is known or an additional identification of $\boldsymbol{g}_{ext}$ is performed.

5.4.4 Numerical experiment: Performance of adaptive feedforward filters

In order to evaluate the performance of the different algorithms, the learning curves of the LMS-based algorithm and the RLS-based algorithms are compared. The system from Chap. 1 is excited using a sinusoidal force $f_{ext} = \cos(\Omega_0 t)$ 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

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

```
Initialize $\boldsymbol{\Phi}^{id,\frac{1}{2}}[0] := \boldsymbol{\Xi}^{id,\frac{1}{2}}$, $\tilde{\boldsymbol{P}}^{est}_{g,u}[0] := \boldsymbol{0}$ and $k := 0$
while $t < t_{id}$ do
    if $\mod(t, t_{rand}) = 0$ 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 $\begin{bmatrix} \mathcal{B}^{id}_{1,1} & \mathcal{B}^{id}_{1,2} \\ \mathcal{B}^{id}_{2,1} & \mathcal{B}^{id}_{2,2} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}^{id,\frac{1}{2}}[k+1] & \boldsymbol{0} \\ \tilde{\boldsymbol{P}}^{est,T}_{g,u}[k+1]\tilde{\boldsymbol{\Phi}}^{id,\frac{1}{2}}[k+1] & \mathcal{B}^{id}_{2,2} \end{bmatrix}$
    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
```

Algorithm 5 Hybrid testing with RLS-based interface synchronization: simultaneous identification and adaptation

```
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
```

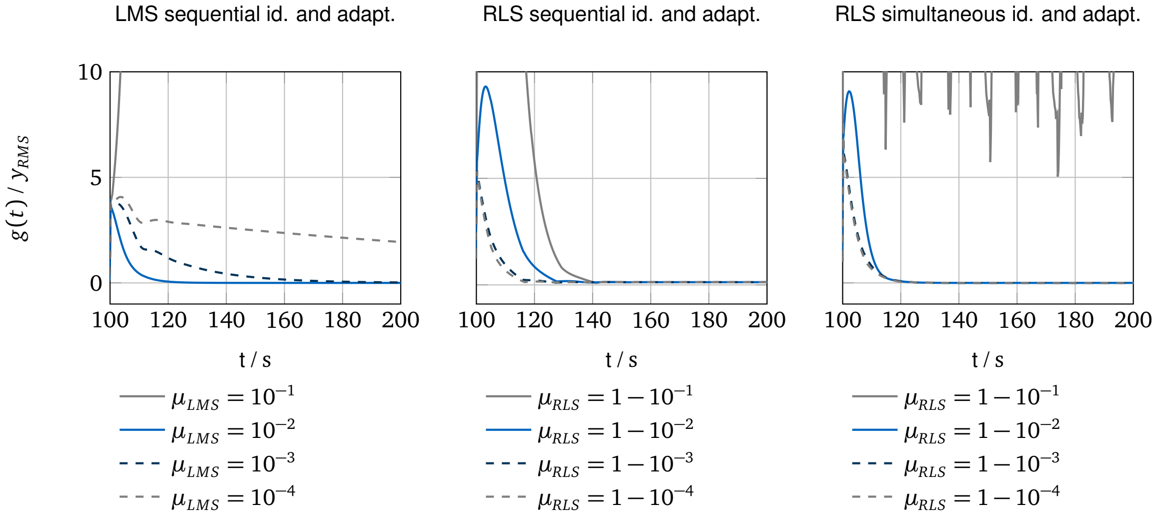


Figure 5.20: Learning curves for three adaptation strategies. The curves show the peak envelope of the interface gap g after an identification period with a duration of $t_{id} = 100$ s.

dependent on the choice of the adaptation gains or forgetting factors. All algorithms were tested with 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. 5.20 shows the envelopes of the interface gap $g[k]$. The interface gap is normalized with the root mean square value y_{RLS} of the reference interface displacements. Note that the envelope of the oscillations is shown in order to improve the readability of the graphs. The LMS-based adaptation law exhibits an unstable behavior for one value of the adaptation gain μ_{LMS}. The less aggressive choices of μ_{LMS} result in a stable adaptation. In contrast, the RLS-based algorithm with sequential identification and adaptation shows fast convergence for all forgetting factors. However, the gap only approaches zero if the dynamics are identified properly during the identification phase. The RLS-based algorithm with simultaneous identification and adaptation converges within less than 20 s. In this case, the most aggressive choice for the forgetting factor μ_{RLS} deteriorates convergence. The reason is the excitation of transient dynamics due to the quick changes in the interface force amplitudes, which is a violation of the steady-state assumptions. Note that the system remains stable.

Fig. 5.21 shows the envelopes of the interface gap $g[k]$ when 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 adaptation law exhibits an unstable behavior for $\mu_{LMS} = 10^{-1}$, $\mu_{LMS} = 10^{-2}$ and $\mu_{LMS} = 10^{-3}$. The RLS-based algorithm with sequential identification and adaptation converges, but an error due to the wrong identification remains. For the RLS-based algorithm with simultaneous identification and adaptation, the dynamics remain stable. The filter convergences and interface synchronization are achieved for choices of the forgetting factor $\mu_{RLS} \leq 1-10^{-2}$.

5.5 Adaptive Filter Based on Finite Impulse Responses

Another approach to design adaptive feedforward filters is applying finite impulse response (FIR) filters instead of harmonic basis functions. The approach is closely linked to the techniques shown in Sec. 5.1 and is based on Least-Mean-Squares filters. The general form is described in textbooks as e.g. [41] or [57]. Fig. 5.22 shows a block diagram of the control scheme.

Applied to real-time hybrid tests, the actuator input is the convolution product of the external excita-

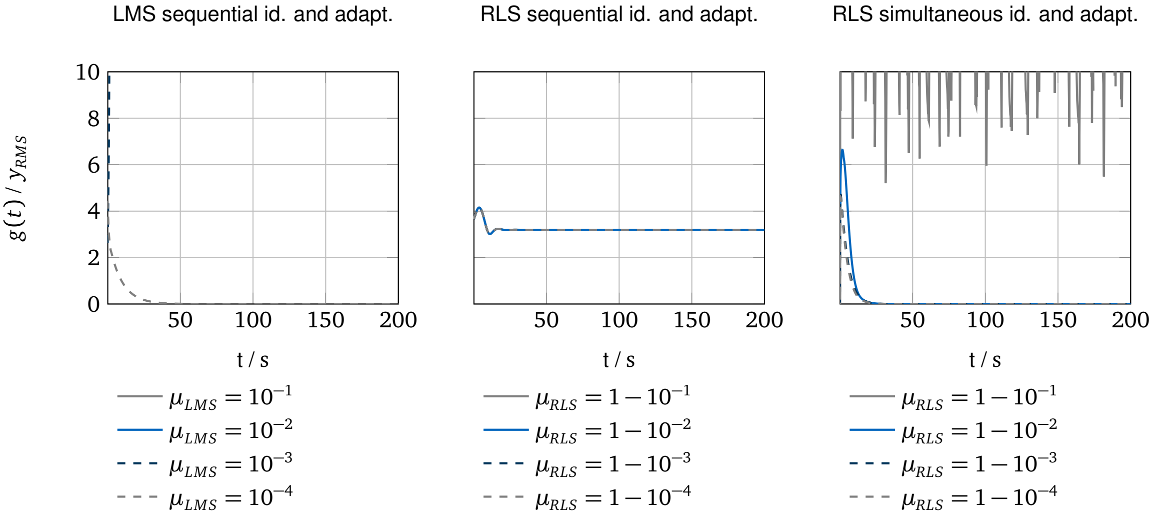


Figure 5.21: Learning curves for three adaptation strategies. The curves show the peak envelope of the interface gap g after an identification period with a duration of $t_{id}=0.01$ s.

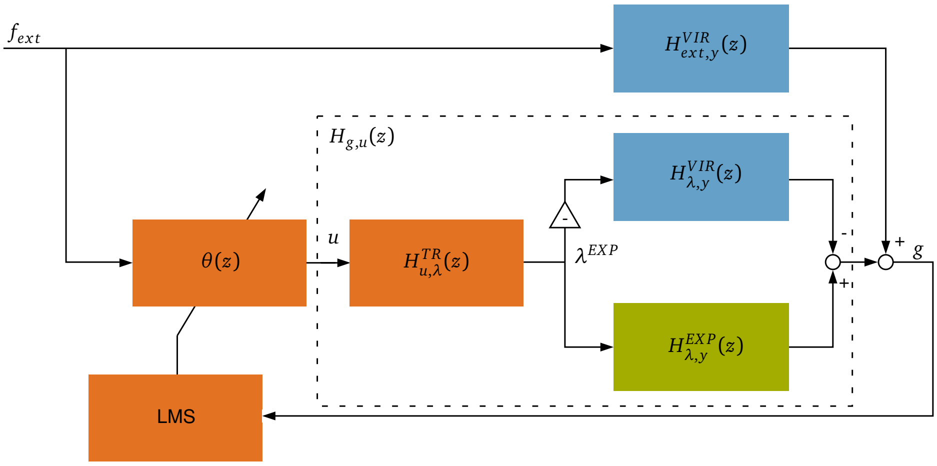


Figure 5.22: Block diagram of a real-time hybrid test with an adaptive feedforward filter based on a FIR filter: For readability reasons and without loss of generality, the excitation on the experimental component are omitted in the diagram.

tions $f_{ext}[k]$ with the feedforward filter coefficients $\boldsymbol{\theta}$:

$$\begin{aligned} u[k] &= \bar{\boldsymbol{f}}_{ext}[k]\boldsymbol{\theta}[k]^T \qquad \text{with} \\ \bar{\boldsymbol{f}}_{ext}[k] &= \begin{bmatrix} f_{ext}[k] & f_{ext}[k-1] & \dots & f_{ext}[k-N_\theta] \end{bmatrix}, \end{aligned} \tag{5.35}$$

where the previous values of the excitation signal are arranged in the vector $\bar{\boldsymbol{f}}_{ext}[k]$ such that the scalar product corresponds to the time discrete convolution product. Correspondingly, the coefficients of the feedforward filter $H_{FF}[n]$ are arranged in the vector $\boldsymbol{\theta}$ in reverse order.

5.5.1 Filtered-x Least Mean Squares Algorithm

The objective of the adaptive filter is to drive the interface gap g to zero. In order to derive an adaptation law, it is important to write the interface gap as function of the filter coefficients $\boldsymbol{\theta}$. In Eq. (5.36), the

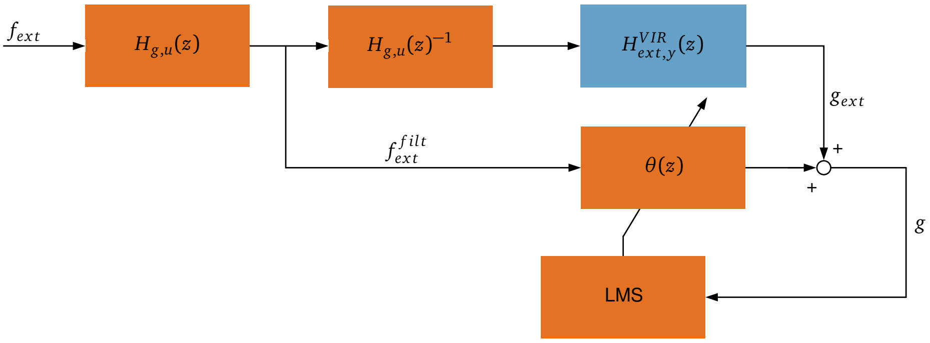


Figure 5.23: The block diagram from Fig. 5.22 is rearranged for the derivation of the fx-LMS algorithm. For readability reasons and without loss of generality, the excitation on the experimental component are omitted in the diagram.

interface gap contains the convolution product of filtered external forces and the feedforward filter:

$$g[k] = \bar{\boldsymbol{f}}^{filt}_{ext}[k]\boldsymbol{\theta}[k]^T + g_{ext}[k] \tag{5.36}$$

The convolution product is written as a vector product and $\bar{\boldsymbol{f}}^{filt}_{ext}[k]$ is a vector which contains values of the filtered excitation forces. The filtered[1] excitation forces are derived by rearranging Fig. 5.22. The rearranged block diagram is shown in Fig. 5.23. In the diagram, the impulse response $H_{g,u}[k]$ between actuator input and the interface gap moves into the direction of the external excitation. As a result, the filtered excitation forces $f^{filt}_{ext}[k]$ are obtained by convolution of the excitation forces with the impulse response $H_{g,u}[k]$. This is always possible since the dimension of all signals is one in this consideration. The convolution product which yields the filtered signal reads

$$f^{filt}_{ext}[k] = \sum_{i=0}^{k} H_{g,u}[k-i]f_{ext}[i]\Delta t. \tag{5.37}$$

Note that similar to Sec. 5.1, only an estimate of the impulse response $H_{g,u}[k]$ is necessary in order to perform the adaptation process. In practice, $H^{est}_{g,u}[k]$ is identified in a process which precedes the actual adaptation process. The corresponding vector which is used in the convolution product of Eq. (5.36) is

$$\bar{\boldsymbol{f}}^{filt}_{ext}[k] = \left[f^{filt}_{ext}[k] \quad f^{filt}_{ext}[k-1] \quad \ldots \quad f^{filt}_{ext}[k-N_\theta]\right]. \tag{5.38}$$

The squared interface gap and a regularization term serve as cost function

$$J[k] = \frac{1}{2}g[k]^2 + \gamma_{LMS}\boldsymbol{\theta}[k]\boldsymbol{\theta}[k]^T, \tag{5.39}$$

where γ_{LMS} is the regularization parameter. The update law for $\boldsymbol{\theta}[k+1]$ is formulated as a gradient descent:

$$\boldsymbol{\theta}[k+1] = \boldsymbol{\theta}[k] - \bar{\mu}_{LMS}\nabla J[k], \tag{5.40}$$

where $\bar{\mu}_{LMS}$ is the adaptation gain and can be used for tuning the filter. Substituting the gradient of the cost function $\nabla J[k]$ yields the update law of Eq. (5.41)

$$\boldsymbol{\theta}[k+1] = \nu_{LMS}\boldsymbol{\theta}[k] - \bar{\mu}_{LMS}\bar{\boldsymbol{f}}^{filt}_{ext}[k]g[k], \tag{5.41}$$

[1] The term "filtered x" refers to the input signal x of the filter. In the case of real-time hybrid testing the variable x corresponds to the excitation force f_{ext}.

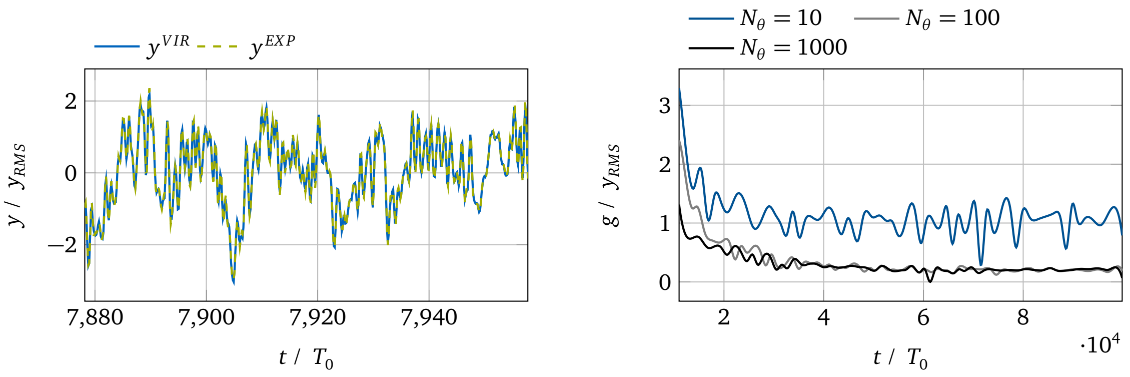


Figure 5.24: Interface displacements (on the left) after convergence using a filter with length $N_\theta = 100$ and learning curves (on the right) of tests with varying filter length.

where $\nu_{LMS} = 1 - \bar{\mu}_{LMS}\gamma_{LMS}$ is the leakage factor which prevents the filter coefficients from drifting. The adaptation gain can be normalized using the filtered external forces $\boldsymbol{f}^{filt}_{ext}$ according to [57] by substituting

$$\bar{\mu}_{LMS} = \frac{\mu_{LMS}}{\boldsymbol{f}^{filt}_{ext}\boldsymbol{f}^{filt^T}_{ext}} \tag{5.42}$$

The normalization term serves as an adaptive step width. The ideal system is stable if the normalized adaptation gain obeys $\mu_{LMS} < 1$. In practice, the value has to be chosen below this boundary due to inaccuracies in the identified filter $H_{g,u}$.

The prerequisite for accurate compensation is that the filter length N_θ is sufficient to represent the impulse response $H^{-1}_{g,u}H_{ext}$. As shown in [74] for periodic excitations, the condition of Eq. (5.43) is sufficient.

$$N_\theta \geq \frac{2\pi}{\Omega_0} f_S = \frac{T_{0,ext}}{T_S} \tag{5.43}$$

Here, Ω_0 is the excitation frequency, $T_{0,ext}$ is the period of the excitation , f_S is the sampling frequency and T_S is the sampling step width. Correspondingly, $T_{0,ext}/T_S$ is the number of time steps during one oscillation. In words, the filter is sufficiently long, if the number of time steps during one oscillation is smaller than the number of filter coefficients.

5.5.2 Numerical experiment: Transient dynamics

The method is demonstrated in a numerical experiment using the system from Sec. 4.2.3 with the parameters $\kappa^{ACT} = 0.1$, $\kappa^{EXP} = 0.2$, $\kappa^{P} = 0.1$, $\phi^{ACT} = 0.1$, $\zeta^{VIR} = 0.2$, and $\zeta^{ACT} = 0.1$. In order to show that the approach works with arbitrary transient excitations, a random noise signal was applied as an external force. Fig. 5.24 gives the interface displacements and the learning curves of the experiment. The displacements are normalized with the root mean square interface displacement y^{RMS} of the reference system. The time is normalized with the period time T_0 which corresponds to the eigenfrequency of the reference system. The subspace plots of Fig. 5.25 show the interface synchronization for different filter lengths N_θ. An undistorted, linear curve with slope 1 indicates a perfect synchronization. Time lags between the interface displacements cause an ellipsoid form of the curve. Amplitude errors cause deviations from the slope. The synchronization plot related to a filter of length $N_\theta = 10$ shows strong deviations from a perfect synchronization. Increasing the filter lengths to $N_\theta = 100$ and $N_\theta = 1000$ improves the results.

In contrast to the approach based on harmonics (Sec. 5.1), the FIR-based FxLMS filter is capable of

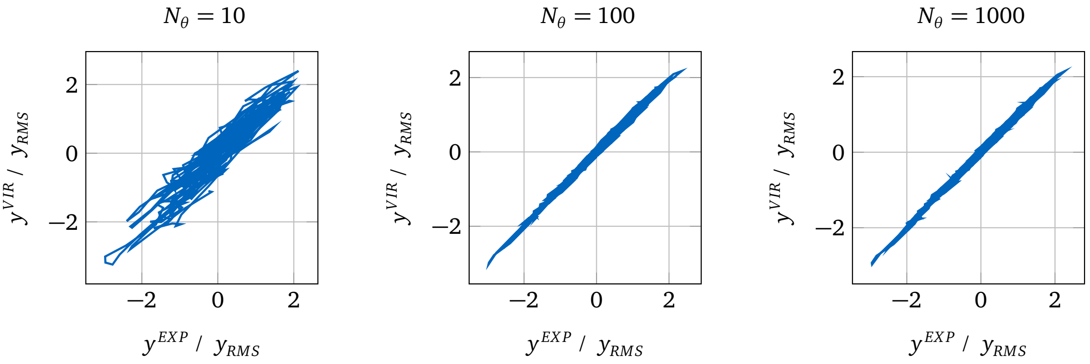


Figure 5.25: Synchronization subspace plot of the interface displacements using adaptive filters with varying filter length.

coupling systems with arbitrary excitations. Transient responses can be simulated if the system dynamics only change slowly. Nevertheless, if harmonic excitations can be assumed, the harmonic approach is preferable. The damping rate has a strong influence on the number of coefficients in the FIR filter (see Sec. 2.1), while their number in the harmonic approach only depends on the number of harmonics in the excitation signal. This can cause computationally demanding filters for lightly damped systems which is relevant especially for test setups with more than one interface DOF.

5.6 Combined Feedback and Feedforward Control

Adaptive feedforward filters with harmonic basis functions, which are discussed in Sec. 5.2 and Sec. 5.4, are well suited to couple systems with harmonic excitations that operate in steady state. If neglecting the transient dynamics is not possible, the method has to be extended. One approach is to combine adaptive feedforward filters and passivity-based control as proposed in [78]. The method allows controlling the slow transients and the low-frequency content by feeding back the virtual interface displacement as a demand signal to the actuation system. The feedback structure of the real-time hybrid testing scheme can lead to unstable behavior. A passivity-based controller—as it is described in Sec. 4.4—stabilizes the system. The signal of the adaptive feedforward controller is augmented to the system and improves the accuracy of the steady-state responses.

A block diagram of the control structure is given in Fig. 5.26: Just as in Sec. 4.4, a displacement controlled actuation system is the prerequisite for implementing the technique. A virtual dissipative element with adaptive damping rate enforces the passivity of the system. The correction signal from the adaptive feedforward filter is augmented to the interface displacement of the virtual component $\boldsymbol{y}^{VIR}$. Hence, the new actuator input reads

$$\boldsymbol{u} = \boldsymbol{y}^{VIR} + \boldsymbol{W}\boldsymbol{\theta}, \tag{5.44}$$

where the basis function matrix $\boldsymbol{W}$ and the parameter vector $\boldsymbol{\theta}$ are defined according to Sec. 5.1. The characteristics of the augmented actuator input are defined by $\boldsymbol{\theta}$, and $\boldsymbol{\theta}$ is updated based on the LMS-based adaptation law from Sec. 5.2. Accordingly, the adaptation law is driven by the residual interface gap $\boldsymbol{g}$.

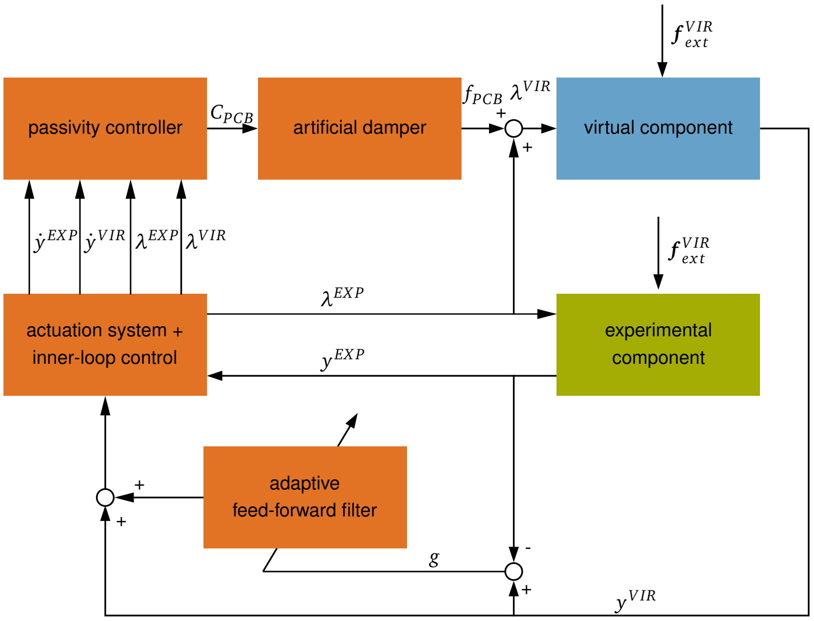


Figure 5.26: Block diagram of a real-time hybrid test with a passivity based controller and an adaptive feedforward filter.

Chapter 6

Experiments with Feedforward Based Approaches

Parts of this chapter have been submitted for publication in [7] and [9].

In this chapter, experimental results related to the techniques from Chap. 5 are presented, and the practical implementation of the techniques is discussed.

6.1 Coupling Two Cantilever Beams

The first section experimentally demonstrates real-time hybrid testing with adaptive feedforward filters as it is presented in Sec. 5.2 and Sec. 5.4. The experiments are performed using two cantilever beams which are coupled during the test. The excitation is chosen such that the coupling of virtual and experimental component is possible with only two actuators.

6.1.1 Test setup

The objective of the experiment is to replicate the dynamics of a reference system as shown in Fig. 6.1. 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 $20\,\mathrm{Nm}$ and washers were placed between the base and the beam. The reason for using the washers is that the modeling of the joints is simplified. Foam layers serve as dampers which are mounted on both subcomponents. The damping elements are necessary to keep the actuator displacements within the workspace at the resonances. The workspace is constrained because standard voice coil actuators use compliant elements for axial guidance. The subcomponents are bolted

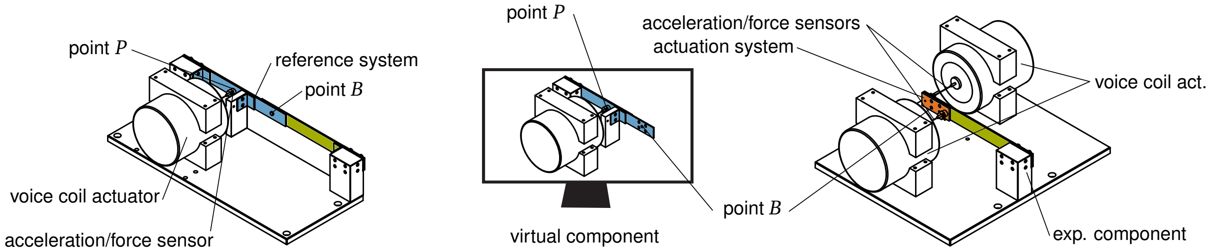


Figure 6.1: Reference system with actuator and sensors used for the admittance measurement. The test setup allows to couple the virtual component and the experimental component such that the dynamic behavior of the reference system is emulated.

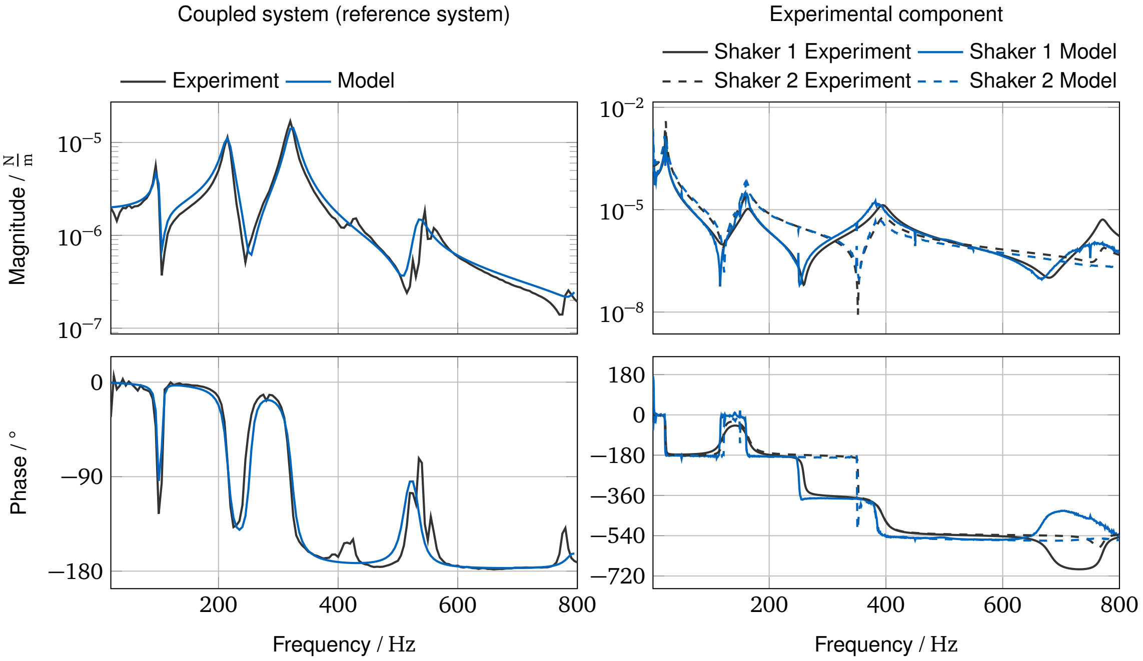


Figure 6.2: Measurement and model: Bode plots of driving point receptance of the coupled system (reference system) at point B and the test rig at mounting points of shaker 1 and 2.

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. 6.3 and Tab. 6.1. 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. 6.4 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 torsional modes. They 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. 6.2. They are retrieved from measurement and a calibrated model. The model consists of finite element models of the beams and rotational spring-damper-mass elements, which represent the joints at the washers. The spring and mass coefficients were updated to match the first four resonance frequencies. The resulting models were used for the virtual subcomponent.

Virtual component

An implicit Newmark time integration scheme (parameters $\beta = 0.25$ and $\gamma = 0.5$) is used to retrieve the response for the virtual component. A calibrated finite element model represents the virtual component. The model is reduced using the Craig-Bampton method (see [26]) with 20 fixed interface modes and the constraint modes. The interfaces used in the reduction are the excitation point and the interface between the substructures. The nodes which define the interfaces were rigidified, and rotations and displacements describe the interfaces. The time step size for simulating the virtual component was $\Delta t = 0.2\,\mathrm{ms}$, as it was for the adaptation law.

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 setup is shown in Fig. 6.5. The position and direction

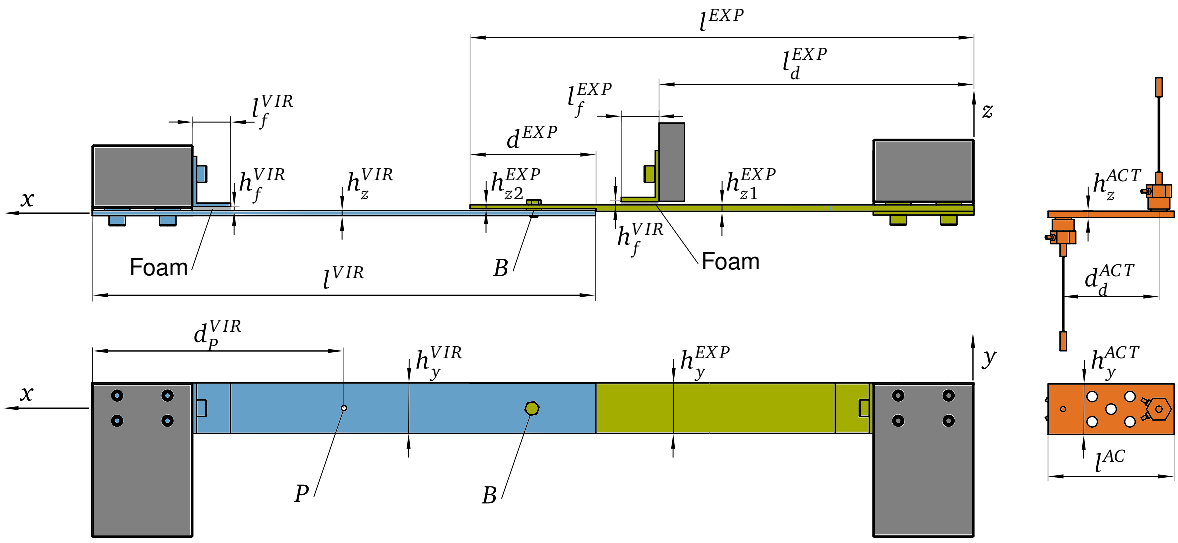


Figure 6.3: Dimensions of the structure used in the experiment: The values can be gathered from Tab. 6.1.

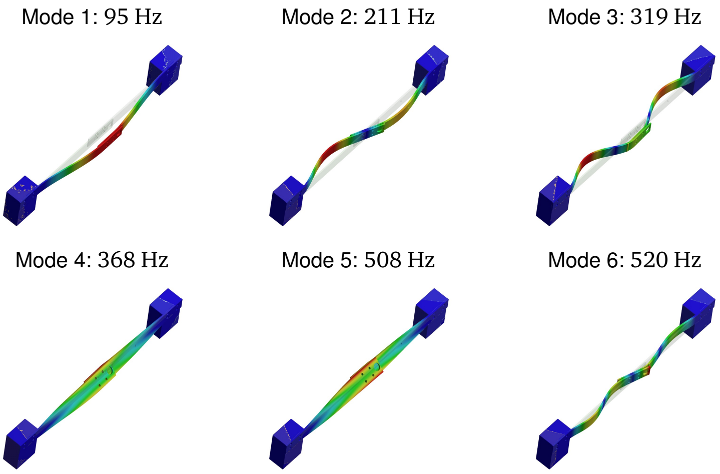

Mode 1: 95 Hz
Mode 2: 211 Hz
Mode 3: 319 Hz
Mode 4: 368 Hz
Mode 5: 508 Hz
Mode 6: 520 Hz

Figure 6.4: First normal modes of the reference system

Test Specimen (EXP)	
d^{EXP}	100 mm
h_{z1}^{EXP}	4 mm
h_{z2}^{EXP}	2 mm
h_y^{EXP}	40 mm
l^{EXP}	400 mm
l_f^{EXP}	30 mm
l_d^{EXP}	250 mm

Virtual Component (VIR)	
d_P^{VIR}	200 mm
h_z^{VIR}	4 mm
h_y^{VIR}	40 mm
h_f^{VIR}	30 mm
l^{VIR}	400 mm
l_f^{VIR}	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 6.1: Dimensions of the structure used in the experiment

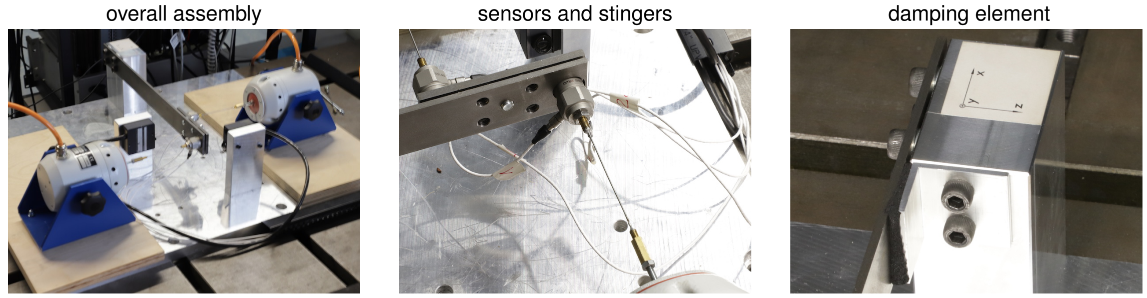


Figure 6.5: Test rig. The laser sensors which can be seen in the overall assembly are not used in the test. The damping element is the black foam structure in the right picture.

of the excitation are chosen such that only those deflections are excited that rotate the interface around the y-axis and cause displacements along the z-axis. As a result, the actuation system which imitates the coupling between the two beams can be realized 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 adaptation 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.

Interface states and forces

The general procedure to retrieve interface states and forces is described in Sec. 3.2. For the experiment, the translational acceleration sensors ($\ddot{z}^{EXP}_{act,1}$ and $\ddot{z}^{EXP}_{act,2}$) and force sensors (actuator forces F_{A1} and F_{A2}) are used. 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, the actuator displacements $z_{act,1}$ and $z_{act,2}$ are used 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. (6.1).

$$\boldsymbol{y}^{VIR} = \begin{bmatrix} z^{VIR}_{act1} \\ z^{VIR}_{act2} \end{bmatrix} = \begin{bmatrix} 1 & -d^{ACT}/2 \\ 1 & d^{ACT}/2 \end{bmatrix} \begin{bmatrix} z^{VIR}_{B} \\ \alpha^{VIR} \end{bmatrix} \qquad \boldsymbol{y}^{EXP} = \begin{bmatrix} z^{EXP}_{act,1} \\ z^{EXP}_{act,2} \end{bmatrix} \tag{6.1}$$

On the experimental component's side, time integration is applied to the acceleration signals in order to retrieve z^{EXP}_{act1} and z^{EXP}_{act2}. Possible drift effects are removed with a peak filter of the interface forces at the excitation frequencies. Moment $M_{B,y}$ and the force $F_{B,z}$ define the interface force vector $\boldsymbol{\lambda}^{VIR}$. According to the sign conventions from Sec. 2.1, the interface forces from the experimental component are applied to the virtual component with opposite sign.

$$\boldsymbol{\lambda}^{VIR} = \begin{bmatrix} F^{VIR}_{B,z} & M^{VIR}_{B,y} \end{bmatrix}^T = -\boldsymbol{\lambda}^{EXP} = -\begin{bmatrix} F^{EXP}_{B,z} & M^{EXP}_{B,y} \end{bmatrix}^T \tag{6.2}$$

Fig. 6.6 illustrates the free-body diagram of the transfer system used in the experiment. The inertia properties are defined by the mass of the steel plate m^{ACT} and moment of inertia θ^{ACT}_y. The interface

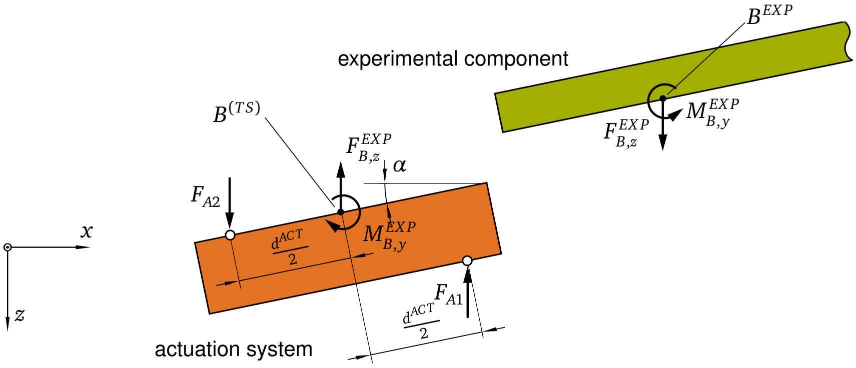


Figure 6.6: The interface state ($\boldsymbol{y}^{EXP}$ and $\boldsymbol{\lambda}^{VIR}$) includes interface rotation, displacement, force and moment. They are calculated via a rigid body model of the transfer system.

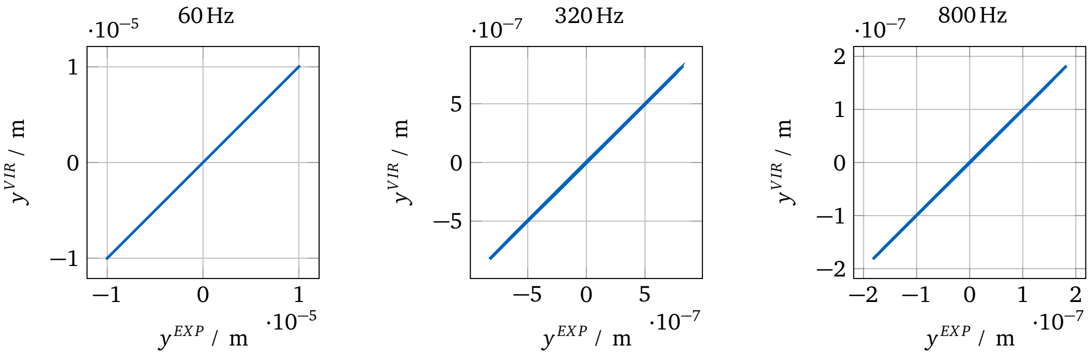


Figure 6.7: The synchronization plots show the relation between interface displacements of the virtual and the experimental component. Ideal synchronization is indicated by a straight diagonal line with a slope of 1. Phase shifts cause an ellipsoidal shape of the synchronization curve while amplitude errors change the slope.

forces are calculated by the solution of the dynamic equation of the actuation system (6.3).

$$\begin{bmatrix} m^{ACT} & 0 \\ 0 & \theta^{ACT}_y \end{bmatrix} \begin{bmatrix} \ddot{z}^{ACT}_B \\ \ddot{\alpha}^{ACT} \end{bmatrix} = -\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} F^{EXP}_{B,z} \\ M^{EXP}_{B,y} \end{bmatrix} + \begin{bmatrix} -1 & 1 \\ {d^{ACT}}/{2} & {d^{ACT}}/{2} \end{bmatrix} \begin{bmatrix} F_{A1} \\ F_{A2} \end{bmatrix} \tag{6.3}$$

6.1.2 Interface Synchronization

The synchronization plots in Fig. 6.7 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 with sinusoidal excitation with the frequencies 60 Hz, 320 Hz and 800 Hz. A QR-RLS adaptation law with simultaneous adaptation and identification was used for synchronization control. Here, a forgetting factor $\mu_{ad} = 0.999$ for the adaptation and a forgetting factor $\mu_{id} = 0.9999$ for the identification were applied. The duration of the identification was $t_{id} = 5$ s. 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.

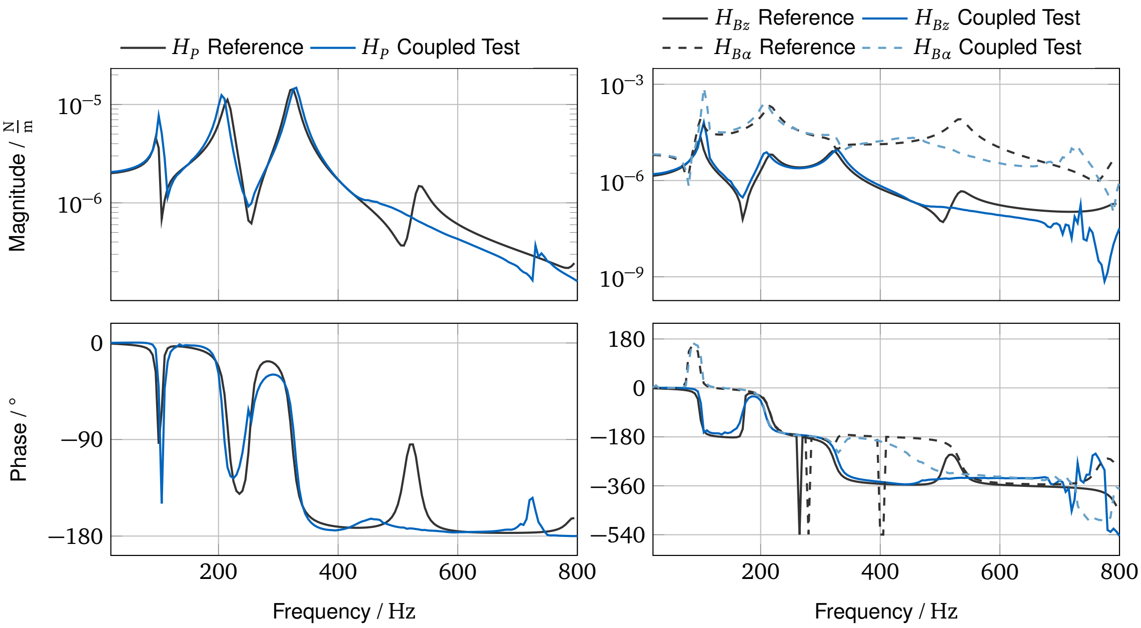


Figure 6.8: Driving point receptance $H_{P_z}(\omega)$, $H_{By}(\omega)$ and $H_{B\alpha}(\omega)$ of the reference system and the coupled test

6.1.3 System Transfer Functions

To verify the adaptive feedforward 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, $H_{P_z}(\omega)$ 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/rotations of the interface B for the assembled system. An RLS-based adaptation law with simultaneous adaptation and identification was used as coupling algorithm. For the adaptation, 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 adaptation algorithm when using a sine sweep excitation—occur at the resonances. The potentially incomplete interface synchronization would distort the results. The results in Fig. 6.8 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. 6.2, 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. This can be concluded from reference measurements without cables and a Laser-Doppler vibrometer. Moments are not accounted for by the sensors and affect the damping of rotational interface modes.

6.1.4 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 adaptation gain choice. For evaluating the two properties, sinusoidal excitation with a frequency of $\Omega = 2\pi \cdot 60\,1/\mathrm{s}$ and with $\Omega_{ext} = 2\pi \cdot 320\,1/\mathrm{s}$ is used. In order to make the two strategies comparable, four different adaptation gains were applied for each strategy. For the analysis, a learning curve—which is the envelope of the normalized mean-square error (NMSE) of Eq. (6.4)—is used:

$$NMSE[i] = \frac{\boldsymbol{g}^T[i]\boldsymbol{g}[i]}{\boldsymbol{y}_{\max}^T \boldsymbol{y}_{\max}} \tag{6.4}$$

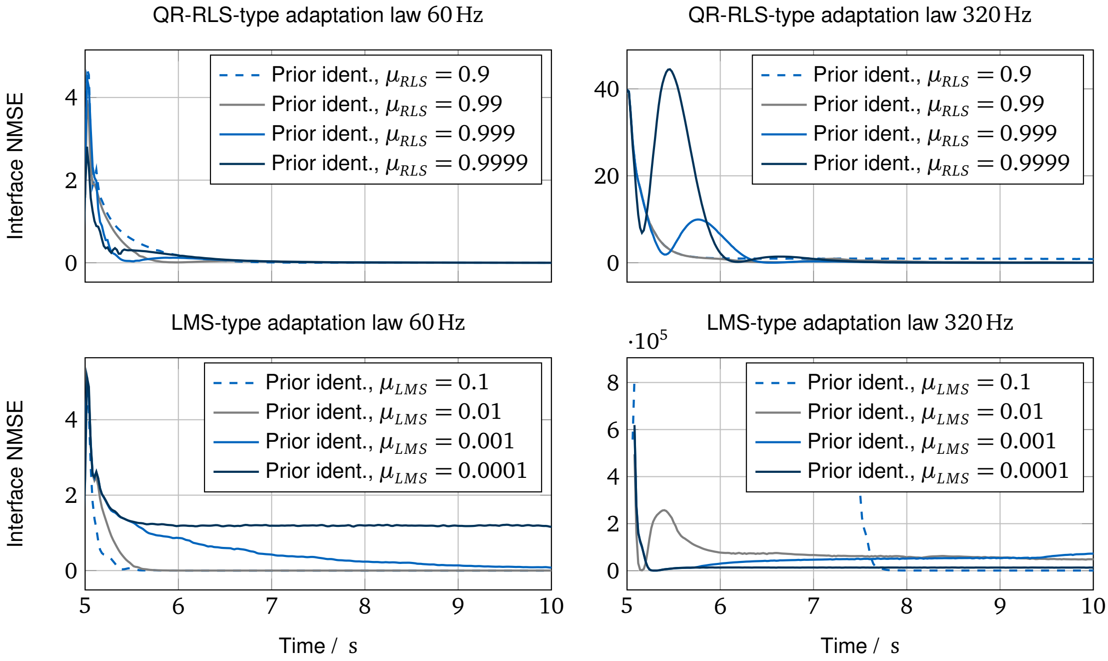


Figure 6.9: Learning curves for different adaptation laws. The curves show the peak envelope of the normalized mean square error between virtual and experimental interface over time. The duration of the prior system identification phase is 5 s. The QR-RLS algorithm is used with simultaneous system identification.

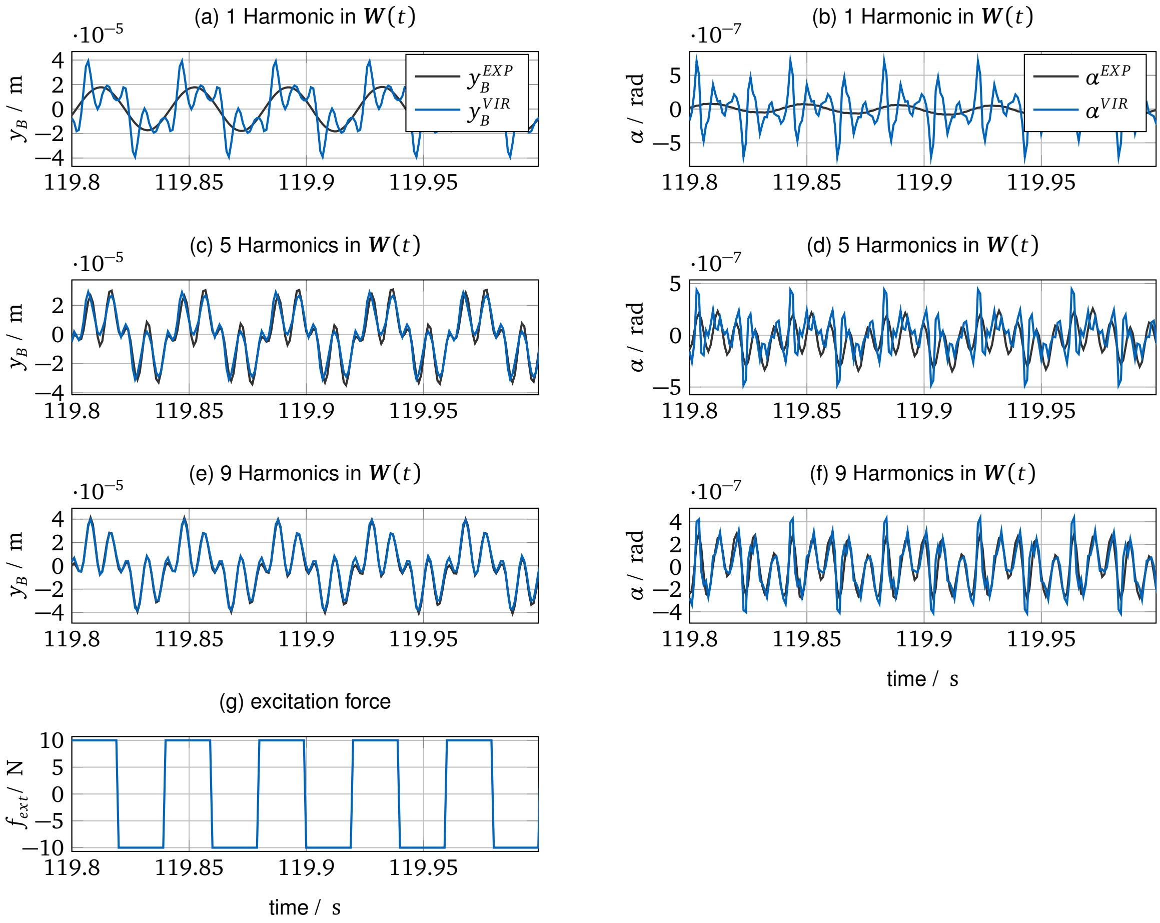
The normalization is performed using the vector of the maximum amplitude $\boldsymbol{y}_{\max}$ of the coupled system. Fig. 6.9 shows the resulting learning curves. Even though the choice for an aggressive forgetting factor μ_{RLS}—corresponding to a small value of μ_{RLS}—may deteriorate the performance of the adaptation law, the results suggest that the stability of the RLS-based adaptation is not affected. In contrast, the choice of a high adaptation gain μ_{LMS} for the LMS adaptation law may cause an unstable behavior as seen at $\Omega = 2\pi \cdot 320\,\mathrm{Hz}$ for $\mu_{LMS} = 0.1$.

6.1.5 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 $T = {}^1\!/\!_{\Omega_0}$). In order 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. 6.10 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)), five harmonics (plot (c) and (d)) and nine harmonics (plot (e) and (f)). An RLS-based adaptation law with simultaneous adaptation, and identification was used as coupling algorithm. For the adaptation 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 $t_{id} = 20$ s. The results show that the increasing number of harmonics enriches the function space for the control signal and improves 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.



(a) 1 Harmonic in $\boldsymbol{W}(t)$

(b) 1 Harmonic in $\boldsymbol{W}(t)$

(c) 5 Harmonics in $\boldsymbol{W}(t)$

(d) 5 Harmonics in $\boldsymbol{W}(t)$

(e) 9 Harmonics in $\boldsymbol{W}(t)$

(f) 9 Harmonics in $\boldsymbol{W}(t)$

(g) excitation force

Figure 6.10: Interface synchronization for a square wave external excitation: The interface synchronization improves with additional harmonics in the basis function matrix $\boldsymbol{W}$.

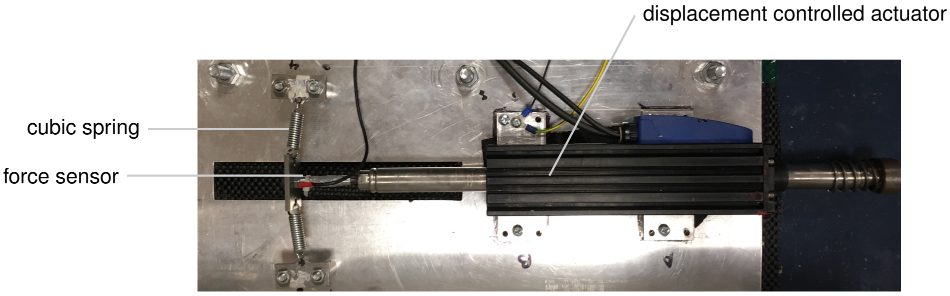


Figure 6.11: Test rig with a cubic spring

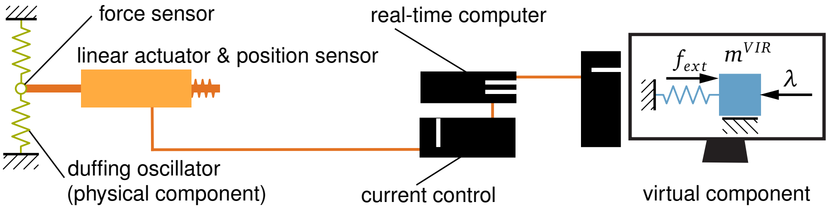


Figure 6.12: Scheme of test setup for the experiments with a cubic spring

6.2 System with a Cubic Spring

Sec. 5.3 presents a method which modifies the adaptation gains of adaptive feed-forward filters in order to preserve the stability of the tests. Sec. 5.6 describes a method which combines adaptive feed-forward filters with feedback based control. For the experimental validation of both approaches, a cubic spring is used as a physical subcomponent and a linear lumped-mass system as a virtual subcomponent. The test rig is shown in Fig. 6.11 and a scheme of the test setup is depicted in Fig. 6.12. The cubic spring is realized using two linear springs with all forces acting perpendicular to the spring axis. The mathematical expression for the spring force is given by Eq. (6.5).

$$f^{EXP} = k_3^{EXP} x^3 + k^{EXP} x \tag{6.5}$$

The spring constants k_3^{EXP} and k^{EXP} were identified using a least-mean-squares fit. They are given in Tab. 6.2. The virtual component is a mass-spring-damper system and receives the external forces. A linear actuator—a Copley ST2508S electromagnetic linear actuator—applies the coupling forces to the experimental component. The position is measured using the internal sensor of the linear actuator. The actuator is controlled using a cascaded control scheme acting with a proportional term on the position demand, and with a proportional and integral term on the velocity demand. Friction has a significant effect on the actuator dynamics and, due to its non-linear nature, the performance of hybrid-testing control schemes deteriorates. For that reason, a friction compensation scheme is implemented which acts on the input to the current control loop. The actuator operates with a position saturation at $2.5 \cdot 10^{-2}$ m around the initial position for safety reasons. The interface forces are measured using a custom-made force sensor. The coupled system exhibits nonlinear dynamics due to the nature of the spring assembly. If this setup is excited with one harmonic component, the response will contain higher harmonics. The presence of these higher harmonics, in general, requires the enrichment of the basis function space with higher harmonics. This means that the frequencies Ω_k of the harmonics in the basis function matrix $\boldsymbol{W}(t)$ are defined as multiples of the periodic excitation basis frequency Ω_0:

$$\Omega_k = k\Omega_0 \qquad \text{with} \qquad k \in [1, 2, ..., n_\Omega]$$

Virtual Component (VIR)	
m^{VIR}	$1\,\mathrm{kg}$
d^{VIR}	$10\,\frac{\mathrm{N \cdot s}}{\mathrm{m}}$
k^{VIR}	$1000\,\frac{\mathrm{N}}{\mathrm{m}}$

Experimental Component (EXP)	
k^{EXP}	$1.95\,\frac{\mathrm{N}}{\mathrm{m}}$
k_3^{EXP}	$0.0014\,\frac{\mathrm{N}}{\mathrm{m}^3}$

Table 6.2: System parameters of the experimental setup

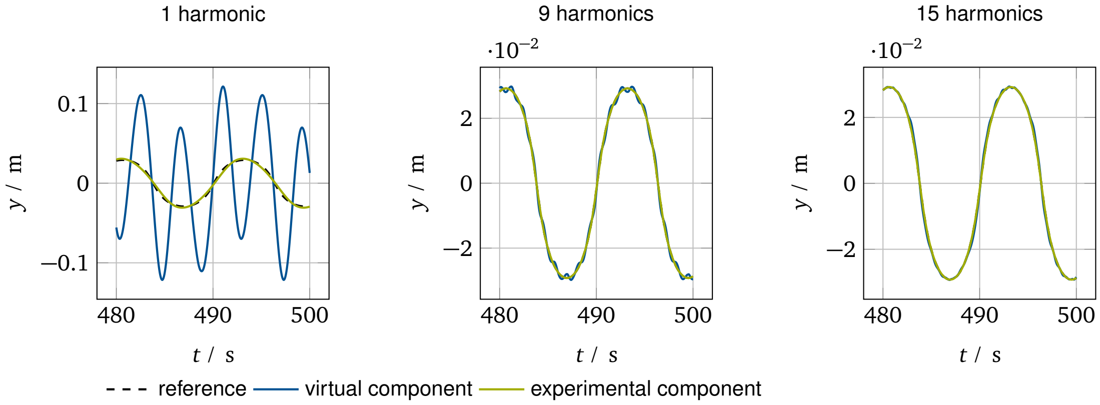


Figure 6.13: Simulation results for a varying number of harmonics in the basis functionspace of $\boldsymbol{W}(t)$. The excitation frequency $0.5\ \mathrm{Hz}$ was chosen such that the nonlinear behavior can be observed. In the first plot, the higher harmonics in the virtual components are excited through the interface forces, but the higher harmonic interface displacements are not synchronized.

Fig. 6.13 shows the effect of the additional harmonics in the basis function matrix of the simulated system. In the first plot (with $n_\Omega = 1$), the higher harmonics in the virtual component are excited through the interface forces, but the higher harmonic interface displacements are not synchronized. The adaptive feed-forward approach applies to any number of harmonics in the basis function matrix. While the system with only one harmonic synchronizes the interfaces only insufficiently, the systems with higher harmonics show improved results. The plots with $n_\Omega = 9$ and with $n_\Omega = 15$ harmonic basis functions demonstrate that the adaptive feedforward can be used to couple non-linear systems. However, for the excitation frequencies and amplitudes described in the following sections, one harmonic basis function is sufficient to couple the virtual and the experimental component satisfactorily.

6.2.1 Passivity Preserving Adaptive Feed-Forward Filters

The following section refers to the validation of the passivity preserving adaptive feed-forward filters which were introduced in Sec. 5.3. The objective of the method is to ensure stable testing independent of the choice of the adaptation gain μ_{LMS}.

The test is performed using the constant parameters $\Delta_\downarrow a_\mu$, $\Delta_\downarrow a_\gamma$, $\Delta_\uparrow a_\gamma$, b_μ, b_γ, γ_{max} and P_{lim} from Tab. 6.3. As mentioned above, one harmonic was used in the basis function matrix $\boldsymbol{W}(t)$. The excitation frequency Ω_0 is varied. The excitation amplitude was adjusted to the excitation frequency because the resulting response amplitudes had to remain within the actuator workspace. The initial adaptation gain μ_{init} was selected such that the resulting adaptation process is unstable without the proposed algorithm. Since the adaptive feed-forward filter is exact only at the excitation frequencies, measurement noise can cause a drift of the interface gap. A peak filter using the excitation frequency was applied at the actuator input in order to prevent those drift effects. Because the objective is to validate the performance

Variable		Values		
step size	$\Delta_{\downarrow} a_{\mu}$		0.001	
initial adaptation gain	μ_{init}		0.1	
exponent	b_{μ}		2	
step size	$\Delta_{\downarrow} a_{\gamma}$		0.001	
step size	$\Delta_{\uparrow} a_{\gamma}$		0.01	
initial regularization factor	γ_{max}		1	
exponent	b_{γ}		2	
power-generation limit	P_{lim}		$-0.3\,\mathrm{W}$	
excitation amplitude	A_{ext}	$10\,\mathrm{N}$	$40\,\mathrm{N}$	$40\,\mathrm{N}$
excitation frequency	f_{ext}	$10\,\mathrm{Hz}$	$20\,\mathrm{Hz}$	$30\,\mathrm{Hz}$

Table 6.3: Parameters used in the experiment

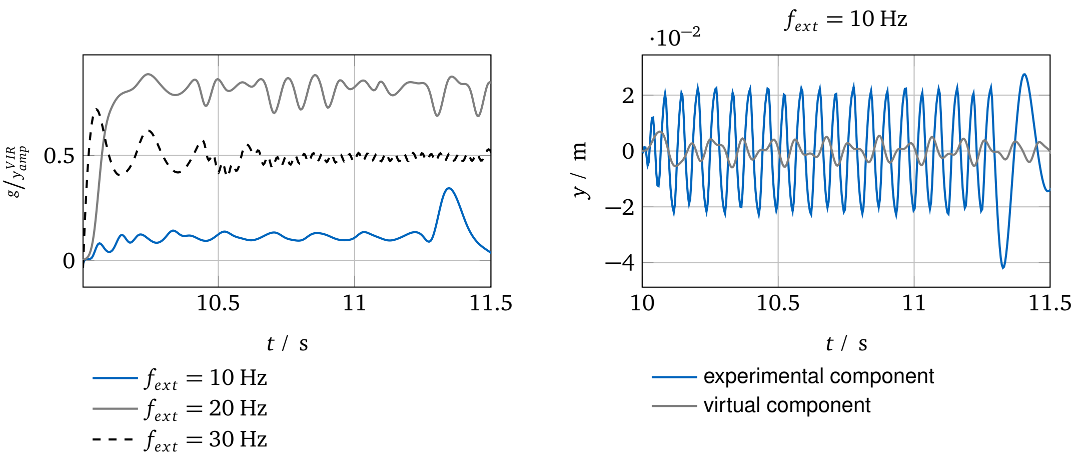


Figure 6.14: Unstable power and energy-inflow to actuator system without power-flow supervision.

of the power-flow supervision, the test for each frequency was performed in two modes: one using power-flow supervision and one using the pure adaptive feed-forward control law without power-flow supervision. In the first experiment, the LMS-based adaptive feed-forward filter approach is applied without power-flow-based stabilization. As a result, the adaptation gain μ_{LMS} and the leakage factor ν_{LMS} stay constant throughout the test. The adaptation gain μ_{LMS} was selected such that the dynamics of the filter are expected to be unstable. Fig. 6.14 shows the interface gap as well as the time-domain synchronization plots. In all cases, unstable filter dynamics can be observed. The actuator operates in a state of saturation and the responses show undefined peaks. Fig. 6.15 exhibits the high power- and energy-outflow from the actuator system caused by the instability.

The power-flow-based stabilization algorithm is activated in the second experiment. The adaptation gains μ_{LMS} in Fig. 6.16 correspondingly drop to a value which allows the stable operation of the filter. The leakage factor ν_{LMS} also drops to values of 0 in the phases where the power-flow constraint is violated. Fig. 6.17 shows the power and energy-outflow due to the initially unstable behavior. The maximum power-outflow is constrained to approximately 3 W. In the energy-outflow plot, a slight energy-outflow can be observed after the stabilization of the test. The reasons for this effect are stick-slip friction effects which are not compensated for by the actuator control. The remaining interface gap may cause

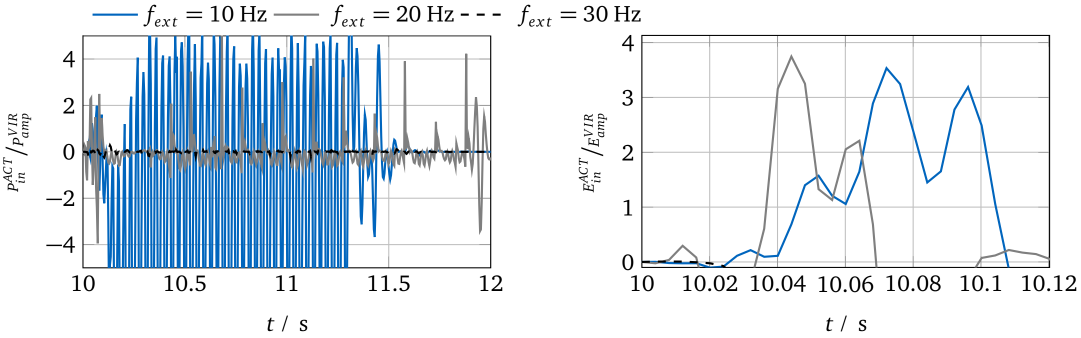


Figure 6.15: Unstable learning curve and interface synchronization without power-flow supervision.

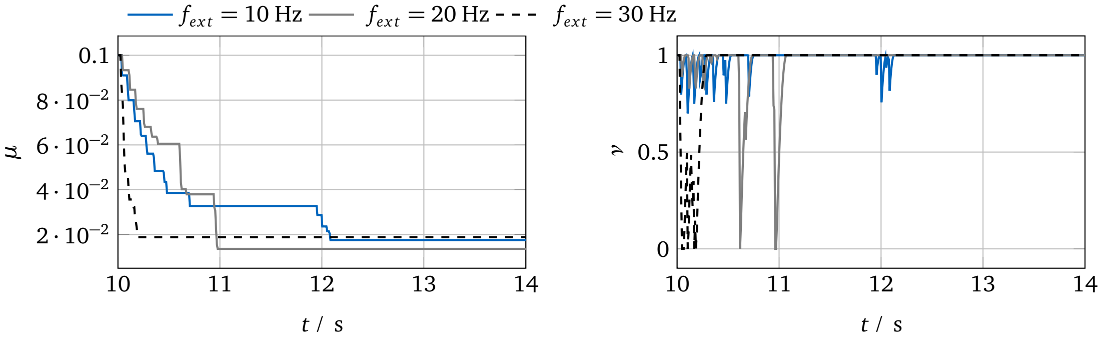


Figure 6.16: Adaptation gain and leakage factor with activated power-flow supervision

the energy-outflow which does not result in unstable behavior. Finally, learning curve and time-domain synchronization plots in Fig. 6.18 show the stabilization effect of the proposed algorithm: Even though displacement peaks initially occur, the system is stabilized after a timespan of less than 0.5 s. After the stabilization, the adaptation continues and results in synchronization between the virtual and the experimental component. Note that the displacement peaks can be prevented in a practical application by the application of a peak or comb filter to the actuator input.

To summarize, the proposed approach enables the stabilization of an initially unstable test with a nonlinear spring. After stabilization, the adaptation gain settles, and the system finally reaches interface synchronization. The method has been applied to an experimental test case which coupled a physical cubic spring with a virtual mass-spring-damper system. The results showed that the proposed method helped to stabilize the filter with initially unstable filter behavior. The high adaptation gain causes instability. The adaptation gain is updated as a reaction to the power-outflow from the actuator system and settles to a positive value. This allows the filter coefficient to converge such that the interface is synchro-

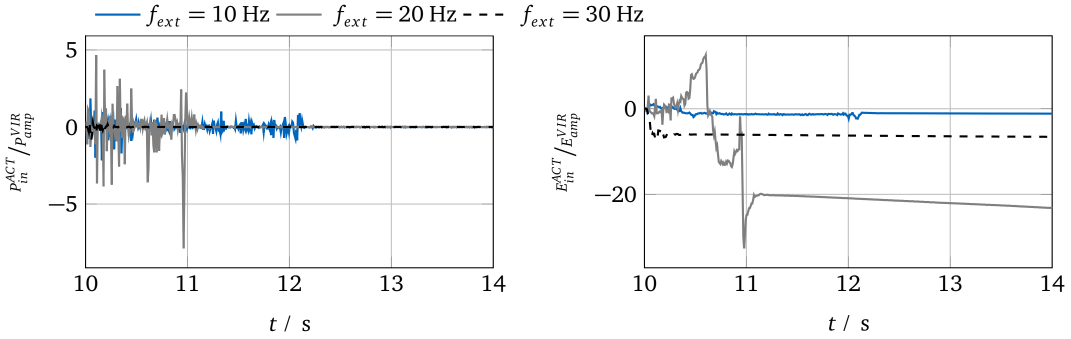


Figure 6.17: Power and energy-inflow to actuator system with activated power-flow supervision.

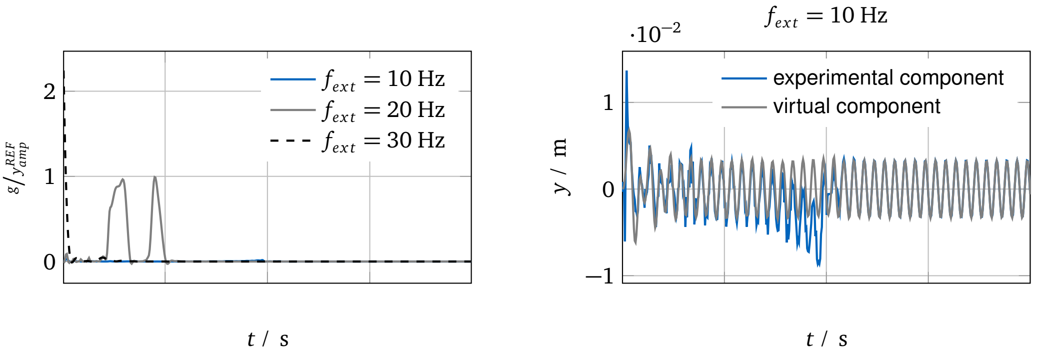


Figure 6.18: Learning curve and interface synchronization with activated power-flow supervision.

nized. In other cases where adaptation with the chosen filter parameters is impossible such as for a high phase error of P_{gu}, the passivity constraint is maintained and the adaptation gain μ_{LMS} is taken down to zero. This means that the adaptation has failed, but possibly damaging behavior is prevented.

6.2.2 Combined Feed-Back and Feed-Forward Control

A method which combines adaptive feed-forward filters with passivity-based control is given in Sec. 5.6. In order to evaluate the method, it was applied to the hybrid test with a cubic spring which was described in Sec. 6.2. The virtual component was excited with three different sinusoidal excitation signals of the form $f_{ext} = A_0 \cos(\Omega_0 t)$:

- $A_0 = 10$ N and $\Omega_0 = 10$ Hz
- $A_0 = 40$ N and $\Omega_0 = 20$ Hz
- $A_0 = 40$ N and $\Omega_0 = 30$ Hz

Different amplitudes for different excitation signals were chosen due to workspace restrictions and signal-to-noise ratio of the sensor signals.
Fig. 6.19 shows the subspace synchronization plots for three different control strategies:

- a pure passivity-based control as it is described in Sec. 5.2
- an adaptive feed-forward filter as it is described in Sec. 4.4 and
- a combination of both control strategies as it is described in Sec. 5.6

In general, synchronization plots show the relation of the interface displacements of two components in one diagram. An undeformed line with slope 1, which crosses the point $y^{EXP} = 0$ / $y^{VIR} = 0$ indicates perfect synchronization. An ellipsoid structure of the synchronization curve signals a phase shift between the interface displacements. If the slope differs from 1, an amplitude error exists. If the curve does not cross the point $y^{EXP} = 0$ / $y^{VIR} = 0$, an position-offset between the two interface displacements is present. The passivity-based controller exhibits a strong phase error at all frequencies. The pure adaptive feed-forward filter shows aligned phases and amplitudes. However, a significant offset can be found. The reason is that the approach only accounts for the harmonics from the excitation signal. Sensor noise can introduce drift effects. The method which combines the two approaches shows the alignment of phase and amplitude and no offsets. Fig. 6.20 shows the tracking of the reference system dynamics. The real-time hybrid test with combined adaptive-feed forward filter and passivity-based control exhibit synchronized interfaces and shows agreement with the reference system. The reference tracking, however, is not perfect. This is due to problems with the identification of the reference

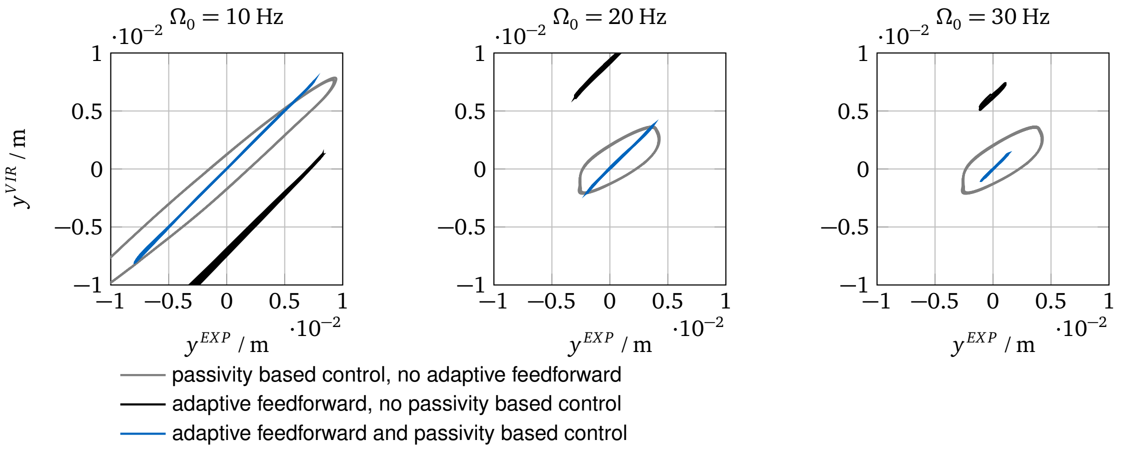


Figure 6.19: Interface synchronization with adaptive feed-forward filters and passivity-based control.

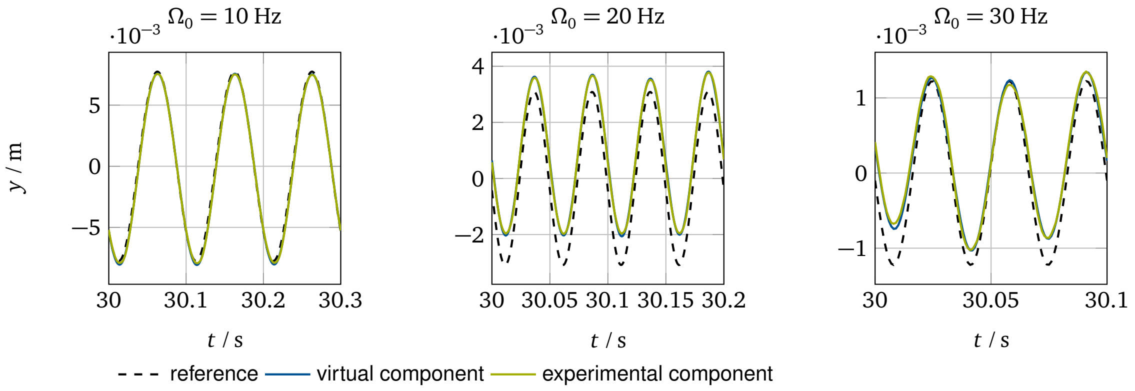


Figure 6.20: Interface synchronization with adaptive feed-forward filters and passivity based control.

system.
In summary, the combined adaptive feed-forward filter with passivity based control offers the opportunity to couple subcomponents with harmonic excitations accurately, stably and without an offset error.

Chapter 7

Closure

Parts of this chapter have been submitted for publication in [7] and [9].

In this thesis, adaptive feedforward filters have been successfully applied to real-time hybrid tests. The following chapter concludes the thesis and discusses suggestions for future research directions.

7.1 Conclusions

Time discrete filters can represent the virtual component Finite impulse response filters are non-recursive time discrete filters which are unconditionally stable. It was shown that the number of filter coefficients depends solely on the length which is determined by the damping in the structure. In contrast, the computational effort of time integration schemes depends on the size of the system. More specifically, it depends on the modal density and the bandwidth of the problem, if reduction methods are used. The computational effort caused by finite impulse response filters is not affected by the modal density. The results indicate that in cases where a relatively short filter can represent the impulse responses, the use of finite impulse response filters reduces the computational effort significantly. This fact is shown with the help of an example in Sec. 2.4.

Interface forces can be estimated using models if no sensors are placed directly at the interface. The use of transmission simulators is proposed. Transmission simulators replicate the dynamics of an interface. They are necessary if the sensor- and actuator-interfaces cause significantly different stiffness or friction properties compared to the emulated system. If the dynamic effects of the transmission simulator are negligible, the problem can be solved using projection. The rotations are coupled but translations are measured, which is similar to the virtual point approach from [93]. If inertia effects as well as stiffness and damping are present, they have to be compensated using models of the transmission simulator. Sec. 3.2 shows that augmented-state Kalman filters can significantly smoothen the resulting interface forces and interface displacements.

Many representations of real-time hybrid testing methods are equivalent and frameworks can help to categorize them. In literature, many methods exist which help to solve control problems related to coupling virtual and experimental components. The coupling problem is discussed in Sec. 4.1. All available methods compensate for delays and actuator dynamics. They can be structured in frameworks such as the ones described by [79] or [102]. The control schemes can include feedback control and feedforward control. The analysis shows that the polynomial forward prediction method (see Sec. 4.3)—which is widely used in literature—is a lead compensator which adapts to the specific frequency of excitation.

Feed-forward filters couple virtual and experimental component without deteriorating the stability of the test. As discussed in Sec. 4.2, feedforward filters generate actuator input based on knowledge of the system. If the system is unknown, adaptive feedforward filters offer superior properties. Finite impulse response based fxLMS filters can be used if transient dynamics are relevant. If steady-state and periodic excitation can be assumed, harmonic basis functions are more efficient than filters based on finite impulse responses.

Adaptive feedforward filters can be used to couple systems with multiple DOF interfaces Harmonic basis functions are an efficient way to reduce the filter size. The algorithms are based on LMS-type adaptation laws and on RLS-type adaptation laws. Both types of algorithms require a prior identification phase and assume steady-state. The LMS-based algorithm (Sec. 5.2) is robust against errors in the identification since it will stay stable with a sufficiently small adaptation gain for phase errors of $\pm\pi/2$. LMS filters are a computationally efficient and easy to implement type of adaptive feedforward filters. 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 adaptation laws exhibit faster convergence (Sec. 5.4). Sequential identification and adaptation with identification errors can cause a remaining interface gap. The simultaneous identification and adaptation process can overcome identification errors during the adaptation phase. An aggressive choice of the forgetting factor, however, can cause the violation of the steady-state assumption, because 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—i.e. inertia and stiffness effects—of the interface using a model. The experiments on a beam test rig (Sec. 6.1) show that the methods can replicate a structure with a multiple DOF interface with standard testing equipment such 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.

Power flow supervision make the stability of tests with adaptive feedforward filters independent of identification errors. In Sec. 5.3, a method for adjusting the parameters of the adaptive feedforward filter based on power-flows in the test setup is proposed. The objective is to maintain a passive behavior of the actuation and control system. The stabilization acts on the leakage factor and the adaptation gain of a LMS adaptation law. A simple numerical system is used to investigate the effect of the algorithm parameters on the stabilization. The study of a simple purely numerical case leads to the following conclusions:

- Step size $\Delta_{\downarrow}a_{\mu}$ defines the convergence speed of the adaptation gain μ_{LMS}. High values can lead to lower values of μ_{LMS}—which, in turn, cause a slower adaptation.

- Step size $\Delta_{\uparrow}a_{\gamma}$ defines the convergence speed of the leakage factor ν_{LMS}. Higher values can reduce the peak values of power outflow but increase the settling time.

- The power-outflow limit P_{lim} has only a slight influence on the final value of the adaptation gain, but lower values of $|P_{lim}|$ reduce the peak interface gap.

In Sec. 6.2, the method has been applied to an experimental test case which coupled a physical cubic spring with a virtual mass-spring-damper system. The results show that the proposed method helps to stabilize the filter with initially unstable filter behavior. In cases where adaptation with the chosen filter parameters is impossible such as for an error in the identified dynamics, the passivity constraint is maintained, and the adaptation gain μ_{LMS} is taken down to zero. The vanishing of the adaptation gain implies that the adaptation has failed, but damaging behavior is prevented.

The combination of adaptive feedforward filters and passivity-based control helps to remove drift effects. Adaptive feedforward filters with harmonic basis functions can lead to transient drift effects since they act at the excitation frequencies. If additionally transient dynamics are supposed to be coupled, the feedforward control can be combined with a feedback control scheme. Because the feedback system has to be stable for adaptive feedforward filters to be applied, passivity-based control is used to enforce stability. The passivity-based control is, in fact, an adaptive damper element which acts on the virtual component. The damping rate is controlled by power outflow from the actuation system which is an indicator for a violation of the passivity requirement. As shown in Sec. 6.2, the approach removes drifts while enabling an accurate and stable coupling of the subcomponents.

7.2 Outlook

As a result of the work of this thesis, several research questions remain open and new questions are arising. Further ideas were generated during the work on the thesis. Some may be useful for defining future research directions:

Efficient computations The efficient computation of the response of the virtual component is a bottleneck. Model order reduction techniques are a way to bring down this computational effort. In future research, different existing model order reduction techniques can be compared. New methods can be developed which include roughly estimated models of the experimental component and help to improve the reduction basis.

Evaluating the quality of the test results It is possible to generate a reference output for simple systems in order to evaluate methods. This procedure is, however, cumbersome because an exact model of the experimental component has to be identified or the virtual component has to be built as a physical structure. The knowledge of the correct overall system behavior contradicts the hybrid testing paradigm. For industrial scale problems, it may be impossible to generate reference outputs. In practice, it is important to know confidence intervals for the response of the emulated system with respect to hybrid testing results. Uncertainty quantification offers a toolbox which is helpful in this field. One idea is to use simplified models to estimate the accuracy of the test. The statistical information can help to design controllers which yield results in a specified uncertainty range.

Contact problems Mechanical contacts occur in various fields of engineering: Examples include air-to-air refueling procedures, satellite docking maneuvers or prosthetic feet touching the ground. Contact problems are specific since the system properties change abruptly during the test. The jumps in the system properties are dependent on their states. Most research on contact problems in hybrid tests has been conducted on pseudo-dynamic tests so far. There is still the need for improved methods to deal with contact problems using real-time hybrid tests.

Learning algorithms The methods which were developed in this work focus on systems with periodic excitations. They exploit the properties of the excitations and learn from the responses of the structures. Proceeding with this idea leads to learning algorithms for more general applications. Batch wise learning algorithms such as iterative learning control could be applied to generalize the ideas for adaptive control proposed in this thesis. Adaptive model predictive control could be a solution for tests where the model structure is known, but model parameters are unknown. Those parameters can be estimated during the test and the resulting model of the experimental component can be used for further analysis. General non-linear effects could be addressed, e.g. using neural networks.

General-purpose software The ultimate vision for real-time hybrid testing is a middleware software which can be used with a wide range of actuators, sensors and simulation software. Ideally, this software works out-of-the-box with industrial robots or Stewart platforms and has an interface with popular simulation software. The requirement for real-time interfaces with simulation software and actuation systems remains problematic because the control software of robots and algorithms in simulation software are mainly closed-source. Learning algorithms can help to couple systems with repeating excitation patterns without intervening in the closed software platforms. On the hardware side, adapter concepts have to be developed which make it possible to adapt to the dynamic properties of the tests: E.g., springs or flexible structures can serve as an interface which allows precise force control if the actuator solely allows displacement control.

New application fields In many cases, real-time hybrid testing still is a highly application-specific technology which demands application-specific solutions. The design and the properties of the hybrid test is dependent on excitation, stiffness- and mass-ratios, the interface complexity, necessary actuation forces, stroke lengths as well as the properties of the virtual component. For this reason, it makes sense to develop also application-specific methods.
New application fields can emerge where the interface between the subcomponents is well defined and simple. Rotor systems can give a natural and straightforward interface which represents the rotational DOF. Examples of future applications of real-time hybrid testing is the test of drive-train components such as dual-mass flywheels and the investigation of problems with brake squeal. Another potential application field is the dynamics of cables. An example is hybrid tests with railway pantographs. The topic has already been addressed in research but remains an interesting research field since the fast simulation of overhead lines is challenging. The test of damper elements can be extended to a broader variety of industries. An example is suspension systems of mountain bikes. Finally, another exciting research topic is the design of real-time hybrid tests of prosthetic feet. Here, the main challenges are the modeling of human gait and control problems due to contact.

Bibliography

[1] Albu-Schäffer, A. and Ott, C. "A Unified Passivity- based Control Framework for Position , Torque and Impedance Control of Flexible Joint". In: *The International Journal of Robotics Research* 26.1 (2007), pp. 23–29. DOI: 10.1177/0278364907073776.

[2] Allen, M. S., Mayes, R. L., and Bergman, E. J. "Experimental modal substructuring to couple and uncouple substructures with flexible fixtures and multi-point connections". In: *Journal of Sound and Vibration* 329.23 (2010), pp. 4891–4906.

[3] Bartl, A., Mayet, J., and Rixen, D. "An adaptive approach to coupling vibration tests and simulation models with harmonic excitation". In: *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM* 2018-July (2018), pp. 262–267. DOI: 10.1109/AIM.2018.8452358.

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

[5] Bartl, A., Wernsen, M., Karamooz Mahdiabadi, M., and Rixen, D. J. "Interface State Estimation for Hardware-In-The-Loop Tests of Structural Dynamic Systems". In: *Proceedings of the 27th ISMA, A Conference on Noise and Vibration Engineering*. KU Leuven, 2016.

[6] Bartl, A., Insam, C., and Rixen, D. "Stability Issues in Hardware-in-the-Loop Tests of Flexible Components". In: *PAMM*. 2018.

[7] Bartl, A., Mahdiabadi, M. K., Insam, C., Mayet, J., and Rixen, D. J. "A Hybrid Testing Method Based on Adaptive Feed-Forward Filters". In: *Submitted to: Mechanical Systems and Signal Processing* (2018).

[8] Bartl, A., Mayet, J., Mahdiabadi, M. K., and Rixen, D. J. "Multi-DoF interface synchronization of real-time-hybrid-tests using a recursive-least-squares adaption law: A numerical evaluation". In: *Conference Proceedings of the Society for Experimental Mechanics Series* 4 (2016), pp. 7–14. ISSN: 21915652. DOI: 10.1007/978-3-319-29763-7_2.

[9] Bartl, A., Peiris, L. D. H., Bois, J. L., Rixen, D., and Plummer, A. "Power-flow-based Stabilization for Adaptive Feed-Forward Filters in Hybrid Testing". In: *In Submission Process: Experimental Techniques* (2018).

[10] Bartl, A. and Rixen, D. J. "Feasibility of a Transmission Simulator Technique for Dynamic Real Time Substructuring". In: *Dynamics of Coupled Structures, Volume 4: Proceedings of the 33rd IMAC, A Conference and Exposition on Structural Dynamics, 2015*. Springer. 2015, p. 41.

[11] Bathe, K.-J. *Finite element procedures*. Klasu-Jurgen Bathe, 2006.

[12] Bauchau, O. *Flexibe multibody dynamics*. Springer Science & Business Media, 2010.

[13] Bayer, V., Dorka, U., Füllekrug, U., and Gschwilm, J. "On real-time pseudo-dynamic sub-structure testing: algorithm, numerical and experimental results". In: *Aerospace Science and Technology* 9.3 (2005), pp. 223–232. ISSN: 12709638. DOI: 10.1016/j.ast.2005.01.009.

[14] Blakeborough, A., Darby, A., and Williams, M. "The Development of Real-Time Substructure Testing". In: *Philosophical transactions: Mathematical, Physical and Engineering Sciences* 359.1786 (2001).

[15] Bois, J. du, Titurus, B., and Lieven, N. “Transfer Dynamics Cancellation in Real-Time Dynamic Sub- structuring”. In: *Proceedings of ISMA 2010*. 2010, pp. 1891–1914.

[16] Bolien, M., Iravani, P., and Du Bois, J. “Toward robotic pseudodynamic testing for hybrid simulations of air-to-air refueling”. In: *IEEE/ASME Transactions on Mechatronics* 22.2 (2017), pp. 1004–1013. DOI: 10.1109/TMECH.2016.2644200.

[17] Bonnet, P. A., Lim, C. N., Williams, M. S., and Blakeborough, A. “Real-time hybrid experiments with Newmark integration , MCSmd outer-loop control and multi-tasking strategies”. In: *Earthquake Engineering & Structural Dynamics* 1.9 (2006), pp. 119–141. DOI: 10.1002/eqe.

[18] Bonnet, P., Williams, M., and Blakeborough, A. “Evaluation of numerical time-integration schemes for real-time hybrid testing”. In: *Earthquake Engng Struct. Dyn. 2008;* 37.May (2008), pp. 1467–1490. DOI: 10.1002/eqe.821.

[19] Braesch, I. “Application of Time-Discrete Filters to Hardware-in-the-Loop Systems”. Semester Thesis. Technical University of Munich, 2018.

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

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

[22] Carrion, J. E. and Spencer, B. F. “Model-based Strategies for Real-time Hybrid Testing”. In: December (2007), p. 211.

[23] Chabaud, V. “Real-time Hybrid Model Testing of Floating Wind Turbines”. Doctoral Thesis. Norwegian University of Science and Technology, 2016.

[24] Chabaud, V., Steen, S., and Skjetne, R. “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.

[25] Craig, R. and Bampton, M. “Coupling of Substructures for Dynamic Analyses”. In: 6.7 (1968), pp. 1313–1319.

[26] Craig, R. and Chang, C. *Substructure Coupling for Dynamic Analysis and Testing*. Tech. rep. 1977.

[27] Cryer, B., Nawrocki, P., and Lund, R. “A Road Simulation System for Heavy Duty Vehicles”. In: *SAE Transactions* 85 (1976), pp. 1322–1334.

[28] Darby, A., Blakeborough, A., and Williams, M. “Improved control algorithm for real-time substructure testing”. In: July 2000 (2001), pp. 431–448.

[29] Darby, A., Williams, M., and Blakeborough, A. “Stability and Delay Compensation for Real-Time Substructure Testing”. In: *Journal of Engineering Mechanics* 128.12 (2002), pp. 1276–1284.

[30] De Klerk, D., Rixen, D. J., and Voormeeren, S. N. “General Framework for Dynamic Substructuring: History, Review and Classification of Techniques”. In: *AIAA Journal* 46.5 (May 2008), pp. 1169–1181. ISSN: 0001-1452. DOI: 10.2514/1.33274.

[31] Derzemitzakis, S. and Mahin, S. A. *Development of Substructuring Techniques for on-Line Computer Controlled Seismic Performance Testing*. Tech. rep. Report No. UCB/EERC-85/04: Earth. Eng. Research Center, Univ. of California, Berkeley, 1985. DOI: 10.12681/eadd/4736.

[32] Dodds, C. and Plummer, A. “Laboratory Road Simulation for Full Vehicle Testing: A Review”. In: 2001. Jan. 2001, pp. 487–494. DOI: 10.4271/2001-26-0047.

[33] Facchinetti, A. and Bruni, S. “Hardware-in-the-loop hybrid simulation of pantographcatenary interaction”. In: *Journal of Sound and Vibration* 331.12 (2012), pp. 2783–2797. DOI: 10.1016/j.jsv.2012.01.033.

[34] Gene, F., Powell, J., and Emami-Naeini, A. *Feedback Control of Dynamic Systems*. Reading: Addison-Wesley, 1994.

[35] Géradin, M. and Rixen, D. *Mechanical Vibrations*. Chichester: John Wileys & Sons, 2015.

[36] Géradin, M. and Rixen, D. J. “Impulse-based substructuring in a floating frame to simulate high frequency dynamics in flexible multibody dynamics”. In: *Multibody System Dynamics* (2017), pp. 1–31. ISSN: 1573272X. DOI: 10.1007/s11044-017-9597-0.

[37] Gordis, J. H. “Integral equation formulation for transient structural synthesis”. In: *AIAA Journal* 33.2 (1995), pp. 320–324. ISSN: 0001-1452. DOI: 10.2514/3.12454.

[38] Gough, V. and Whitehall, S. “Universal tire test machine”. In: *Proceedings 9th Int. Technical Congress F.I.S.I.T.A.* 1962, pp. 117–135.

[39] Guyan, R. “Reduction of stiffness and mass matrices”. In: *AIAA Journal* 3.2 (1965), pp. 380–280.

[40] Hannaford, B., Member, S., and Ryu, J.-h. “Time-Domain Passivity Control of Haptic Interfaces”. In: 18.1 (2002), pp. 1–10.

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

[42] Hilber, H. M., Hughes, T. J., and Taylor, R. L. “Improved numerical dissipation for time integration algorithms in structural dynamics”. In: *Earthquake Engineering & Structural Dynamics* 5 (1977), pp. 283–292.

[43] Hirschmann, H. “Kinematische und regelungstechnische Konzeption einer Aktorregelung für strukturdynamische Hardware-in-the-Loop Tests”. Semester Thesis. Technical University of Munich, 2016.

[44] Hogan, N. and Buerger, S. “Impedance and Interaction Control”. In: *Robotics and automation handbook*. 2005, p. 19. ISBN: 0849318041.

[45] Horiuchi, T., Inoue, M., Konno, T., and Namita, Y. “Real-time Hybrid Experimental System with Actuator Delay Compensation and its Application to a Piping System with Energy Absorber.” In: *Earthquake Engineering and Structural Dynamics* 28.10 (1999), pp. 1121–1141.

[46] Hosseini, S. “Interface Synchronization of Substructured Systems via Unsupervised Learning”. Semester Thesis. Technical University of Munich, 2017.

[47] Hughes, T. *The finite element method: linear static and dynamic finite element analysis*. Courier Corporation, 2012.

[48] Insam, C., Bartl, A., and Rixen, D. “A Step Towards Testing of Foot Prosthesis using Real-Time Substructuring (RTS)”. In: *IMAC - Conference & Exposition on Structural Dynamics*. 2019.

[49] Ioannou, P. and Sun, J. *Robust adaptive control*. Dover Publications, 2013.

[50] Isermann, R. *Mechatronic Systems*. Springer London, 2005. ISBN: 1852339306.

[51] Isermann, R., Schaffnit, J., and Sinsel, S. “Hardware-in-the-Loop Simulation for the Design and Testing of Engine-Control Systems”. In: *IFAC Proceedings Volumes* 31.4 (1998), pp. 1–10. DOI: 10.1016/S1474-6670(17)42125-2.

[52] Jetmundsen, B., Bielawa, R., and Flannelly, W. “Generalized Frequency Domain Substructure Synthesis”. In: *Journal of the American Helicopter Society* 33.1 (1988), pp. 55–65.

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

[54] Kalman, R. E. “A New Approach to Linear Filtering and Prediction Problems”. In: *Journal of Basic Engineering* 82.1 (1960). ISSN: 00219223. DOI: 10.1115/1.3662552.

[55] Kamenetsky, M. and Widrow, B. “A Variable Leaky LMS Adaptive Algorithm”. In: *Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, 2004*. Vol. 1. 650. 2004, pp. 125–128. ISBN: 0780386221.

[56] Kammermeier, B. “Implementierung einer Aktorregelung für Dynamische Echtzeit-Substrukturierung”. Bachelor Thesis. Technical University of Munich, 2014.

[57] Kuo, S. and Morgan, D. *Active Noise Control Systems*. John Wileys & Sons, 1996.

[58] Kyrychko, Y. N., Blyuss, K. B., Gonzalez-Buelga, A., Hogan, S. J., Wagg, D. J., and Gonzalez-Buelga, A. “Real-time dynamic substructuring in a coupled oscillator–pendulum system”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 462.2068 (Apr. 2006), pp. 1271–1294. ISSN: 1364-5021. DOI: 10.1098/rspa.2005.1624.

[59] Lang, G. and Snyder, D. “Understanding the Physics of Electrodynamic Shaker Performance”. In: *Sound and Vibration* October (2001), pp. 1–10.

[60] Lawrence, D. A. *Position Based Impedance Control Stabilty*. 1987.

[61] Li, G., Na, J., Stoten, D., and Ren, X. “Adaptive neural network feedforward control for dynamically substructured systems”. In: *IEEE Transactions on Control Systems Technology* 22.3 (2014), pp. 944–954. DOI: 10.1109/TCST.2013.2271036.

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

[63] Macdonald, H., Green, T., and Williams, B. “Analysis and control of a moving coil electrodynamic actuator”. In: *Proceedings of IECON ’93 - 19th Annual Conference of IEEE Industrial Electronics* 3 (1993), pp. 2184–2189. DOI: 10.1109/IECON.1993.339415.

[64] MacNeal, R. “Hybrid Method of Component Mode Synthesis”. In: *Computers and Structures* 1.4 (1971), pp. 581–601.

[65] Madisetti, V. *Digital Signal Processing Fundamentals*. Vol. 20094251. Electrical Engineering Handbook. Bosa Roca: Taylor & Francis, Nov. 2009. ISBN: 978-1-4200-4606-9.

[66] Merlet, J. *Parallel Robots*. Dordrecht: Springer, 2006. ISBN: 9781402041327.

[67] Meyer, C. H. “Echtzeitfähige FIR-Modelle für Real-time Substructuring”. Semester Thesis. Technical University of Munich, 2015.

[68] Nakashima, M., Kato, H., and Takaoka, E. “Development of Real-Time Pseudo Dynamic Testing”. In: *Earthquake Engineering & Structural Dynamics* (1992), pp. 79–92.

[69] Newman, W. “Stability and Performance Limits of Interaction Controllers”. In: *Journal of Dynamic Systems, Measurement and Control* 114.583 (1992). DOI: 10.1109/TIE.2015.2448513.

[70] Newmark, N. “A method of computation for structural dynamics”. In: *American Society of Civil Engineers, Journal of the Engineering Mechanics Devision* 85.3 (1559), pp. 67–94.

[71] Niemeyer, G. and Slotine, J. “Stable Adaptive Teleoperation”. In: *IEEE Journal of Oceanic Engineering* 16.I (1991).

[72] Ogata, K. *Modern Control engineering*. Englewood Cliffs, N.J: Prentice-Hall, 1970.

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

[74] Paschedag, J. “Dämpfung der Motorvibrationen im Kfz mittels Filtered-x-LMS-Algorithmus: Implementierung am und Konvergenzanalyse”. In: *Technical Reports on Automatic Control* 2.June (2007).

[75] Pasma, E., Seijs, M. van der, Klaassen, S., and Kooij, M. van der. “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*. 2018.

[76] Pegon, P. and Pinto, A. V. “Pseudo-dynamic testing with substructuring at the ELSA laboratory”. In: *Earthquake Engineering and Structural Dynamics* 29.7 (2000), pp. 905–925. ISSN: 00988847. DOI: 10.1002/1096-9845(200007)29:7<905::AID-EQE941>3.0.CO;2-P.

[77] Peiffer, F. *Mechanical systems dynamics*. Springer Science & Business Media, 2008.

[78] Peiris, H., Plummer, A., and Bois, J. du. “Passivity Control in Real-time Hybrid Testing”. In: *2018 UKACC 12th International Conference on Control* (2018), pp. 317–322.

[79] Plummer, A. “Model-in-the-Loop Testing”. In: *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering* 220.3 (Jan. 2006), pp. 183–199. DOI: 10.1243/09596518JSCE207.

[80] Priesner, R. and Jakubek, S. “Mechanical impedance control of rotatory test beds”. In: *IEEE Transactions on Industrial Electronics* 61.11 (2014), pp. 6264–6274. ISSN: 02780046. DOI: 10.1109/TIE.2014.2308159.

[81] Rabiner, L. R. and Gold, B. *Theory and Application of Digital Signal Processing*. Bell Telephone Laboratories, Inc., 1975. ISBN: 0139141014. DOI: 10.1109/TASSP.1975.1162708. arXiv: 1001.2267.

[82] Reichart, R. “Online-Identifikation von zeitdiskkreten Modellen an einem Versuchstand für Echtzeit-Substrukturierung”. Bachelor Thesis. Technical University of Munich, 2014.

[83] Ricci, S., Peeters, B., Debille, J., Britte, L., and Faignet, E. “Virtual shaker testing: A novel approach for improving vibration test performance”. In: *Proceedings of the ISMA 2008* (2008), pp. 1767–1845.

[84] Riebe, S. and Ulbrich, H. “Modelling and online computation of the dynamics of parallel kinematic with six degrees-of-freedom”. In: *Archive of Appied Mechanics* 72 (2003), pp. 817–829.

[85] Riebe, S. “Aktive Schwingungsisolierung und Bahnregelung von Hexapodsystemen”. Doctoral Dissertation. Technical University of Munich, 2005.

[86] Rixen, D. “Substructuring using Impulse Response Functions for Impact Analysis”. In: *Structural Dynamics, Conference Proceedings of the Society for Experimental Mechanics, Series 12* 3 (2011), pp. 637–646. ISSN: 21915644. DOI: 10.1007/978-1-4419-9834-7.

[87] Rixen, D. J. “A Substructuring Technique Based on Measured and Computed Impulse Response Functions of Components”. In: *ISMA*. 2010.

[88] Rixen, D. J. and Van Der Valk, P. L. C. “An impulse based substructuring approach for impact analysis and load case simulations”. In: *Journal of Sound and Vibration* 332.26 (Dec. 2013), pp. 7174–7190. ISSN: 0022460X. DOI: 10.1016/j.jsv.2013.08.004.

[89] Rixen, D. and Haghighat, N. “Truncating the Impulse Responses of Substructures to Speed Up the Impulse-based Substructuring”. In: *Proceedings of the SEM IMAC XXX Conference, January 30–February 2, 2012*. 2012.

[90] Rixen, D. “Dual Craig–Bampton Method for Dynamic Substructuring”. In: *Journal of Computational Mathematics* 168.1-2 (2004), pp. 383–391.

[91] Rubin, S. “Improved Component-Mode Representation for Structural Dynamic Analysis”. In: *AIAA Journal* 13.8 (1975), pp. 995–1006.

[92] Sauder, T., Chabaud, V., Thys, M., Bachynski, E., and Sæther, L. O. “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.

[93] Seijs, M. van der, Bosch, D. van den, Rixen, D., and Klerk, D. de. “An Improved Methodology for the Virtual Point Transformation of Measured Frequency Response Functions in Dynamic Substructuring”. In: *COMPDYN 2013*. Vol. 2013. June. 2013, pp. 12–14. DOI: 10.13140/RG.2.1.2715.3126.

[94] Seijs, M. van der and Rixen, D. J. “Efficient impulse based substructuring using truncated impulse response functions and mode superposition”. In: *Proceedings of the International Conference on Noise and Vibration Engineering ISMA 2012* 9 (2012), pp. 3487–3499.

[95] Shabana, A. *Dynamics of Multibody Systems*. Cambridge University Press, 1998.

[96] Shing, B. “Integration schmes for real-time hybrid testing”. In: *Hybrid Simulation: Theory, Implmentation and Applications*. Ed. by Saouma, V. and Sivaselvan, M. Leiden: Taylor & Francis/Balkema, 2008.

[97] Shing, P. and Mahin, S. A. “Elimination of spurious higher-mode response in pseudodynamic tests”. In: *Earthquake Engineering and Structural Dynamics* 15 (1987), pp. 425–445.

[98] Slater, H., Atkinson, D., and Jack, A. “Real-time emulation for power equipment development. II. The virtual machine”. In: *Electric Power Applications, IEE Proceedings -* 145.3 (1998), pp. 153–158. DOI: 10.1049/ip-epa:19981849.

[99] Steurer, M., Edrington, C., Sloderbeck, M., Ren, W., and Langston, J. “A megawatt-scale power hardware-in-the-loop simulation setup for motor drives”. In: *IEEE Transactions on Industrial Electronics* 57.4 (2010), pp. 1254–1260. DOI: 10.1109/TIE.2009.2036639.

[100] Stewart, D. “A Platform with Six Degrees of Freedom”. In: *Aircraft Engineering and Aerospace Technology* 38.4 (1966), pp. 30–35.

[101] Stoten, D., Li, G., and Tu, J. “Model predictive control of dynamically substructured systems with application to a servohydraulically actuated mechanical plant”. In: *IET Control Theory & Applications* 4.2 (2010), pp. 253–264.

[102] Stoten, D., Tu, J., and Li, G. “Synthesis and control of generalized dynamically substructured systems”. In: *Systems and Control Engineering* 223 (2010), pp. 371–392.

[103] Takanashi, K., Udagawa, K., Seki, M., Okada, T., and Tanaka, H. “Nonlinear earthquake response analysis of structures by a computer-actuator on-line system”. In: *Trans. Architectural Institute of Japan* 229 (1975), pp. 77–83.

[104] Terkovics, N., Neild, S. A., Lowenberg, M., Szalai, R., and Krauskopf, B. “Substructurability: The effect of interface location on a real-time dynamic substructuring test”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 472.2192 (2016). ISSN: 14712946. DOI: 10.1098/rspa.2016.0433.

[105] Treis, M. “Dynamikmodellierung eines Hexapods”. Semester Thesis. Technical University of Munich, 2018.

[106] Tu, J., Stoten, D., Hyde, R., and Li, G. “A state-space approach for the control of multivariable dynamically substructured systems”. In: *Journal of Systems and Control engineering* 225.7 (2011), pp. 935–953.

[107] Voormeeren, S. N. “Dynamic Substructuring Methodologies for Integrated Dynamic Analysis of Wind Turbines door”. Doctoral Thesis. Technical University of Delft, 2012. ISBN: 9789491104107.

[108] Wagg, D. J. and Stoten, D. P. "Substructuring of dynamical systems via the adaptive minimal control synthesis algorithm". In: *Earthquake Engineering & Structural Dynamics* 30.6 (2001), pp. 865–877.

[109] Wallace, M., Wagg, D., and Neild, S. "An adaptive polynomial based forward prediction algorithm for multi-actuator real-time dynamic substructuring". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 461.2064 (Dec. 2005), pp. 3807–3826. DOI: 10.1098/rspa.2005.1532.

[110] Zeng, G. and Hemami, A. "An overview of robot force control". In: *Robotica* 15.5 (1997). DOI: 10.1017/S026357479700057X.

[111] Zhang, K. "Schnittstellen-Synchronisierung von Harware-In-the-Loop Tests mit strukturdynamischen Komponenten". Master Thesis. Technical University of Munich, 2016.

[112] Zhang, W., Mei, G., Wu, X., and Shen, Z. "Hybrid Simulation of Dynamics for the Pantograph-Catenary System". In: *Vehicle System Dynamics* 38.6 (2010), pp. 393–414.

[113] Zienkiewicz, O. and Taylor, R. *The finite element method*. London: MacGraw-Hill, 1977.

Publications by the Author

Bartl, A., Mayet, J., and Rixen, D. "An adaptive approach to coupling vibration tests and simulation models with harmonic excitation". In: *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM* 2018-July (2018), pp. 262–267. DOI: 10.1109/AIM.2018.8452358.

Bartl, A., Mayet, J., and Rixen, D. "Adaptive Feedforward Compensation for Real Time Hybrid Testing with Harmonic Exciation." In: *Proceedings of the 11th International Conference on Engineering Vibration* September (2015), pp. 7–10.

Bartl, A., Wernsen, M., Karamooz Mahdiabadi, M., and Rixen, D. J. "Interface State Estimation for Hardware-In-The-Loop Tests of Structural Dynamic Systems". In: *Proceedings of the 27th ISMA, A Conference on Noise and Vibration Engineering*. KU Leuven, 2016.

Bartl, A., Insam, C., and Rixen, D. "Stability Issues in Hardware-in-the-Loop Tests of Flexible Components". In: *PAMM*. 2018.

Bartl, A., Mahdiabadi, M. K., Insam, C., Mayet, J., and Rixen, D. J. "A Hybrid Testing Method Based on Adaptive Feed-Forward Filters". In: *Submitted to: Mechanical Systems and Signal Processing* (2018).

Bartl, A., Mayet, J., Mahdiabadi, M. K., and Rixen, D. J. "Multi-DoF interface synchronization of real-time-hybrid-tests using a recursive-least-squares adaption law: A numerical evaluation". In: *Conference Proceedings of the Society for Experimental Mechanics Series* 4 (2016), pp. 7–14. ISSN: 21915652. DOI: 10.1007/978-3-319-29763-7_2.

Bartl, A., Peiris, L. D. H., Bois, J. L., Rixen, D., and Plummer, A. "Power-flow-based Stabilization for Adaptive Feed-Forward Filters in Hybrid Testing". In: *In Submission Process: Experimental Techniques* (2018).

Bartl, A. and Rixen, D. J. "Feasibility of a Transmission Simulator Technique for Dynamic Real Time Substructuring". In: *Dynamics of Coupled Structures, Volume 4: Proceedings of the 33rd IMAC, A Conference and Exposition on Structural Dynamics, 2015*. Springer. 2015, p. 41.

Insam, C., Bartl, A., and Rixen, D. "A Step Towards Testing of Foot Prosthesis using Real-Time Substructuring (RTS)". In: *IMAC - Conference & Exposition on Structural Dynamics*. 2019.

Supervised Student Theses

Braesch, I. “Application of Time-Discrete Filters to Hardware-in-the-Loop Systems”. Semester Thesis. Technical University of Munich, 2018.

Hirschmann, H. “Kinematische und regelungstechnische Konzeption einer Aktorregelung für strukturdynamische Hardware-in-the-Loop Tests”. Semester Thesis. Technical University of Munich, 2016.

Hosseini, S. “Interface Synchronization of Substructured Systems via Unsupervised Learning”. Semester Thesis. Technical University of Munich, 2017.

Kammermeier, B. “Implementierung einer Aktorregelung für Dynamische Echtzeit-Substrukturierung”. Bachelor Thesis. Technical University of Munich, 2014.

Meyer, C. H. “Echtzeitfähige FIR-Modelle für Real-time Substructuring”. Semester Thesis. Technical University of Munich, 2015.

Reichart, R. “Online-Identifikation von zeitdiskkreten Modellen an einem Versuchstand für Echtzeit-Substrukturierung”. Bachelor Thesis. Technical University of Munich, 2014.

Treis, M. “Dynamikmodellierung eines Hexapods”. Semester Thesis. Technical University of Munich, 2018.

Zhang, K. “Schnittstellen-Synchronisierung von Harware-In-the-Loop Tests mit strukturdynamischen Komponenten”. Master Thesis. Technical University of Munich, 2016.

Appendix A

Derivation of IIR-Filters

The follwoing derivations refer to the time discrete filters in Sec. 2.4. For each damping case three steps are taken, namely the inverse Laplace transformation, discretization and the z-transformation.

A.1 Derivation for $\omega_i = 0$

Inverse Laplace Transformation

$$h_i(t) = \mathcal{L}^{-1}\left\{\left(s^2 + 2\delta_i s\right)^{-1}\right\} = \sigma(t)\cdot\frac{1}{2\delta_i}\left(1 - \mathrm{e}^{-2\delta_i t}\right)$$

Discretization

$$h_i[n] = \sigma[n]\cdot\frac{1}{2\delta_i}\left(1 - \left(\mathrm{e}^{-2\delta_i \Delta t}\right)^n\right)$$

z-Transformation

$$h_i(z) = \frac{1}{2\delta_i}\left(\frac{1}{1 - z^{-1}} + \frac{-1}{1 - \mathrm{e}^{-2\delta_i \Delta t} z^{-1}}\right)$$

A.2 Derivation for $\omega_i > 0$ and $\frac{\delta_i}{\omega_i} < 1$

Inverse Laplace Transformation

$$h_i(t) = \mathcal{L}^{-1}\left\{\left(s^2 + 2\left(\frac{\delta_i}{\omega_i}\right)\omega_i s + \omega_i^2\right)^{-1}\right\} = \sigma(t)\cdot\frac{1}{\omega_k\sqrt{1-\left(\frac{\delta_i}{\omega_i}\right)^2}}\mathrm{e}^{-\delta t}\sin\left(\omega_k\sqrt{1-\left(\frac{\delta_i}{\omega_i}\right)^2}\,t\right)$$

Discretization

$$h_i[n] = \sigma[n]\cdot\frac{1}{\omega_k\sqrt{1-\left(\frac{\delta_i}{\omega_i}\right)^2}}\left(\mathrm{e}^{-\delta_i \Delta t}\right)^n \sin\left(\omega_k\sqrt{1-\left(\frac{\delta_i}{\omega_i}\right)^2}\,\Delta t\, n\right)$$

z-Transformation

$$h_i(z) = \frac{1}{\omega_k\sqrt{1-\left(\frac{\delta_i}{\omega_i}\right)^2}}\frac{\left(\mathrm{e}^{-\delta_i \Delta t}\sin(\omega_k\sqrt{1-\left(\frac{\delta_i}{\omega_i}\right)^2}\Delta t)\right)z^{-1}}{1-\left(2\,\mathrm{e}^{-\delta_i \Delta t}\cos(\omega_k\sqrt{1-\left(\frac{\delta_i}{\omega_i}\right)^2}\Delta t)\right)z^{-1} + \mathrm{e}^{-2\delta_i \Delta t}z^{-2}}$$

A.3 Derivation for $\omega_i > 0$ and $\frac{\delta_i}{\omega_i} = 1$

Inverse Laplace Transformation

$$h_i(t) = \mathcal{L}^{-1}\left\{\left(s^2 + 2\omega_k + \omega_k^2\right)^{-1}\right\} = \sigma(t) \cdot t\,\mathrm{e}^{-\omega_k t}$$

Discretization

$$h_i[n] = \sigma[n] \cdot \Delta t\, n \left(\mathrm{e}^{-\omega_i \Delta t}\right)^n$$

z-Transformation

$$h_i(z) = \Delta t \frac{\mathrm{e}^{-\omega_i \Delta t} z^{-1}}{(1 - \mathrm{e}^{-\omega_i \Delta t} z^{-1})^2} = \Delta t \frac{\mathrm{e}^{-\omega_i \Delta t} z^{-1}}{1 - 2\,\mathrm{e}^{-\omega_i \Delta t} z^{-1} + \mathrm{e}^{-2 omega_i \Delta t} z^{-2}}$$

A.4 Derivation for $\omega_i > 0$ and $\frac{\delta_i}{\omega_i} > 1$

Inverse Laplace Transformation

$$h_i(t) = \mathcal{L}^{-1}\left\{\left(s^2 + 2\delta_i\, s + \omega_i^2\right)^{-1}\right\} = \sigma(t) \cdot \frac{1}{2\,\omega_k \sqrt{\left(\frac{\delta_i}{\omega_i}\right)^2 - 1}} \mathrm{e}^{-\delta t} \left(\mathrm{e}^{\omega_k \sqrt{\left(\frac{\delta_i}{\omega_i}\right)^2 - 1 t}} - \mathrm{e}^{-\omega_k \sqrt{\left(\frac{\delta_i}{\omega_i}\right)^2 - 1 t}}\right)$$

Discretization

$$h_i[n] = \sigma[n] \cdot \frac{1}{2\,\omega_k \sqrt{\left(\frac{\delta_i}{\omega_i}\right)^2 - 1}} \cdot \left(\left(\mathrm{e}^{-\delta \Delta t + \omega_k \sqrt{\left(\frac{\delta_i}{\omega_i}\right)^2 - 1 \Delta t}}\right)^n - \left(\mathrm{e}^{-\delta \Delta t - \omega_k \sqrt{\left(\frac{\delta_i}{\omega_i}\right)^2 - 1 \Delta t}}\right)^n\right)$$

z-Transformation

$$h_i(z) = \frac{1}{2\,\omega_k \sqrt{\left(\frac{\delta_i}{\omega_i}\right)^2 - 1}} \left(\frac{1}{1 - \left(\mathrm{e}^{\Delta t(-\delta + \omega_k \sqrt{\left(\frac{\delta_i}{\omega_i}\right)^2 - 1})}\right) z^{-1}} + \frac{-1}{1 - \left(\mathrm{e}^{\Delta t(-\delta - \omega_k \sqrt{\left(\frac{\delta_i}{\omega_i}\right)^2 - 1})}\right) z^{-1}}\right)$$

A.5 IIR Coefficients

	b_1	a_1	a_2
rigid-body modes	$\frac{1-\exp(-\delta_i \Delta t)}{c_k}$	$(1+\exp(-\delta_i \Delta t))$	$\exp(-\delta_i \Delta t)$
underdamped modes	$\frac{\sin\left(\omega_i \sqrt{1-\left(\frac{\delta_i}{\omega_i}\right)^2} \Delta t\right) \exp(-\delta_i \Delta t)}{\omega_i \sqrt{1-\left(\frac{\delta_i}{\omega_i}\right)^2}}$	$-2\cos\left(\omega_i \sqrt{1-\left(\frac{\delta_i}{\omega_i}\right)^2} \Delta t\right) \exp(-\delta_i \Delta t)$	$\exp(-2\delta_i \Delta t)$
critically damped modes	$\Delta t \exp(-\omega_i \Delta t)$	$-2\exp(-\omega_i \Delta t)$	$\exp(-2\omega_i \Delta t)$
overdamped mode	$\frac{\sinh\left(\omega_i \sqrt{\left(\frac{\delta_i}{\omega_i}\right)^2-1} \Delta t\right) \exp(-\delta_i \Delta t)}{\omega_i \sqrt{\left(\frac{\delta_i}{\omega_i}\right)^2-1}}$	$-2\cosh\left(\omega_i \sqrt{\left(\frac{\delta_i}{\omega_i}\right)^2-1} \Delta t\right) \exp(-\delta_i \Delta t)$	$\exp(-2\delta_i \Delta t)$

Table A.1: Coefficients of IIR filters (adapted from [19])

Appendix B

Simulation Parameters

Virtual Component (VIR)		Experimental Component (EXP)		Actuator (ACT)	
m^{VIR}	$0.1\,\mathrm{kg}$	m^{EXP}	$0.01\,\mathrm{kg}$	m^{ACT}	$0.1\,\mathrm{kg}$
d^{VIR}	$0.05\,\frac{\mathrm{N \cdot s}}{\mathrm{m}}$	d^{EXP}	$0.05\,\frac{\mathrm{N \cdot s}}{\mathrm{m}}$	d^{ACT}	$1\,\frac{\mathrm{N \cdot s}}{\mathrm{m}}$
k^{VIR}	$1000\,\frac{\mathrm{N}}{\mathrm{m}}$	k^{EXP}	$1000\,\frac{\mathrm{N}}{\mathrm{m}}$	k^{ACT}	$100\,\frac{\mathrm{N}}{\mathrm{m}}$

Table B.1: System parameters used in the lumped mass system.

Transfer System ACT			Test Specimen (EXP)		
Dimensions	a	$0.05\,m$	Young's Modulus	E	$210 \cdot 10^9\,\frac{N}{m^2}$
	b	$0.01\,m$	Length	l	$0.5\,m$
	c	$0.1\,m$	Cross section width	w	$0.05\,m$
	d	$0.12\,m$	Cross section height	h	$0.008\,m$
Mass	m	$0.2\,kg$			
Moment of Inertia	Θ	$6.7333 \cdot 10^{-4}\,kg\,m^2$			

Table B.2: System parameters used in the numerical case study for interface force and state estimation in Sec. 3.2.

Variable		Values			
step-size	$\Delta_{\downarrow}a_{\mu}$	0.0001	0.0004	0.0007	0.001
initial adaptation gain	μ_{init}	10			
exponent	b_{μ}	10			
step-size	$\Delta_{\downarrow}a_{\gamma}$	0.001			
step-size	$\Delta_{\uparrow}a_{\gamma}$	0.01			
initial regularization factor	γ_{max}	1			
exponent	b_{γ}	2			
power-generation limit	P_{lim}	$-1\,\mathrm{W}$			
excitation amplitude	A_{ext}	$10\,\mathrm{N}$			
excitation frequency	f_{ext}	$50\,\mathrm{Hz}$			

Table B.3: Parameters for the numerical experiment different adaptation gain step-sizes $\Delta_{\downarrow}a_{\mu}$

Variable		Values			
step-size	$\Delta_{\downarrow}a_{\mu}$	0.0001			
initial adaptation gain	μ_{init}	10			
exponent	b_{μ}	10			
step-size	$\Delta_{\downarrow}a_{\gamma}$	0.001			
step-size	$\Delta_{\uparrow}a_{\gamma}$	0.001	0.004	0.007	0.01
initial regularization factor	γ_{max}	1			
exponent	b_{γ}	2			
power-generation limit	P_{lim}	$-1\,\mathrm{W}$			
excitation amplitude	A_{ext}	$10\,\mathrm{N}$			
excitation frequency	f_{ext}	$50\,\mathrm{Hz}$			

Table B.4: Parameters for the numerical experiment different leakage factor step-sizes $\Delta_{\uparrow}a_{\gamma}$

Variable		Values			
step-size	$\Delta_{\downarrow} a_{\mu}$	0.0001			
initial adaptation gain	μ_{init}	10			
exponent	b_{μ}	10			
step-size	$\Delta_{\downarrow} a_{\gamma}$	0.001			
step-size	$\Delta_{\uparrow} a_{\gamma}$	0.01			
initial regularization factor	γ_{max}	1			
exponent	b_{γ}	2			
power-generation limit	P_{lim}	$-0.1\,\mathrm{W}$	$-0.4\,\mathrm{W}$	$-0.7\,\mathrm{W}$	$-1\,\mathrm{W}$
excitation amplitude	A_{ext}	$10\,\mathrm{N}$			
excitation frequency	f_{ext}	$50\,\mathrm{Hz}$			

Table B.5: Parameters for the numerical experiment varying power-outflow limit P_{lim}

Appendix C

Stability Analysis of Feedback Systems

C.1 Non-Dimensional Form of Open-Loop Transfer Function with Actuator Dynamics

The following section refers to the derivation of the non-dimensional open-loop transfer function in Sec. 4.2.3. The transfer function of the assembly of actuator system and experimental component $H^{TR}(s)$, the transfer function of the virtual component $H^{VIR}(s)$, the transfer function of the experimental component $H^{EXP}(s)$, the transfer function of the coupled reference system $H^{REF}(s)$ and the controller $C(s)$ are given in Eqs. (C.1).

$$
\begin{aligned}
C(s) &= k^P \\
H^{TR}(s) &= \frac{\Lambda^{EXP}}{U} = \frac{m^{EXP}s^2 + d^{EXP}s + k^{EXP}}{(m^{ACT}+m^{EXP})s^2 + (d^{ACT}+d^{EXP})s + k^{ACT}+k^{EXP}} \\
&= \frac{m^{EXP}}{m^{ACT}+m^{EXP}} \cdot \frac{s^2 + 2\zeta^{EXP}\omega_0^{EXP}s + \omega_0^{EXP^2}}{s^2 + \frac{2\zeta^{ACT}\sqrt{k^{ACT}m^{ACT}} + 2\zeta^{EXP}\sqrt{k^{EXP}m^{EXP}}}{m^{ACT}+m^{EXP}}s + \omega_0^{TR^2}} \\
H^{VIR}(s) &= \frac{Y^{VIR}}{\Lambda^{EXP}} = \frac{Y^{VIR}}{F_{ext}} = \frac{1}{m^{VIR}s^2 + d^{VIR}s + k^{VIR}} \\
&= \frac{1}{m^{VIR}} \cdot \frac{1}{s^2 + 2\zeta^{VIR}\omega_0^{VIR}s + \omega_0^{VIR^2}} \\
H^{EXP}(s) &= \frac{Y^{EXP}}{\Lambda^{EXP}} = \frac{1}{m^{EXP}s^2 + d^{EXP}s + k^{EXP}} \\
&= \frac{1}{m^{EXP}} \cdot \frac{1}{s^2 + 2\zeta^{EXP}\omega_0^{EXP}s + \omega_0^{EXP^2}} \\
H^{REF}(s) &= \frac{Y^{VIR}}{F_{ext}} = \frac{Y^{EXP}}{F_{ext}} = \frac{1}{m^{VIR}s^2 + d^{VIR}s + k^{VIR} + k^{EXP}}
\end{aligned}
\tag{C.1}
$$

$\zeta^{VIR} = {}^{d^{VIR}}/_{2\sqrt{m^{VIR}k^{VIR}}}$, $\zeta^{EXP} = {}^{d^{EXP}}/_{2\sqrt{m^{EXP}k^{EXP}}}$ and $\zeta^{ACT} = {}^{d^{ACT}}/_{2\sqrt{m^{ACT}k^{ACT}}}$ are the damping ratios of the virtual component, the experimental component and the actuator. $\omega_0^{VIR} = \sqrt{{}^{k^{VIR}}/_{m^{VIR}}}$, $\omega_0^{EXP} = \sqrt{{}^{k^{EXP}}/_{m^{EXP}}}$, $\omega_0^{ACT} = \sqrt{{}^{k^{ACT}}/_{m^{ACT}}}$ and $\omega_0^{TR} = \sqrt{{}^{(k^{EXP}+k^{ACT})}/_{m^{ACT}}}$ eigenfrequencies of the virtual component and the actuation system with the experimental component. The open-loop transfer function reads

$$
\begin{aligned}
H_{OL}(s) =& C \cdot H^{TR}(s) \cdot \left(H^{VIR}(s) + H^{EXP}(s)\right) \\
&= \left(\frac{m^{EXP}}{m^{VIR}} \cdot \frac{s^2 + 2\zeta^{EXP}\omega_0^{EXP}s + \omega_0^{EXP^2}}{s^2 + 2\zeta^{VIR}\omega_0^{VIR}s + \omega_0^{VIR^2}} + 1\right) \\
&\cdot \frac{k^P}{m^{ACT}+m^{EXP}} \cdot \frac{1}{s^2 + \frac{2\zeta^{ACT}\sqrt{k^{ACT}m^{ACT}} + 2\zeta^{EXP}\sqrt{k^{EXP}m^{EXP}}}{m^{ACT}+m^{EXP}}s + \omega_0^{TR^2}}.
\end{aligned}
\tag{C.2}
$$

The system properties can be described with the mass ratio ϕ^{ACT}, the stiffness ratios κ^{P}, κ^{EXP} and κ^{ACT}, and the damping ratios ζ^{VIR} and ζ^{ACT}.

$$\begin{aligned}
&\kappa^{EXP} = \frac{k^{EXP}}{k^{VIR}}, \qquad \kappa^{ACT} = \frac{k^{ACT}}{k^{VIR}}, \qquad \kappa^{P} = \frac{k^{P}}{k^{VIR}}\\
&\phi^{EXP} = \frac{m^{EXP}}{m^{VIR}} \qquad \phi^{ACT} = \frac{m^{ACT}}{m^{VIR}}
\end{aligned} \tag{C.3}$$

In order to make $H_{OL}(s)$ non-dimensional, the following terms are substituted by expressions which depend only on non-dimensional parameters and the eigenfrequency of the coupled system ω_0:

$$\begin{aligned}
\omega_0^{VIR^2} &= \frac{\phi^{EXP}+1}{\kappa^{EXP}+1}\cdot\omega_0^2 \qquad \text{with} \qquad \omega_0 = \sqrt{\frac{k^{EXP}+k^{VIR}}{m^{EXP}+m^{VIR}}}\\
\omega_0^{TR^2} &= \frac{\kappa^{ACT}+\kappa^{EXP}}{\phi^{ACT}+\phi^{EXP}}\cdot\omega_0^{VIR^2} = \frac{\kappa^{ACT}+\kappa^{EXP}}{\phi^{ACT}+\phi^{EXP}}\cdot\frac{\phi^{EXP}+1}{\kappa^{EXP}+1}\cdot\omega_0^2\\
\omega_0^{ACT^2} &= \frac{\kappa^{ACT}}{\phi^{ACT}}\cdot\omega_0^{VIR^2} = \frac{\kappa^{ACT}}{\phi^{ACT}}\cdot\frac{\phi^{EXP}+1}{\kappa^{EXP}+1}\cdot\omega_0^2\\
\omega_0^{EXP^2} &= \frac{\kappa^{EXP}}{\phi^{EXP}}\cdot\omega_0^{VIR^2} = \frac{\kappa^{EXP}}{\phi^{EXP}}\cdot\frac{\phi^{EXP}+1}{\kappa^{EXP}+1}\cdot\omega_0^2\\
\frac{k^{P}}{m^{ACT}+m^{EXP}} &= \frac{\kappa^{P}}{\phi^{ACT}+\phi^{EXP}}\cdot\omega_0^{VIR^2} = \frac{\kappa^{P}}{\phi^{ACT}+\phi^{EXP}}\cdot\frac{\phi^{EXP}+1}{\kappa^{EXP}+1}\cdot\omega_0^2\\
\frac{2\zeta^{EXP}\sqrt{k^{EXP}m^{EXP}}}{m^{ACT}+m^{EXP}} &= \frac{2\zeta^{EXP}\sqrt{\kappa^{EXP}\phi^{EXP}}}{\phi^{ACT}+\phi^{EXP}}\omega_0^{VIR} = \frac{2\zeta^{EXP}\sqrt{\kappa^{EXP}\phi^{EXP}}}{\phi^{ACT}+\phi^{EXP}}\cdot\sqrt{\frac{\phi^{EXP}+1}{\kappa^{EXP}+1}}\cdot\omega_0\\
\frac{2\zeta^{ACT}\sqrt{k^{ACT}m^{ACT}}}{m^{ACT}+m^{EXP}} &= \frac{2\zeta^{ACT}\sqrt{\kappa^{ACT}\phi^{ACT}}}{\phi^{ACT}+\phi^{EXP}}\omega_0^{VIR} = \frac{2\zeta^{ACT}\sqrt{\kappa^{ACT}\phi^{ACT}}}{\phi^{ACT}+\phi^{EXP}}\cdot\sqrt{\frac{\phi^{EXP}+1}{\kappa^{EXP}+1}}\cdot\omega_0
\end{aligned} \tag{C.4}$$

The open-loop transfer function then reads

$$\begin{aligned}
H_{OL}(s) = &\left(\phi^{EXP}\cdot\frac{s^2+2\zeta^{EXP}\sqrt{\frac{\kappa^{EXP}}{\phi^{EXP}}\cdot\frac{\phi^{EXP}+1}{\kappa^{EXP}+1}}\cdot\omega_0\cdot s+\frac{\kappa^{EXP}}{\phi^{EXP}}\cdot\frac{\phi^{EXP}+1}{\kappa^{EXP}+1}\cdot\omega_0^2}{s^2+2\zeta^{VIR}\sqrt{\frac{\phi^{EXP}+1}{\kappa^{EXP}+1}}\cdot\omega_0\cdot s+\frac{\phi^{EXP}+1}{\kappa^{EXP}+1}\cdot\omega_0^2}+1\right)\cdot\\
&\cdot\frac{\frac{\kappa^{P}}{\phi^{ACT}+\phi^{EXP}}\cdot\frac{\phi^{EXP}+1}{\kappa^{EXP}+1}\cdot\omega_0^2}{s^2+2\left(\frac{\zeta^{ACT}\sqrt{\kappa^{ACT}\phi^{ACT}}}{\phi^{ACT}+\phi^{EXP}}+\frac{\zeta^{EXP}\sqrt{\kappa^{EXP}\phi^{EXP}}}{\phi^{ACT}+\phi^{EXP}}\right)\cdot\sqrt{\frac{\phi^{EXP}+1}{\kappa^{EXP}+1}}\cdot\omega_0\cdot s+\frac{\kappa^{ACT}+\kappa^{EXP}}{\phi^{ACT}+\phi^{EXP}}\cdot\frac{\phi^{EXP}+1}{\kappa^{EXP}+1}\cdot\omega_0^2}.
\end{aligned} \tag{C.5}$$

In the next step, the non-dimensional complex frequency $\hat{s} = \frac{s}{\omega_0}$ is used. The resulting transfer function $H_{OL}(\hat{s})$ is non-dimensional:

$$\begin{aligned}
H_{OL}(\hat{s}) = &\left(\phi^{EXP}\cdot\frac{\hat{s}^2+2\zeta^{EXP}\sqrt{\frac{\kappa^{EXP}}{\phi^{EXP}}\cdot\frac{\phi^{EXP}+1}{\kappa^{EXP}+1}}\cdot\hat{s}+\frac{\kappa^{EXP}}{\phi^{EXP}}\cdot\frac{\phi^{EXP}+1}{\kappa^{EXP}+1}}{\hat{s}^2+2\zeta^{VIR}\sqrt{\frac{\phi^{EXP}+1}{\kappa^{EXP}+1}}\hat{s}+\frac{\phi^{EXP}+1}{\kappa^{EXP}+1}}+1\right)\cdot\\
&\cdot\frac{\frac{\kappa^{P}}{\phi^{ACT}+\phi^{EXP}}\cdot\frac{\phi^{EXP}+1}{\kappa^{EXP}+1}}{\hat{s}^2+2\left(\frac{\zeta^{ACT}\sqrt{\kappa^{ACT}\phi^{ACT}}}{\phi^{ACT}+\phi^{EXP}}+\frac{\zeta^{EXP}\sqrt{\kappa^{EXP}\phi^{EXP}}}{\phi^{ACT}+\phi^{EXP}}\right)\cdot\sqrt{\frac{\phi^{EXP}+1}{\kappa^{EXP}+1}}\cdot\hat{s}+\frac{\kappa^{ACT}+\kappa^{EXP}}{\phi^{ACT}+\phi^{EXP}}\cdot\frac{\phi^{EXP}+1}{\kappa^{EXP}+1}}
\end{aligned} \tag{C.6}$$

Two special cases are given in Eq. (C.7) and Eq. (C.8) where the mass of the experimental component and the stiffness of the experimental component, respectively, disappear.

$$\begin{aligned}\phi^{EXP}=0: \qquad H_{OL}(\hat{s}) &= \left(\frac{\frac{\kappa^{EXP}}{\kappa^{EXP}+1}}{\hat{s}^2+2\zeta^{VIR}\sqrt{\frac{1}{\kappa^{EXP}+1}}\cdot\omega_0\cdot\hat{s}+\frac{1}{\kappa^{EXP}+1}}+1\right)\cdot\\ &\cdot\frac{\frac{\kappa^P}{\phi^{ACT}}\cdot\frac{1}{\kappa^{EXP}+1}}{\hat{s}^2+2\cdot\zeta^{ACT}\sqrt{\frac{\kappa^{ACT}}{\phi^{ACT}}\cdot\frac{1}{\kappa^{EXP}+1}}\cdot\hat{s}+\frac{\kappa^{ACT}+\kappa^{EXP}}{\phi^{ACT}}\cdot\frac{1}{\kappa^{EXP}+1}}\end{aligned} \tag{C.7}$$

$$\begin{aligned}\kappa^{EXP}=0: \qquad H_{OL}(\hat{s}) &= \left(\phi^{EXP}\cdot\frac{\hat{s}^2+2\zeta^{EXP}\sqrt{\frac{\phi^{EXP}+1}{\phi^{EXP}}}\cdot\hat{s}+\frac{\phi^{EXP}+1}{\phi^{EXP}}}{\hat{s}^2+2\zeta^{VIR}\sqrt{\phi^{EXP}+1}\hat{s}+\phi^{EXP}+1}+1\right)\cdot\\ &\cdot\frac{\frac{\kappa^P\cdot\left(\phi^{EXP}+1\right)}{\phi^{ACT}+\phi^{EXP}}}{\hat{s}^2+2\frac{\zeta^{ACT}\sqrt{\kappa^{ACT}\phi^{ACT}}\cdot\sqrt{\phi^{EXP}+1}}{\phi^{ACT}+\phi^{EXP}}\cdot\hat{s}+\frac{\kappa^{ACT}\cdot(\phi^{EXP}+1)}{\phi^{ACT}+\phi^{EXP}}}\end{aligned} \tag{C.8}$$

C.2 Closed-loop Transfer Function with Actuator Dynamics

The following section refers to the derivation of the closed-loop transfer function in Sec. 4.2.3. The Laplace domain interface displacements are

$$\begin{aligned} Y^{EXP} &= -CH^{TR}H^{EXP}(-Y^{VIR}+Y^{EXP})\\ \Rightarrow \qquad Y^{EXP}(CH^{TR}H^{EXP}+1) &= CH^{TR}H^{EXP}Y^{VIR}\end{aligned} \tag{C.9}$$

$$\text{and} \qquad Y^{VIR}=H^{VIR}F_{ext}+CH^{TR}H^{VIR}\left(-Y^{VIR}+Y^{EXP}\right) \tag{C.10}$$

Substituting (C.9) in (C.10) yields

$$\begin{aligned} Y^{VIR} &= H^{VIR}F_{ext}+CH^{TR}H^{VIR}\left(-Y^{VIR}+\frac{CH^{TR}H^{EXP}}{CH^{TR}H^{EXP}+1}Y^{VIR}\right)\\ Y^{VIR}&\left(1-CH^{TR}H^{VIR}\left(-1+\frac{CH^{TR}H^{EXP}}{CH^{TR}H^{EXP}+1}\right)\right)=H^{VIR}F_{ext}.\end{aligned} \tag{C.11}$$

Division by F_{ext} yields the closed-loop transfer function

$$\begin{aligned} H_{CL}=\frac{Y^{VIR}}{F_{ext}} &= \frac{H^{VIR}}{1+CH^{TR}H^{VIR}\left(1-\frac{CH^{TR}H^{EXP}}{CH^{TR}H^{EXP}+1}\right)}\\ &= \frac{(CH^{TR}H^{EXP}+1)H^{VIR}}{CH^{TR}H^{EXP}+1+CH^{TR}H^{VIR}(CH^{TR}H^{EXP}+1-CH^{TR}H^{EXP})}\\ &= \frac{CH^{TR}H^{EXP}H^{VIR}+H^{VIR}}{CH^{TR}(H^{EXP}+H^{VIR})+1}.\end{aligned} \tag{C.12}$$

Appendix D

Derivation of Recursive Least Squares Adaptation Laws

D.1 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. 5.2.2 and Sec. 5.4.2.

$$\begin{aligned}
\boldsymbol{P}_{g,u}\boldsymbol{\theta} &= \begin{bmatrix} \mathrm{Re}\left(H_{g,u}(\Omega_1)\right) & -\mathrm{Im}\left(H_{g,u}(\Omega_1)\right) & 0 & 0 \\ \mathrm{Im}\left(H_{g,u}(\Omega_1)\right) & \mathrm{Re}\left(H_{g,u}(\Omega_1)\right) & 0 & 0 \\ 0 & 0 & \mathrm{Re}\left(H_{g,u}(\Omega_2)\right) & -\mathrm{Im}\left(H_{g,u}(\Omega_2)\right) \\ 0 & 0 & \mathrm{Im}\left(H_{g,u}(\Omega_2)\right) & \mathrm{Re}\left(H_{g,u}(\Omega_2)\right) \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} \\
&= \begin{bmatrix} \mathrm{Re}\left(H_{g,u}(\Omega_1)\right)\theta_1 - \mathrm{Im}\left(H_{g,u}(\Omega_1)\right)\theta_2 \\ \mathrm{Im}\left(H_{g,u}(\Omega_1)\right)\theta_1 + \mathrm{Re}\left(H_{g,u}(\Omega_1)\right)\theta_2 \\ \mathrm{Re}\left(H_{g,u}(\Omega_2)\right)\theta_3 - \mathrm{Im}\left(H_{g,u}(\Omega_2)\right)\theta_4 \\ \mathrm{Im}\left(H_{g,u}(\Omega_2)\right)\theta_3 + \mathrm{Re}\left(H_{g,u}(\Omega_2)\right)\theta_4 \end{bmatrix} \\
&= \underbrace{\begin{bmatrix} \theta_1 & -\theta_2 & 0 & 0 \\ \theta_2 & \theta_1 & 0 & 0 \\ 0 & 0 & \theta_3 & -\theta_4 \\ 0 & 0 & \theta_4 & \theta_3 \end{bmatrix}}_{\tilde{\boldsymbol{\theta}}} \underbrace{\begin{bmatrix} \mathrm{Re}\left(H_{g,u}(\Omega_1)\right) \\ \mathrm{Im}\left(H_{g,u}(\Omega_1)\right) \\ \mathrm{Re}\left(H_{g,u}(\Omega_2)\right) \\ \mathrm{Im}\left(H_{g,u}(\Omega_2)\right) \end{bmatrix}}_{\tilde{\boldsymbol{P}}_{g,u}}
\end{aligned}$$

D.2 Recursive Least Squares based on QR Decomposition

The following derivation refers to the RLS algorithm, which is used in Sec. 5.4.1. In Eq. 5.33, the matrix $\boldsymbol{\Phi}$ is only changed by the rank-n_{int} matrix $\boldsymbol{P}^T_{g,u}\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{P}_{g,u}$ 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. (5.32). $\boldsymbol{\Phi}$ 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. (D.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.

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

Using the definition of the square-root factors in Eq. (D.1), the update Eqs. (5.33) can be split into factors according to Eq. (D.2). One can verify that Eq. (D.2) is equivalent to Eqs. (5.33).

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

If an orthogonal Transformation with the matrix $\mathcal{C}$ ($\mathcal{C}^T\mathcal{C} = \boldsymbol{I}$) according to Eq. (D.3) exists, Eq. (D.2)—which corresponds to the update equations—holds (proof in [65]).

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

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. (5.33). 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. (D.4) for $\boldsymbol{\theta}[k+1]$.

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