

**A subspace based approach to the design,
implementation and validation of algorithms
for active vibration isolation control**

**A subspace based approach to the design,
implementation and validation of algorithms
for active vibration isolation control**

Gerard Nijse

Ph.D. Thesis • University of Twente • The Netherlands

Thesis committee:

Prof. dr. ir. H.J. Grootenboer (chairman)	University of Twente
Prof. dr. ir. J.B. Jonker (supervisor)	University of Twente
Dr. ir. J. van Dijk (supervisor)	University of Twente
Dr. ir. A.P. Berkhoff	TNO Science and Industry, Delft
Prof. dr. ir. A. de Boer	University of Twente
Prof. dr. A.J. van der Schaft	University of Groningen, Groningen
Prof. dr. ir. M. Steinbuch	Eindhoven University of Technology, Eindhoven

© Copyright, G. Nijse, Enschede, 2006

Cover design by W. Hagedoorn, Almelo and E.G.F. van den Berg, Amsterdam
English text revisions by kOog voor Tekst, Koog aan de Zaan
Printed by PrintPartners Ipskamp, Enschede

ISBN 90-365-2319-2

A SUBSPACE BASED APPROACH TO THE DESIGN,
IMPLEMENTATION AND VALIDATION OF ALGORITHMS
FOR ACTIVE VIBRATION ISOLATION CONTROL

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof. dr. W.H.M. Zijm,
volgens besluit van het College voor Promoties
in het openbaar te verdedigen
op donderdag 9 februari 2006 om 16.45 uur

door

Gerard Nijse
geboren op 19 augustus 1974
te Ede

Dit proefschrift is
goedgekeurd door de promotor
prof. dr. ir. J.B. Jonker.

The fear of the Lord is the beginning of knowledge.

– *Holy Bible, Proverbs 1:7*

Contents

1	Introduction	1
1.1	Vibration isolation	1
1.2	Magnetic resonance imaging scanner	3
1.3	Problem statement and assumptions	6
1.3.1	Problem statement	6
1.3.2	Assumptions	7
1.4	Experimental vibration isolation setups	7
1.4.1	The SR-SISO vibration isolation setup	7
1.4.2	The SR-MIMO vibration isolation setup	8
1.5	Contributions and outline	9
1.5.1	Contributions	9
1.5.2	Outline	11
2	Background on control and identification	13
2.1	Introduction	13
2.2	Feedforward and feedback control	14
2.2.1	The one-degree-of-freedom mass/damper/spring system and the control objective	14
2.2.2	Feedforward control	15
2.2.3	Feedback control	19
2.3	Fixed gain control versus adaptive control	22
2.4	Multiple input multiple output active vibration isolation control systems	24
2.4.1	Block diagrams of feedforward and feedback active vibration isolation control systems	24
2.4.2	The internal model control principle	27
2.5	Black-box identification of the plants	28
2.5.1	Subspace model identification	29
2.5.2	Determining the model quality of an identified model	30
2.5.3	Guidelines to obtain a proper data set for identification	31
3	Design of fixed gain and adaptive controllers for narrowband and broadband disturbance rejection	33
3.1	Introduction	33
3.2	Wiener controller	34
3.2.1	Control objective	34

3.2.2	Filter structure	35
3.2.3	Feedforward control scenario	36
3.2.4	Improving the robustness of the controller with regularization	40
3.2.5	Feedback control scenario	41
3.3	Filtered-x least mean square algorithm	45
3.3.1	Derivation of the algorithm	46
3.3.2	Convergence analysis of the adaptive controller coefficients	49
3.3.3	Incorporation of state space models	51
3.3.4	Improving the robustness of the adaptive controller with regularization	53
3.4	Post-conditioning the adaptive controller for faster convergence	53
3.4.1	Post-conditioning in the filtered-x least mean square algorithm	54
3.4.2	Application to the filtered-error least mean square algorithm	58
3.5	Principle component least mean square algorithm	62
4	Subspace model identification of the transfer paths of the experimental vibration isolation setups	65
4.1	Introduction	65
4.2	Identification of the transfer paths of the SR-SISO vibration isolation setup	65
4.2.1	Acquisition of a proper data set for identification	66
4.2.2	Identification of the primary path and the secondary path	66
4.3	Identification of the transfer paths of the SR-MIMO vibration isolation setup	70
4.3.1	Acquisition of proper data sets for identification	70
4.3.2	Identification of the primary path	70
4.3.3	Identification of the secondary path	74
5	Control of narrowband disturbances on the SR-MIMO vibration isolation set-up	79
5.1	Introduction	79
5.2	Eliminating a single frequency disturbance output	80
5.2.1	Preliminaries concerning the experiments	80
5.2.2	Eliminating a disturbance output containing 528 Hz	81
5.2.3	Eliminating a disturbance output containing 565 Hz	84
5.3	Eliminating a disturbance output containing two frequencies	86
5.3.1	Preliminaries concerning the experiments	86
5.3.2	Performance of the algorithms	86
6	Control of broadband disturbances on the vibration isolation setups	89
6.1	Introduction	89
6.2	SR-SISO vibration isolation setup	89
6.2.1	Preliminaries concerning the experiments	90
6.2.2	The inverse outer factor and the adjoint of the inner factor	91
6.2.3	Fixed gain and adaptive feedback control	93
6.3	SR-MIMO vibration isolation setup	97
6.3.1	Fixed gain feedforward control	97
6.3.2	Adaptive feedforward control	102
6.3.3	Fixed gain feedback control	106
6.3.4	Adaptive feedback control	109

7	Conclusions and recommendations	113
7.1	Conclusions	113
7.2	Recommendations	115
A	Filter structures	117
A.1	Introduction	117
A.2	The z -transform variable and the unit delay operator q^{-1}	117
A.3	Finite impulse response filter structure	118
A.4	Infinite impulse response filter structure	122
A.5	State space filter structures	125
B	The feedforward model based Wiener controller	129
C	Simulation example: subspace model identification of the primary path of the one-degree-of-freedom mass/damper/spring system	131
C.1	Introduction	131
C.2	Without process and measurement noise.	132
C.3	With 50 dB process noise and 50 dB measurement noise.	135
D	Extracting a finite impulse response model for purposes of tonal control	139
E	Hardware used for the identification and control tasks	141
E.1	The components used for data acquisition	141
E.2	dSPACE systems	142
E.2.1	The dSPACE DS1103 controller board	142
E.2.2	The modular dSPACE system with DS1005 controller board	143
F	The two-step regularization approach	145
	Summary	147
	Samenvatting	149
	Glossary	151
	Patent and publications	155
	References	158
	Acknowledgements	165
	Curriculum Vitae	167

Chapter 1

Introduction

1.1 Vibration isolation

Isolating vibrations is an important issue in a wide range of engineering applications where mechanical parts need to be connected and, at the same time, the transfer of vibration energy between the parts needs to be minimal. Vibrations can make machines less accurate, result in unwanted noise, and cause fatigue in parts of the structure or even cause damage. Examples can be found in: precision technology, transport vehicles, space and aerospace applications, biomedical applications, oil and gas platforms and buildings and constructions in a zone of seismic activity.

Vibration isolation is aimed at reducing the transmission of vibration energy from one body or structure to another. The body where the vibration originates from and the structure on which this body is mounted are called, in short, the ‘source’ and the ‘receiver’, respectively. For an illustration, see figure 1.1. The source - which is assumed to be rigid - needs to be appropriately connected to a receiver in order to constrain the motions of the source relative to the receiver. In this thesis, the receiver is considered to be flexible. A well-known technique to reduce the transfer of vibration energy is passive vibration isolation (PVI) by using springs and dampers (Hansen and Snyder, 1997; Mead, 1998; Preumont, 2002). See figure 1.2 for a depiction of this. The stiffness of the mount determines the fundamental resonance frequency of the mounted system and vibrations with a frequency higher than the fundamental resonance frequency are attenuated. Unfortunately, however, other design requirements (like static stability) often impose a minimum allowable stiffness, thus limiting the achievable vibration isolation by passive means. Vibrations with lower frequencies especially are difficult to isolate. Passively isolating these kinds of vibrations would result in relatively large displacements of the source. This is not desirable, as in many applications the requirements concerning the support functionality are rather strict and cannot be relaxed (much).

A more promising approach to vibration isolation is hybrid vibration isolation, which combines PVI and active vibration isolation control (AVIC). However, in this thesis it is proposed to use a special form of hybrid vibration isolation using statically determinate stiff supports (Super, 2006; Nijssen et al., 2005). The approach is addressed later in this chapter when the setups are described. This approach will be denoted as statically determinate hybrid vibration isolation control (SD-AVIC) to differentiate it from the general AVIC approach as described in e.g. (Hansen and Snyder, 1997; Fuller et al., 1996). The aim of the control system

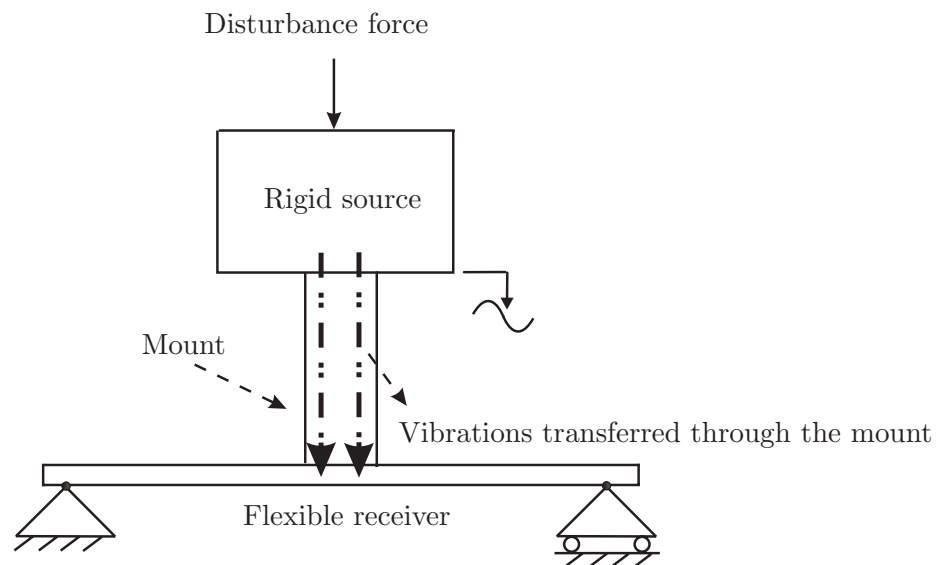


Figure 1.1: The vibrations produced by the vibrating source are transferred through the mount to the receiver. As a result, the receiver also vibrates.

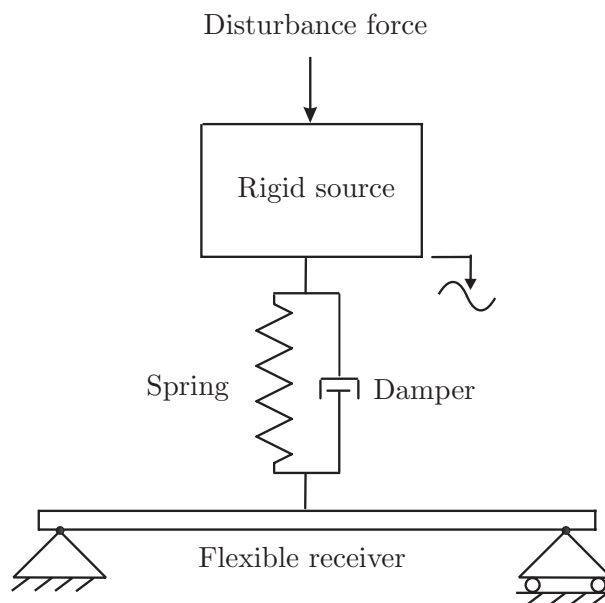


Figure 1.2: The vibrating source is isolated from the receiver by a damper and a spring. By adjusting the spring constant and the damper constant, a certain level of vibration isolation can be established.

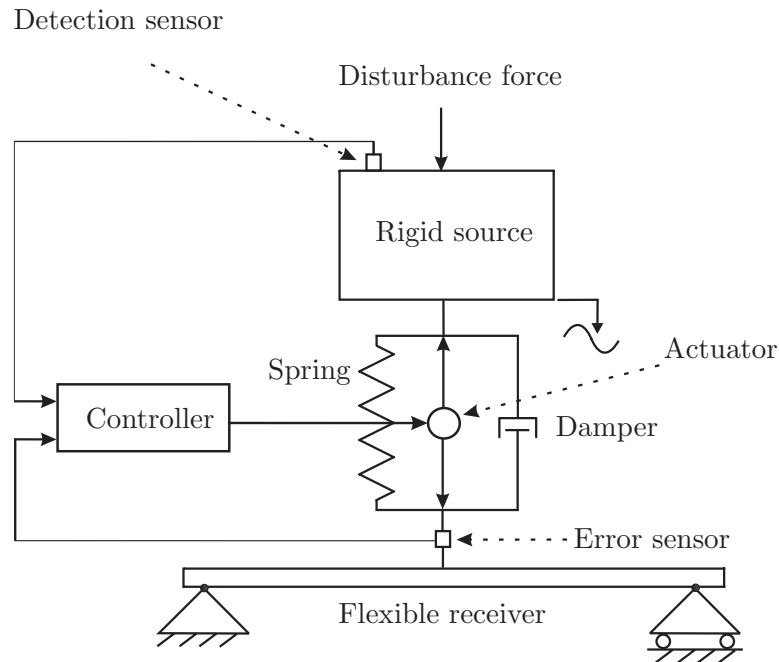


Figure 1.3: *The vibrating source is isolated from the receiver by a spring and a damper. Furthermore, an SD-AVIC system is used to produce an anti-disturbance force.*

in SD-AVIC is to produce anti-disturbance forces that counteract with the disturbance forces stemming from the source. Using this approach, the vibration energy transfer from the source to the receiver is blocked in the mount due to the anti-forces. For this, along with the springs and dampers, an electrical circuit is needed that consists of sensors, actuators and a real-time control system. This is depicted in figure 1.3 for a one-degree-of-freedom (1DOF) AVIC configuration. The detection sensor provides information about the disturbance. The error sensor provides information about the force that needs to be attenuated. The information gathered from both the detection sensor and the error sensor is used by the controller to steer the actuator and to generate an anti-disturbance force.

The main contribution of the research described in this thesis is the design of controllers generating these anti-forces using techniques common in the field of signal processing.

Disturbances on the source (or produced by it) can be narrowband or broadband. Broadband disturbances are random by nature and distribute their energy over a broad frequency band. Narrowband disturbances consist of a sine or a multi-sine. Examples of narrowband signals include signals produced by reciprocal machines, such as internal combustion engines, compressors, refrigerators and vacuum pumps. Examples of broadband signals include signals produced by jet engines and magnetic resonance imaging (MRI) scanners (see figure 1.4).

1.2 Magnetic resonance imaging scanner

A typical application in which vibration isolation is necessary to prevent undesired sound radiation is the MRI scanner. The MRI scanner is widely used in hospitals for both medical



Figure 1.4: A photograph of an MRI scanner. The patient lies on the small table that slides into the scanner.

diagnosis and clinical research into the human body. For examples of this, see e.g. (Roozen et al., 2005; Woodward, 2001; Westbrook, 2001). In (Price et al., 2001) it is reported that some scanners can produce acoustic noise levels up to 130 dB(A). This is equivalent to the noise experienced at approximately 15 meters distance of a military jet that takes off with an afterburner. Price et al. (2001) points out that medical personnel are likely to suffer hearing damage before patients due to the fact that they are exposed to the high noise levels for longer time periods.

The noise production can be explained as follows. Two main components of the MRI scanner are the large and heavy main magnet and the smaller and lighter gradient coil that is placed inside the main magnet (see figure 1.5). The main magnet provides a strong magnetic field (up to several Tesla), while the gradient coil produces a weak magnetic field. The gradient coil is turned on and off very rapidly in a specific manner to alter the main magnetic field on a very local level. Due to the strong main magnetic field, excessive Lorentz forces are produced that act on the gradient coil. As a result, the gradient coil vibrates. This produces undesired sound radiation. Indirectly, the forces also cause vibrations in the main magnet housing (receiver) to which the gradient coil (source) is connected. As a consequence, the housing will also emit sound. In (Roozen et al., 2005) experiments are described with a gradient coil that is attached by means of rigid mounts and a gradient coil that is attached by means of a mount that consists of an air spring for PVI and a piezo-electric actuator for control. For an illustration, see figure 1.6. The control system is comprised of single input single output (SISO) feedback controllers, with each SISO controller steering an actuator. The sound pressure level at the operator position that is at some distance from the scanner drops by about 8 dB when using the mount with the air spring instead of the rigid mounts. An additional reduction up to 10 dB in particular frequency regions between 1 and 2 kHz is obtained by switching the control system on. This shows the potential of applying control techniques to reduce the transfer of vibration energy.

In order to attempt to increase vibration isolation performance, in this thesis the use of SD-AVIC (as mentioned earlier) is suggested, along with multivariable control approaches.

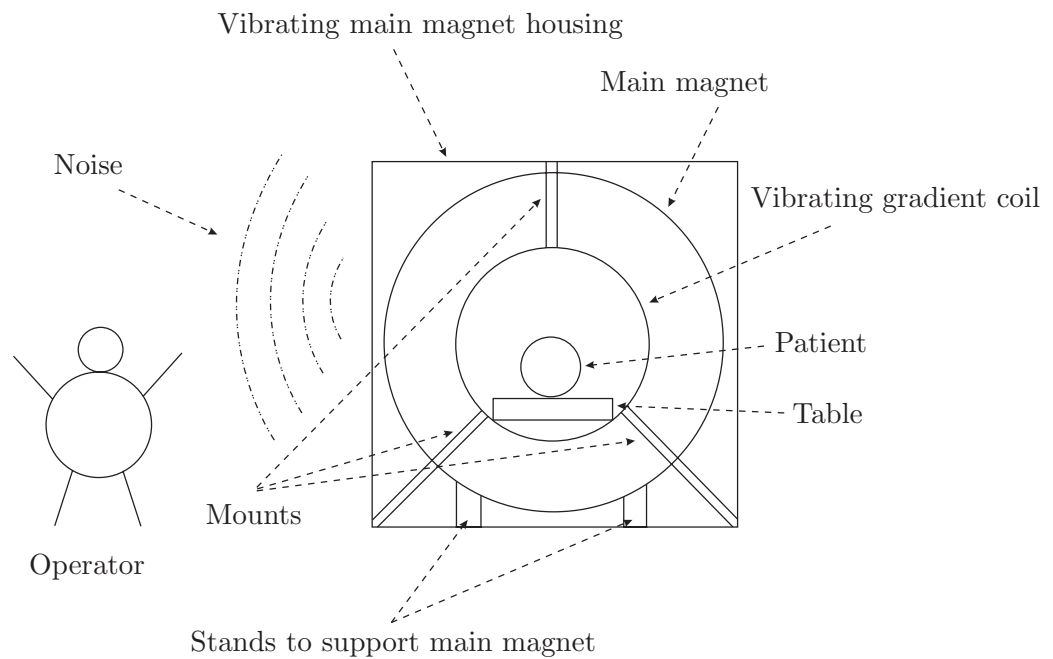


Figure 1.5: A schematic picture of an MRI scanner (front view) with the patient lying inside and the operator standing next to the scanner.

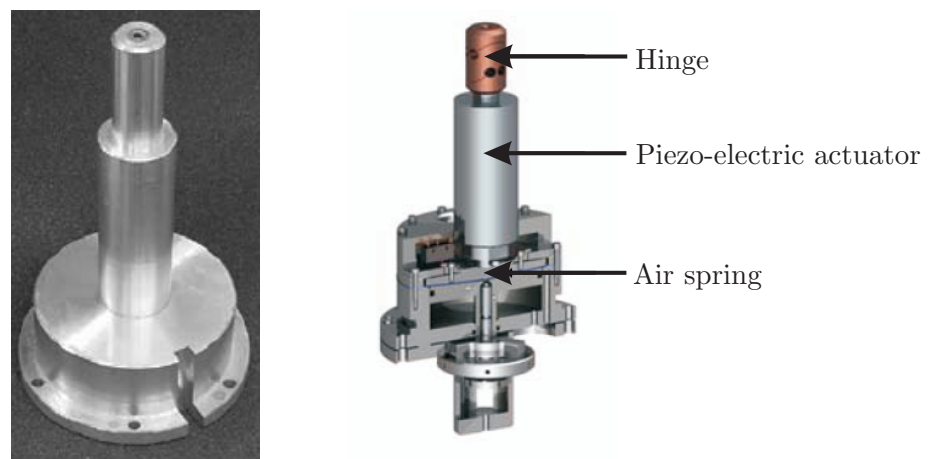


Figure 1.6: Rigid mount (left) and hybrid mount (right).

However, in order to be able to design, implement and validate multivariable control approaches for SD-AVIC, a number of problems has to be solved.

1.3 Problem statement and assumptions

1.3.1 Problem statement

In this thesis model-based control approaches are investigated. These approaches are both adaptive and fixed gain, as well as feedforward and feedback oriented. The algorithms, that were initially designed for application in the field of signal processing (Elliott, 2000*b*), are modified and applied to SD-AVIC in mechanical structures. However, the following problems need to be solved (solutions to these problems are described in the contributions):

1. **Designing controllers.** Three fundamental decisions have to be made in designing a controller that should be robust, computationally feasible and numerically stable. First, it has to be decided if the controller is feedforward or feedback. Second, it has to be decided if the controller is adaptive or fixed gain. Third, the control algorithm has to be chosen, for example an H_2 or H_∞ control algorithm.
2. **Computational complexity.** SD-AVIC systems are time critical. Assuming that the controller is updated on a sample-by-sample basis (which often is the case), at each sample instant both a new controller and the actuator steering signals must be computed. Although integrated microprocessors become faster every day, controlling large scale applications can still pose computational problems. These are especially pronounced when broadband disturbance signals need to be cancelled in a multiple actuator and multiple error sensor scenario.
3. **Robustness under varying conditions.** SD-AVIC systems should be able to operate a number of days in a row in a possibly changing environment. This means that there should be mechanisms in the SD-AVIC system that deal with those changes. Another requirement for the controller is that it has to be numerically robust. Small perturbations of the parameters of the controller (or the model) must not lead to an unstable control process.
4. **Modelling the plant under control.** The control approaches investigated are done using an internal model of the plant under control. Therefore, the performance of the controller depends to a large extent on the quality of the models. In standard AVIC, the models are often constructed from basic physical principles or from frequency response data. However, for complex real-life multiple input multiple output (MIMO) SD-AVIC systems it is difficult to determine models that are sufficiently accurate.
5. **Fast adaptation.** Due to resonances in mechanical structures, adaptive SD-AVIC systems are often hampered by slow convergence of the controller coefficients. In general, however, it is desirable that the SD-AVIC system yields fast optimum performance after it is switched on. Therefore, the SD-AVIC system has to be designed in such a way that fast convergence of the controller parameters is enabled.
6. **Validation on experimental setups.** In order to demonstrate the potential of the proposed control approaches, they are validated experimentally on two setups. These setups are described in section 1.4.

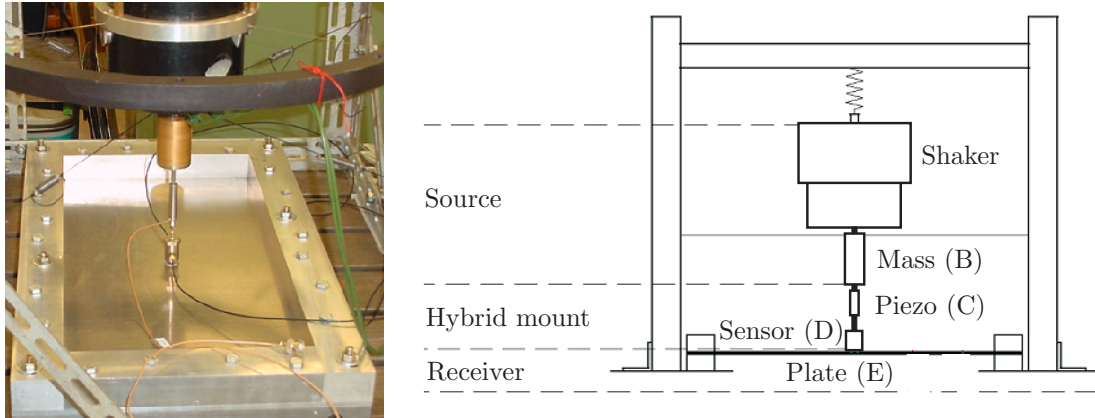


Figure 1.7: *The SR-SISO vibration isolation setup (left) and a schematic representation (right).*

1.3.2 Assumptions

Three basic assumptions are made:

- **Narrowband and broadband disturbances.** The disturbances under consideration are both narrowband and broadband. In this thesis, 'broad' means that the frequency of the disturbance varies from 0-1 kHz.
- **Stationary disturbance signal.** Only stationary and semi-stationary disturbances are considered. Semi-stationary means the disturbance can be considered stationary during a period ranging from a few seconds to a few minutes.
- **Linear time invariant.** It is assumed that the system is linear time invariant (LTI). Looking ahead, it can be said that in the research actuators are used that are made of piezo-electric material. The material exhibits a nonlinear hysteresis cycle. To satisfy the assumption of linearity, the amplitudes of the disturbances and the voltages applied to the piezo-electric actuators are limited in such a way that a small hysteresis cycle is obtained and linearity can be assumed.

1.4 Experimental vibration isolation setups

At the laboratory of Mechanical Automation two SD-AVIC systems are developed (Super, 2006): a single reference input single actuator input and single error sensor output (SR-SISO) vibration isolation setup and a single reference input multiple actuator input and multiple error sensor output (SR-MIMO) vibration isolation setup.

1.4.1 The SR-SISO vibration isolation setup

In figure 1.7 on the left, a photograph of the SR-SISO vibration isolation setup is shown. A schematic representation is depicted on the right. An electro-dynamic shaker (indicated by 'A' in the figure) excites a mass (B). The mass, which serves as a rigid vibration source, is

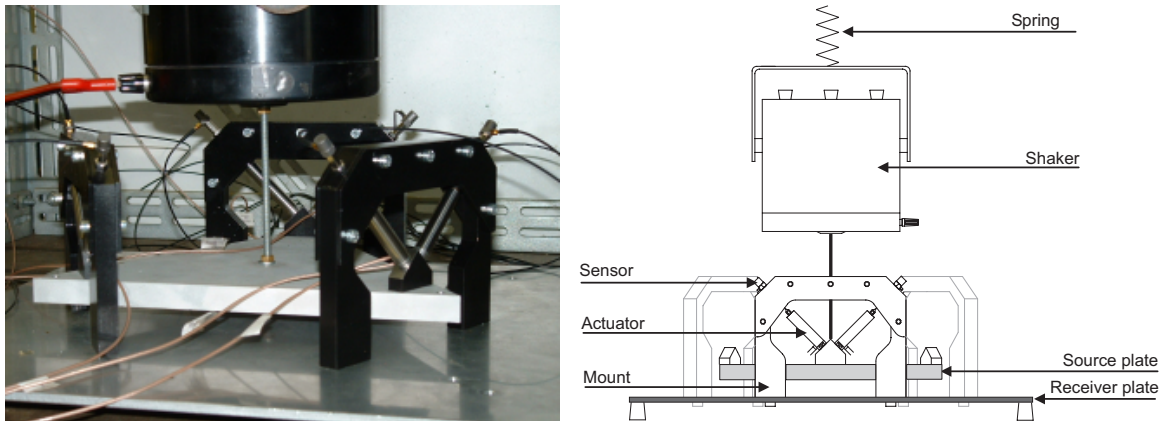


Figure 1.8: *The SR-MIMO vibration isolation setup (left) and a schematic representation (right).*

connected to a clamped metal plate (E) (i.e. the flexible receiver) by a piezo-electric actuator (C) that operates as a statically determinate hybrid vibration isolation mount. The objective of the setup is to determine if the receiver can be isolated from the source (B) by the vibration isolation mount, so as to reduce disturbances stemming from the source at the receiver. This is achieved by minimizing the signal from the error sensor output (D) and by steering the piezo-electric actuator input. In this thesis acceleration sensors are used; see also appendix E.

1.4.2 The SR-MIMO vibration isolation setup

The SR-SISO vibration isolation setup is limited by the fact that the source is isolated from the receiver in a single degree of freedom. In general, that is not a very realistic assumption. A rigid source that needs to be isolated from a flexible receiver commonly vibrates in multiple degrees of freedom. To be able to perform more realistic experiments, a SR-MIMO vibration isolation setup is developed.

In figure 1.8 on the left, a photograph of the SR-MIMO vibration isolation setup is shown. A schematic representation is depicted on the right. The triangular plate (i.e. the source) is excited by an electro-dynamic shaker. The shaker is attached to a frame by a spring and moves the source dominantly in the vertical direction. The source is connected to the receiver by three hybrid mounts that consist of two piezo-electric actuators each (a mount is shown in figure 1.9, on the left). These mounts provide high stiffness connections in the two actuator directions and low stiffness in the unactuated directions. Thus, SD-AVIC is established in a total of six degrees of freedom (three mounts \times two degrees of freedom) and PVI is established in the unactuated directions. Every mount has two error sensors that are aligned with the piezo-electric actuators. The objective of the SR-MIMO vibration isolation setup is to determine if the receiver can be isolated from the source by the vibration isolation mounts, so as to reduce disturbances stemming from the source at the receiver. SD-AVIC is established by minimizing the signals from the six error sensor outputs and by steering the six piezo-electric actuator inputs. A suspended construction is chosen instead of a standing construction such as a Stewart platform. See e.g. (Horodincea et al., 2002) for

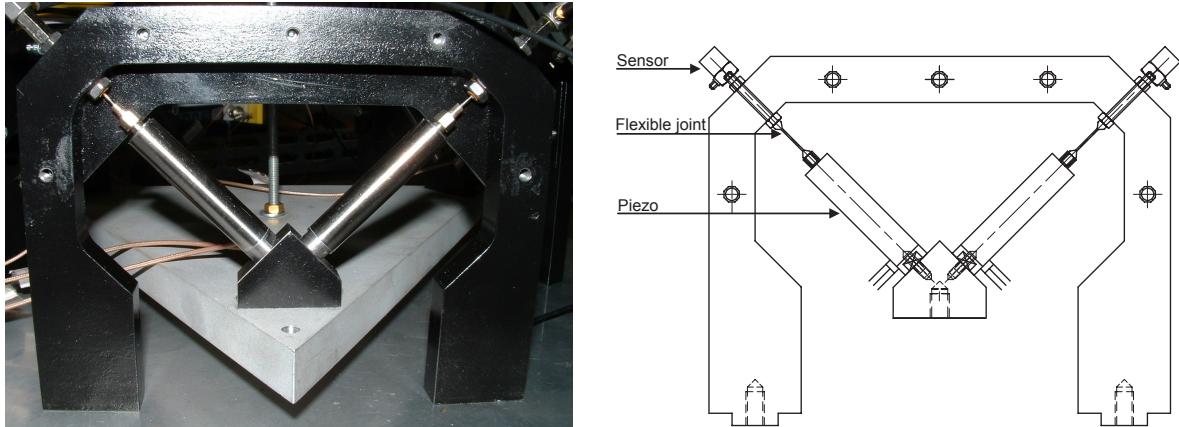


Figure 1.9: One of the three mounts that carry the source (left) and a schematic representation (right). The two piezo-electric actuators can be seen clearly.

an overview of Stewart platforms. The reason behind this choice is that in the case of a standing construction, the joints should be designed on buckling, which leads to thicker (and thus stiffer) joints. Stiffer joints provide less PVI in the unactuated directions. That would compromise the actual operation of the hybrid mount.

Both SD-AVIC setups are constructed so that they potentially enable control of disturbances from approximately 150 Hz upwards. The fact itself that frequencies below approximately 150 Hz are not considered is not an inherent limitation of the posed control methods. It does indicate, however, that constraints need to be incorporated in the design of the system to facilitate SD-AVIC in the lower frequency region.

With these two setups it is possible to investigate the problems indicated in section 1.3.

1.5 Contributions and outline

1.5.1 Contributions

The contributions of this thesis can be formulated as follows:

- To obtain accurate models suitable for SD-AVIC, in this thesis a black-box modelling strategy is used. The plants are identified using the black-box past output¹ multivariable output error subspace (PO-MOESP) subspace model identification (SMI) routine (Verhaegen, 1993). Identification is done using short batches of recorded (sampled) time-domain data. Furthermore, the benefit of SMI is fully utilized in the sense that all transfer paths are identified in one single (multivariable) identification experiment.
- A new modelling approach for adaptive SD-AVIC is adopted. It is proposed to model (some of) the plants for adaptive SD-AVIC by state space models (SSMs) that are obtained by SMI and to model the controller by a polynomial model, such as a finite impulse response model. This is in contrast to traditional approaches in which both

¹The name *Past-Output* refers to the construction of the instrumental variable that is used to exclude state and measurement noise from the data equation.

the plants and the adaptive controller are modelled by polynomial models. By using this approach, it is possible to increase the numerical robustness and to reduce the processor load resulting from updating the adaptive controller coefficients on the real-time arithmetic. Another benefit of this approach is that it paves the way for a broad spectrum of control engineering tools for designing (numerically) reliable feedforward and feedback controllers.

- A state space based inner/outer factorization is used in all broadband adaptive controllers in simulation and *real-time*, i.e. the inverse outer factor model is included in the control scheme to significantly speed up the convergence of the adaptive controller coefficients. This approach is applied to both feedforward and feedback control scenarios. It is demonstrated that for the setups under consideration, fast adaptation of the controller coefficients is possible.
- One of the main problems related to controller implementation for SD-AVIC is that significant actuator stroke may occur (especially in the low frequency area), which may lead to actuator saturation. Regularization can be applied to prevent saturation in certain frequency areas and to be able to successfully implement the controller. However, utilizing regularization in adaptive controllers that contain the inverse outer factor model may come with a significant overhead on the processor load, thereby hampering the performance of the controller. To solve that problem, in this thesis a two-step regularization approach is proposed. By setting two scalar parameters, the adaptive controller can be implemented and regularized without effecting the computational arithmetic in a negative way. This approach can be used for fixed gain controllers as well.
- Adaptive feedforward controllers are designed, implemented and validated on the SR-MIMO setup to cancel narrowband disturbances. For control, two least mean square based control algorithms are used: the traditional filtered-x least mean square algorithm and the principle component least mean square algorithm. The latter algorithm enables rapid adaptation of the controller coefficients due to its decoupling properties. Controllers are used to reject a disturbance signal that contains one frequency and one that contains two frequencies. To be able to achieve the latter, the controllers are implemented in a parallel way utilizing a twin reference input approach. The control approaches are validated in both simulation and real-time.
- A state space-based fixed gain single input six output regularized H_2 *feedforward* controller is designed, implemented and validated on the SR-MIMO setup, to cancel broadband disturbances in the frequency region of 0-1 kHz. To be able to implement the controller on the real-time arithmetic, the two-step regularization approach is utilized, which makes the controller more robust and prevents actuator saturation. Furthermore, the order of the controller is reduced in a way to allow the controller to be implemented. Both simulation and real-time results are presented.
- Also on the SR-MIMO setup a six input six output regularized H_2 *feedback* controller is designed, implemented and validated, to cancel broadband disturbances of 0-1 kHz. The controller is designed using the internal model principle, which allows the consideration of the feedback control problem as equivalent to the feedforward one. Due to the complexity of the controller, lower order counterparts of the plant models are estimated, so that the controller can be implemented. For a stable implementation of the controller

and to prevent actuator saturation, the two-step regularization approach is applied. Simulation and real-time results are presented.

- An adaptive broadband feedback controller is designed, implemented and validated on the SR-SISO vibration isolation setup in both simulation and real-time, to be able to cancel broadband disturbances of 0-2 kHz. To speed up the convergence of the adaptive controller, the state space based inverse outer factor model is included in the control scheme. In the internal model control based approach the subspace models are utilized to obtain a proper estimate of the disturbance signal. For control, an efficient adjoint least mean square based control algorithm is used, in which the error sensor output is filtered with a low order inner factor only. For a stable implementation of the controller and to prevent actuator saturation, regularization is applied.
- An adaptive broadband *feedforward* controller as well as an adaptive broadband *feedback* controller are designed, implemented and validated on the SR-MIMO vibration isolation setup in both simulation and real-time to be able to cancel broadband disturbances of 0-1 kHz. To speed up the convergence of the adaptive controller, the state space-based inverse outer factor model is included. The two-step regularization approach is used to make the controllers more robust, to prevent actuator saturation and to achieve a stable implementation of the (feedback) controller. For control, an efficient adjoint least mean square based algorithm is used in which the error sensor outputs are filtered with a compact low dimensional semi inner factor model only.
- Although the software package Matlab (and its toolboxes) delivers tools for the design of control systems, support for AVIC systems is limited. To speed up the development time, an AVIC toolbox is developed for Matlab. The toolbox is referred to as *Active Vibration Isolation Control Toolbox for Matlab* and enables the user to evaluate different control strategies presented in the thesis. Access to the toolbox is enabled by a graphical interface that allows the user to easily change settings and to investigate optimal parameters (Niewold, 2004).

1.5.2 Outline

Chapter 2 provides a general background on SD-AVIC. The differences between feedforward control and feedback control are explained and it is explained when to use fixed gain control and when to use adaptive control. While fixed gain control relies on an off-line computed controller, adaptive control enables one to update the controller on a sample-by-sample basis. Finally, block diagrams of MIMO feedforward and feedback SD-AVIC systems are presented and an introduction to SMI is given.

Chapter 3 considers the Wiener controller, which is the best fixed gain controller in the sense of 2-norm minimization of the error sensor outputs. Furthermore, the chapter introduces the filtered-x least mean square adaptive algorithm for driving the coefficients of an adaptive controller. Also, the principle component least mean square algorithm is presented which is ideally suited for cancelling tonal disturbance signals. Furthermore, new hybrid algorithms are introduced.

Chapter 4 gives the SMI of the plants of the SR-SISO vibration isolation set-up and the SR-MIMO vibration isolation set-up. It is demonstrated that accurate linear models

can be found.

Chapter 5 describes feedforward narrowband simulation and real-time experiments on the SR-MIMO vibration isolation set-up. For control the filtered-x least mean square algorithm is used as well as the principle component least mean squares algorithm. Although the principle component least mean squares algorithm converges faster than the filtered-x least mean squares algorithm, both algorithms are able to successfully reject narrowband disturbances at the error sensor outputs in both simulation and real-time.

Chapter 6 presents broadband feedforward and feedback control experiments that are performed on the SR-SISO vibration isolation set-up and the SR-MIMO vibration isolation set-up. Both fixed gain and adaptive control experiments are performed. In all cases, the results of the adaptive and fixed gain control experiments match closely and the real-time and simulation experiments correspond well.

Chapter 2

Background on control and identification

2.1 Introduction

This chapter introduces the concepts of SD-AVIC using a 1DOF vibration isolation system as an example. Furthermore, background information on control and identification theory is presented.

The following topics are discussed in more detail:

- **Feedforward control and feedback control.** Feedforward control depends on feedforward reference inputs that are coherent with the disturbance outputs that need to be suppressed (Elliott, 2000*b*; Kuo and Morgan, 1996). This is in contrast to feedback control, which does not require feedforward reference inputs, but which depends on the error sensor outputs. Both control approaches are used in this thesis and it is important to understand the similarities and differences between them.
- **Fixed gain control and adaptive control.** A controller can be implemented as a fixed gain controller or as an adaptive controller. In the adaptive control strategy, the controller parameters are continuously updated, while in the fixed gain control strategy the controller parameters are computed in advance and are not (continuously) updated.
- **Multiple input multiple output active vibration isolation control systems.** In line with the remainder of this thesis, block diagrams are presented of MIMO feedforward and feedback SD-AVIC systems. Also, it is explained how the feedback control problem can be turned into a comparable feedforward one by using the internal model control (IMC) principle (Elliott, 2000*b*).
- **Identification of the transfer paths with subspace model identification.** In the last section of this chapter it is explained what particular plants are identified with SMI and, in a basic way, how SMI is performed. Note that only the fundamentals of SMI are presented. Furthermore, guidelines are provided for choosing an input/output data set that is suitable for SMI.

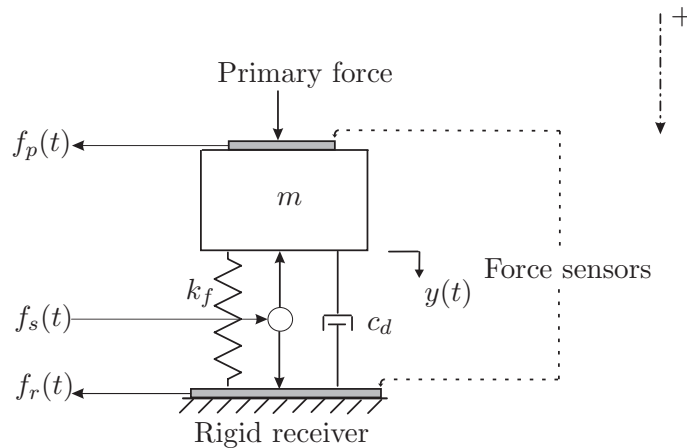


Figure 2.1: 1DOF MDS system.

Note that this chapter is only meant as an introduction. More comprehensive material can be found in many articles and textbooks. The following textbooks, for instance, are good starting points for studying active (vibration isolation) control theory (Nelson and Elliott, 1992; Kuo and Morgan, 1996; Widrow and Walach, 1996; Elliott, 2000*b*; Fuller et al., 1996; Hansen and Snyder, 1997).

This chapter is organized as follows. In section 2.2 feedforward control and feedback control are explained. Section 2.3 treats the fixed gain control and adaptive control strategies. Section 2.4 gives block diagrams of MIMO feedforward and feedback SD-AVIC systems and explains the IMC principle. Finally, in section 2.5 the basics of SMI are introduced.

2.2 Feedforward and feedback control

This section is organized as follows. Section 2.2.1 discusses the 1DOF MDS system, which is the starting point for the examination of feedforward control and feedback control. Feedforward control is explained in section 2.2.2 and feedback control in section 2.2.3.

Without ignoring the overall picture, this chapter focuses on *force* feedforward and feedback control. Specifically, it does not discuss *velocity* and *acceleration* feedforward and feedback control, since the goal is only to clarify the differences between feedforward and feedback control and not to explain the differences between the three methods. For a more detailed overview on force feedback, velocity feedback and acceleration feedback, see (Hansen and Snyder, 1997, chapter 12).

2.2.1 The one-degree-of-freedom mass/damper/spring system and the control objective

Figure 2.1 depicts a 1DOF MDS system. The dash-dotted arrow in the upper right corner of the figure (with the plus sign right next to it) denotes the positive direction. The rigid mass m (source) is excited by a primary force that can only work in the vertical direction. The primary force is assumed to be measured by a force sensor (depicted in the figure as a small dark rectangular area on top of the source structure) that outputs the electrical signal $f_p(t)$

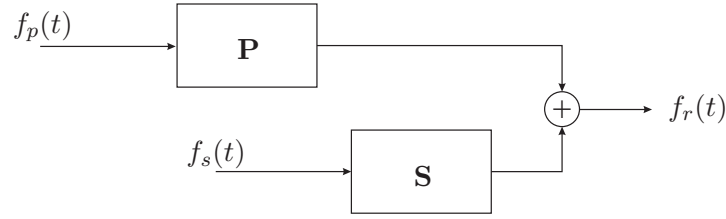


Figure 2.2: Block diagram of the 1DOF MDS system with an actuator.

that is directly related to the primary force. In the remainder of this thesis, this will simply be referred to as ‘the force $f_p(t)$ ’, even though $f_p(t)$ is not the force itself but a signal related to it. The force $f_p(t)$ results in a displacement $y(t)$ of the source and in a remaining force that acts on the immovable rigid receiver. This resulting force is assumed to be measured with a force sensor that outputs the electrical signal $f_r(t)$. In the remainder of this thesis, this will simply be referred to as ‘the resulting force $f_r(t)$ ’. The force $f_p(t)$ in SD-AVIC applications is referred to as the *disturbance input* or *reference input*, i.e. this is the true disturbance that is produced by the source. The reference input establishes the resulting force $f_r(t)$, which is referred to as the *disturbance output* in the active control context. To minimize the resulting force $f_r(t)$, PVI is provided by the spring with spring constant k_f and the damper with damper constant c_d . In addition, SD-AVIC is provided by the force $f_s(t)$. To be exact, $f_s(t)$ is a signal that serves as input to an actuator (the actuator is represented in the figure by a circle). Furthermore, $f_s(t)$ is computed by a controller that is not displayed in the figure. In this example the actuator is positioned between the source and the receiver, parallel to the spring and the damper. However, other placements of the actuator are possible. For example, the actuator can act exclusively on the source or the receiver (Hansen and Snyder, 1997). Without ignoring these other possibilities it is assumed here that the actuator is placed as it is. In figure 2.2 a block diagram is shown of the 1DOF MDS system depicted in figure 2.1. In the block diagram, \mathbf{P} represents the SISO transfer path from the force $f_p(t)$ to the resulting force $f_r(t)$ (assuming $f_s(t) = 0$). This transfer path is called the primary path. \mathbf{S} represents the SISO transfer path from the force $f_s(t)$ to the resulting force $f_r(t)$ (assuming $f_p(t) = 0$). This transfer path is called the secondary path¹. The goal of the SD-AVIC is to steer the force $f_s(t)$, so that:

$$f_r(t) = 0. \quad (2.1)$$

If this condition is met, disturbances induced by the force $f_p(t)$ are defined as being isolated from the receiver structure.

2.2.2 Feedforward control

It is assumed that the force $f_s(t)$ is generated by a controller that is denoted by the symbol \mathbf{W} from now on. The feedforward control approach assumes knowledge of the force $f_p(t)$ so that the force $f_s(t)$ can be computed. This is visualized in figure 2.3. A block diagram of the feedforward control system depicted in figure 2.3 is shown in figure 2.4. For the 1DOF MDS, the optimum - in the sense that $f_r(t) = 0$ - feedforward controller \mathbf{W} can be derived in an

¹Throughout this thesis \mathbf{P} and \mathbf{S} are always used to denote the primary path and secondary path, respectively.

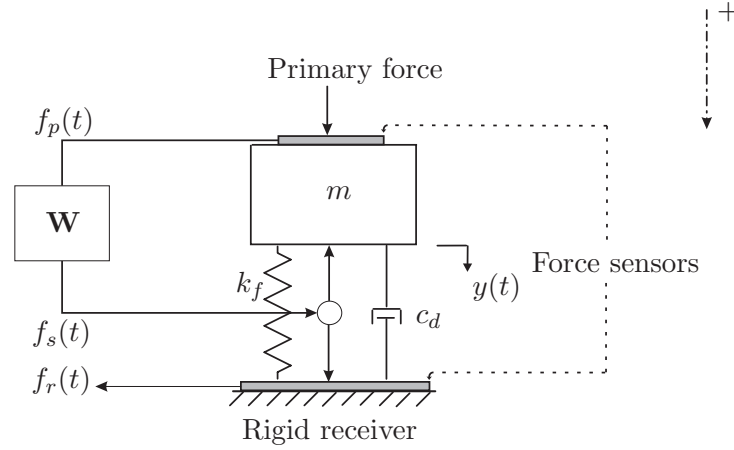


Figure 2.3: The 1DOF MDS system with an added feedforward controller to steer the actuator.

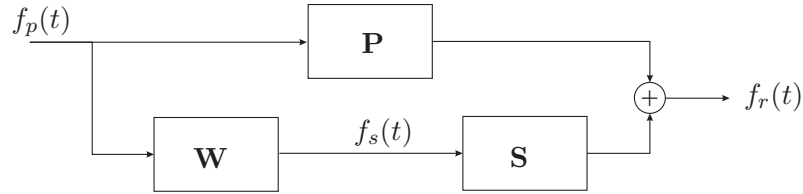


Figure 2.4: Block diagram of the 1DOF MDS system with an added feedforward controller to steer the actuator.

analytical fashion. This can be done by computing the force balance for the source and by computing the resulting force acting on the receiver (Hansen and Snyder, 1997):

$$\begin{aligned} \text{Force balance source:} \quad m\ddot{y}(t) + c_d\dot{y}(t) + k_f y(t) + f_s(t) &= f_p(t) \\ \text{Resulting force receiver:} \quad c_d\dot{y}(t) + k_f y(t) + f_s(t) &= f_r(t) \end{aligned}, \quad (2.2)$$

where $\{\dot{\cdot}\}$ and $\{\ddot{\cdot}\}$, represent the first and second time derivative, respectively. In the Laplace domain (Hansen and Snyder, 1997; Kwakernaak and Sivan, 1991), the two expressions in equation (2.2) can be written as:

$$\begin{aligned} ms^2y(s) + c_dsy(s) + k_fy(s) + f_s(s) &= f_p(s) \\ c_dsy(s) + k_fy(s) + f_s(s) &= f_r(s) \end{aligned}, \quad (2.3)$$

assuming that the initial conditions are zero. As stated in equation (2.1), the goal of the controller \mathbf{W} is to determine the force $f_s(t)$ (i.e. $f_s(s)$ in the Laplace domain) in such a way that the resulting force $f_r(t) = 0$, i.e. $f_r(s) = 0$. Substituting the condition $f_r(s) = 0$ in the second expression in equation (2.3), gives the following solution for $f_s(s)$:

$$f_s(s) = -(k_f + sc_d)y(s). \quad (2.4)$$

To determine the optimum controller, the force $f_s(s)$ in equation (2.4) needs to be expressed as a function of the force $f_p(s)$. To do so, the solution for $y(s)$, which can be determined from equation (2.4) and which is given by:

$$y(s) = -\frac{f_s(s)}{k_f + c_d s},$$

is substituted in the first expression in equation (2.3). This way, the optimum force $f_s(s)$ can be derived that needs to be executed by the actuator:

$$f_s(s) = -\frac{c_d s + k_f}{m s^2} f_p(s). \quad (2.5)$$

The feedforward controller \mathbf{W} is now given in the Laplace domain by:

$$\mathbf{W}(s) = -\frac{c_d s + k_f}{m s^2}. \quad (2.6)$$

The force $f_s(s)$ in equation (2.5) drives the resulting force $f_r(s)$ exactly to zero, i.e. perfect vibration isolation is established. The feedforward controller in equation (2.5) is not implemented in practice though, since it contains a repeated pole in the origin, i.e. a double integrator². See e.g. (El-Sinawa and Kashani, 2001) where the effect of the double integrator is explained. The aim here is however to demonstrate the difference between feedforward control and feedback control and not to highlight implementation issues. Furthermore, the controller \mathbf{W} can be used to derive an implementable stable controller that is valid for a particular frequency ω_1 only, i.e. assuming that $f_p(t)$ contains a single frequency component only, which results in:

$$f_s(j\omega_1) = \frac{c_d j\omega_1 + k_f}{\omega_1^2 m} f_p(j\omega_1).$$

The controller \mathbf{W} in equation (2.6) can also be computed in a different, more intuitive, manner. From the expressions in equation (2.3) the primary path \mathbf{P} and the secondary path \mathbf{S} can be derived in the Laplace domain by setting $f_s(s) = 0$ and $f_p(s) = 0$, respectively. The secondary path \mathbf{S} is given by:

$$\mathbf{S}(s) = \frac{f_r(s)}{f_s(s)} = \frac{m s^2}{m s^2 + c_d s + k_f}. \quad (2.7)$$

The primary path \mathbf{P} is given by:

$$\mathbf{P}(s) = \frac{f_r(s)}{f_p(s)} = \frac{c_d s + k_f}{m s^2 + c_d s + k_f}. \quad (2.8)$$

Obviously, if $f_p(s) \neq 0$, to achieve that:

$$f_r(s) = \{\mathbf{P}(s) + \mathbf{S}(s)\mathbf{W}(s)\} f_p(s) = 0,$$

²Of course, in a real-life application there is always the possibility to ‘regularize’ the controller $\mathbf{W}(s)$ and to slightly push the poles away from the origin, so that the following expression remains for $\check{f}_s(s)$ (where the breve on top of ‘f’ refers to the regularized solution for ‘f’):

$$\check{f}_s(s) = -\frac{c_d s + k_f}{m(s + \epsilon)^2} f_p(s),$$

with $\epsilon \in \mathbb{R}^+$ a small positive constant.

it follows that:

$$\mathbf{P}(s) + \mathbf{S}(s)\mathbf{W}(s) = 0.$$

Therefore the following equation applies to the controller $\mathbf{W}(s)$:

$$\mathbf{W}(s) = -\mathbf{S}^{-1}(s)\mathbf{P}(s) = -\frac{ms^2 + c_d s + k_f}{ms^2} \frac{c_d s + k_f}{ms^2 + c_d s + k_f} = -\frac{c_d s + k_f}{ms^2},$$

which immediately yields the result for $\mathbf{W}(s)$ from equation (2.6). Note that $\{\cdot\}^{-1}$ indicates the inverse.

Considerations in the application of feedforward control

Feedforward control can be a solution if the reference input ($f_p(t)$ in our example) can be measured and if³:

- *the reference input is deterministic and thus predictable.* The reference input can be a single sinusoidal signal, for instance. In real-life applications information about the reference input can be constructed from a number of sources. For example, in rotating machines it can be constructed from the tachometer indicating the number of rotations of the machine shaft per time unit.
- *The reference input is random and thus unpredictable.* In that case, however, the active vibration isolator has to be sufficiently removed from the source so that the feedforward controller \mathbf{W} has time to respond to the reference input. In practice though, the active vibration isolator is often close to the source and the feedforward controller \mathbf{W} has no time to respond to the reference input, i.e. the controller action is too late. Specifically, if the primary path \mathbf{P} features a delay Δ_P , a response on the input affects the error sensor output after Δ_P samples⁴. ‘Ideally’, in the case of random disturbances, proper feedforward control can be established if the delay Δ_{RT} in the real-time control system meets the criterium $\Delta_{RT} \leq \Delta_P - \Delta_S$, with Δ_S the delay in the secondary path \mathbf{S} . See figure 2.5 in which the delays are depicted. Otherwise, if this condition is not met, a decrease of the controller \mathbf{W} is to be expected, since the controller action is too late.

Benefits of feedforward control include the fact that it is theoretically able to achieve perfect control. Feedback control, which is discussed later, is theoretically not able to achieve perfect control due to the fact that the controller action is always too late. Another benefit is that - assuming a stable primary path \mathbf{P} , a stable secondary path \mathbf{S} and a stable feedforward controller \mathbf{W} - a feedforward control system always results in a bounded (stable) error sensor output. On the other hand, applying feedforward control is not always possible since it requires knowledge of the feedforward reference input (Doelman, 1993). That knowledge cannot always be obtained (easily) and, therefore, a sensory system may be necessary to do so. However, implementation of such a system can be expensive. If the feedforward reference input cannot be obtained, feedback control can be an alternative.

³Note that the focus here is only on the reference signal and that the controller bandwidth, actuator limits, causality constraints and so on are not discussed at length. A later chapter will deal with some of these issues in more detail.

⁴Especially in active noise control systems the delay Δ_P in the primary path \mathbf{P} can be significant in comparison to the delay Δ_S in the secondary path \mathbf{S} due to the ‘low’ speed of sound.

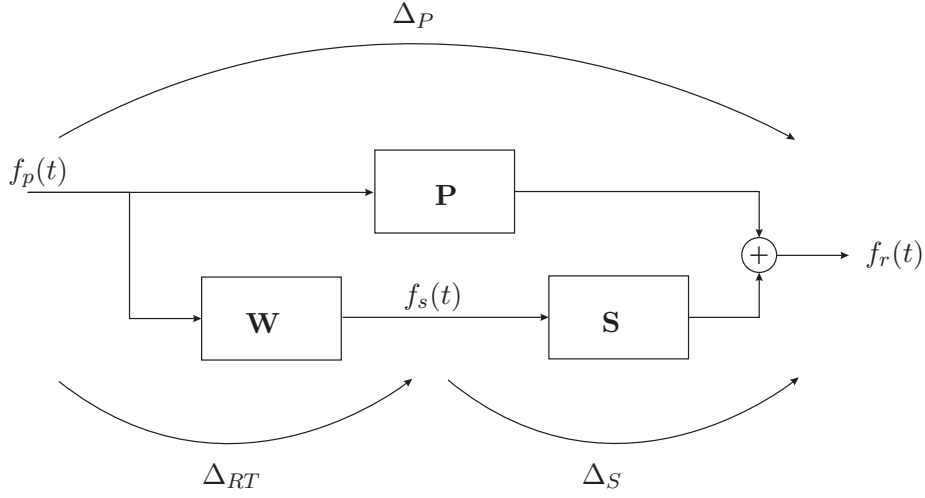


Figure 2.5: Block diagram of the 1DOF MDS system with a feedforward controller indicating the delays.

2.2.3 Feedback control

Instead of deriving the optimum force $f_s(t)$ from the force $f_p(t)$ as in the feedforward control approach, in the feedback control approach the force $f_s(t)$ is derived from the resulting force $f_r(t)$, which is visualized in figure 2.6. The block diagram corresponding with the feedback control system shown in figure 2.6 is depicted in figure 2.7. As in the feedforward control approach, a feedback controller \mathbf{W} is computed analytically. Without ignoring the fact that there are other possibilities, here the case is considered where the total force $f_r(t)$ transmitted into the receiver is the feedback variable. Furthermore, the force $f_s(t)$ is generated by a feedback controller \mathbf{W} , which is a scalar feedback gain K_f , i.e. $\mathbf{W} = K_f$. Other feedback controllers can be considered though, see e.g. (Hansen and Snyder, 1997). The resulting closed loop system can be derived from the equations describing the force balance for the source and the resulting force acting on the receiver, respectively:

$$\begin{aligned} \text{Force balance source:} \quad m\ddot{y}(t) + c_d\dot{y}(t) + k_f y(t) - K_f f_r(t) &= f_p(t) \\ \text{Resulting force receiver:} \quad c_d\dot{y}(t) + k_f y(t) - K_f f_r(t) &= f_r(t) \end{aligned} \quad , \quad (2.9)$$

which are very similar to the expressions in equation (2.2). In the Laplace domain, the two expressions in equation (2.9) can be written as:

$$\begin{aligned} ms^2y(s) + c_dsy(s) + k_f y(s) - K_f f_r(s) &= f_p(s) \\ c_dsy(s) + k_f y(s) - K_f f_r(s) &= f_r(s) \end{aligned} \quad , \quad (2.10)$$

assuming that the initial conditions are zero. The closed loop system $\mathbf{K}(s)$ from the force $f_p(t)$ to the resulting force $f_r(t)$ can be derived in the Laplace domain from the two expressions in equation (2.10) and is given by:

$$\mathbf{K}(s) = \frac{f_r(s)}{f_p(s)} = \frac{\mathbf{P}(s)}{1 + \mathbf{S}(s)\mathbf{W}(s)} = \frac{c_d s + k_f}{m(1 + K_f)s^2 + c_d s + k_f}. \quad (2.11)$$

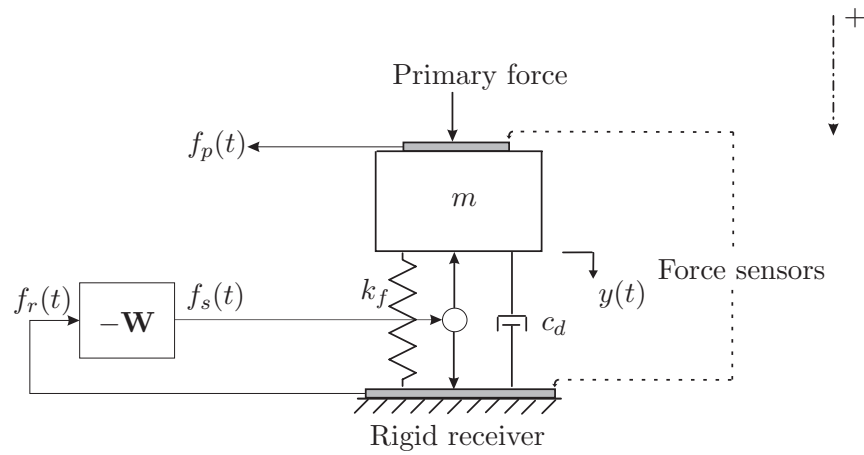


Figure 2.6: The 1DOF MDS system with an added feedback controller for SD-AVIC.

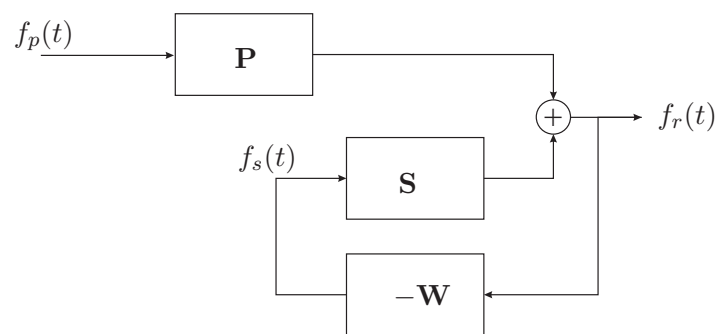


Figure 2.7: Block diagram of the 1DOF MDS system with an added feedback controller.

Table 2.1: Sizes of the constants used for the feedback gain simulation experiments with the 1DOF MDS system.

Item	Size	Unit
Rigid mass m	0.1	kg
Spring constant k_f	10	Nm ⁻¹
Damper constant c_d	0.01	Nsm ⁻¹

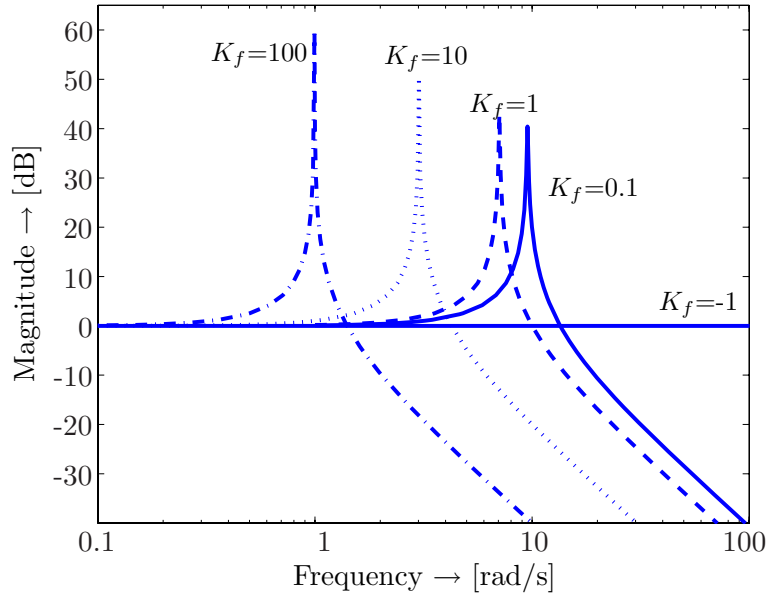


Figure 2.8: Magnitude of the closed loop response $\mathbf{K}(s)$ described by equation (2.11) for various values of the force feedback gain K_f .

Equation (2.11) shows that the stability of the closed loop system $\mathbf{K}(s)$ can be influenced by adjusting the gain K_f , i.e. the poles of the closed loop system $\mathbf{K}(s)$ change if K_f is changed. In figure 2.8 the influence of various values of the feedback gain K_f is illustrated. Table 2.1 shows what sizes are used for the mass, damper and spring. It can be seen that the magnitude of the resonance peak increases with increasing K_f . Obviously, if $K_f = -1$, the closed loop equals 1, which results in a straight line at 0 dB in figure 2.8. For this particular problem, if $K_f < -1$, instability is established, which is not desirable. Therefore, the gain K_f should not be set below -1. From the figure it is clear that the larger the feedback gain K_f that is chosen, the larger the bandwidth over which vibration isolation control is achieved. Thus, the vibration isolation performance increases. However, in this example this goes at the expense of the single resonance peak that becomes more dominant. Reducing the resonance peak can be accomplished by changing the size of the damper constant c_d and the spring constant k_f . However, that is not shown here and would result in less PVI. Finally it should be noted that force feedback control can be considered as an increase of the *effective* mass. Specifically, see equation (2.11): instead of solely having m in the denominator, now the term $m(1 + K_f)$ appears.

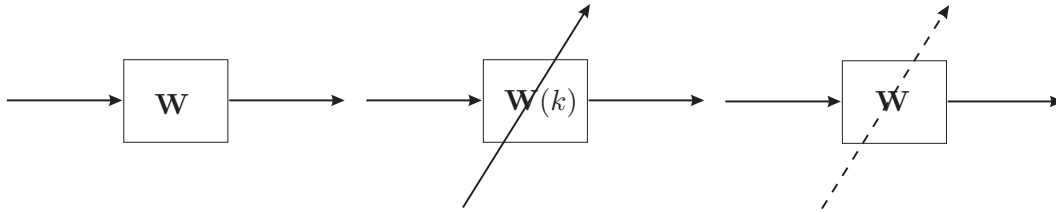


Figure 2.9: Fixed gain controller that is not updated online (left). Adaptive controller that is updated online on a sample-by-sample basis (middle). Controller that can be both fixed gain and adaptive (right).

Considerations in the application of feedback control

If the delay Δ_P in the primary path \mathbf{P} is large in comparison with the delay Δ_S in the secondary path \mathbf{S} and the delay Δ_{RT} in the real-time control system, it is better to look at the possibility of feedforward control. That is, assuming that the reference input can be obtained. If these two conditions are not met, feedback control can be a solution. However, feedback control has two pronounced disadvantages that make it less attractive than feedforward control. These are:

- the tendency to get an unstable closed loop for a feedback *loop* gain that is set too high. See for example (Hansen and Snyder, 1997, Section 5.5.3) for a detailed explanation.
- the feedback controller does not react to a disturbance output before a control error has occurred. See e.g. (Elliott, 2000b, chapter 6) and (Doelman, 1993). For that reason, feedback control will commonly yield limited performance as compared to feedforward control.

2.3 Fixed gain control versus adaptive control

The feedforward/feedback controller can be designed as a fixed gain controller or an adaptive controller. Fixed gain controllers are computed in advance (off-line) and do not have the ability to change the coefficients if necessary, while adaptive controllers are working online and are able to change the coefficients while in operation. The symbol for a fixed gain controller is depicted on the left in figure 2.9; the symbol in the middle represents an adaptive controller, and the one on the right is used to indicate a controller that can be either fixed gain or adaptive. The latter symbol is used in certain block diagrams if the nature of the controller is not yet set. In the figure the normalized⁵ discrete time sample instant k is used instead of the continuous time symbol t , since the adaptive controller is implemented on digital hardware. k is used to be consistent with the remainder of the thesis. Furthermore, two types of adaptive controllers can be distinguished:

- **Indirect adaptive controllers.** An indirect adaptive controller can be considered a fixed gain controller that is updated from time to time. Figure 2.10 depicts a block diagram of a particular indirect adaptive feedforward controller applied to the 1DOF

⁵Normalization actually means that the notation kT should be used instead of just k , but here kT is simply abbreviated to k . Thus, k is the normalized sample number.

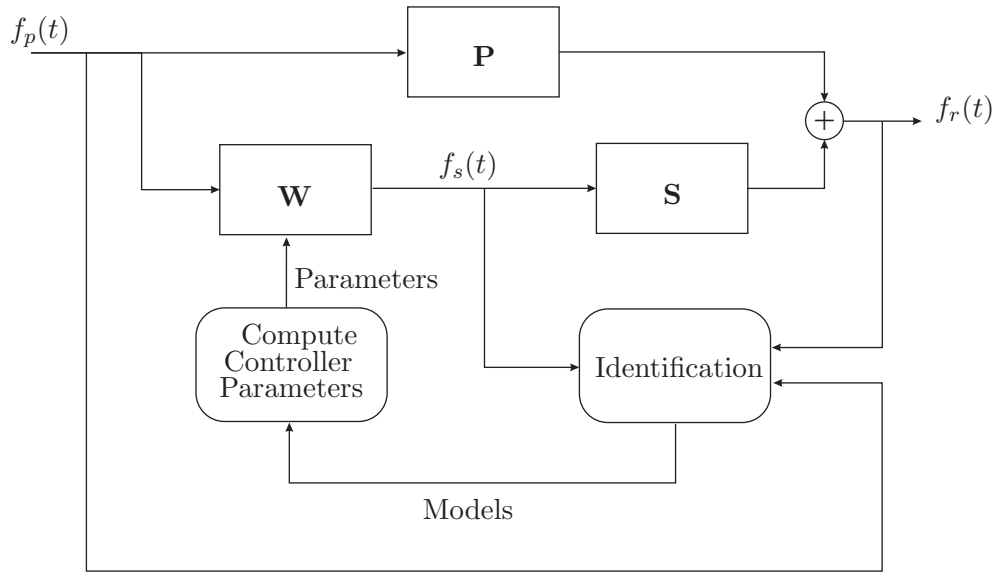


Figure 2.10: Block diagram of the 1DOF MDS system with an added feedforward indirect adaptive controller to steer the actuator.

MDS system. This thesis does not treat indirect adaptive controllers. The interested reader is referred to (Schofield, 2003), where an indirect adaptive controller is applied in combination with SMI.

- **Direct adaptive controllers.** These types of controllers compute the controller parameters directly and continuously. Required models of the plant under control are not necessarily updated. In figure 2.11 it is visualized how direct adaptive feedforward control takes place on the 1DOF MDS system. As indicated in the figure, the mechanism that updates the feedforward adaptive controller commonly uses information from the reference input $f_p(t)$ and the resulting force $f_r(t)$. In the remainder of this thesis the focus is on direct adaptive controllers. For brevity's sake, the term *adaptive control(ler)* is used instead of *direct adaptive control(ler)*.

An adaptive mechanism is preferable especially for feedforward controllers, so that the controller can adapt to changes in the reference input and the plant responses. For feedback control it is desirable to use a fixed gain controller, so that it is 'known' in advance that a stable closed loop is obtained. However, although a fixed gain controller is suited to the feedback control approach, an *adaptive* controller is usually chosen to compensate for plant and signal variations.

There are, however, two disadvantages to applying adaptive control. An adaptive controller:

- *results in extra computations that need to be performed by the fixed or floating point arithmetic on which the controller is executed.* The computational burden is especially pronounced if the adaptive controller is updated on a sample-by-sample basis. Obviously, a higher sample frequency results in a higher processor load.

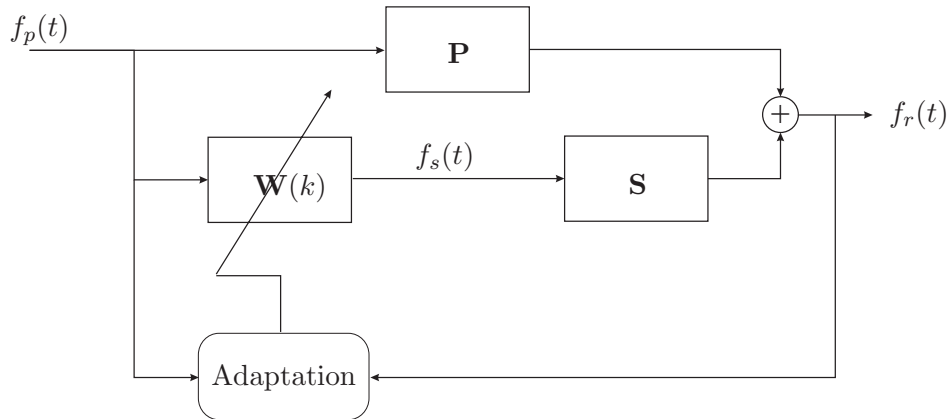


Figure 2.11: A block diagram of the 1DOF MDS system with an added feedforward direct adaptive controller to steer the actuator.

- *requires time to converge to the proper solution.* Although in some applications the convergence speed is less critical, in other cases fast convergence is required. For example, assuming feedforward control, if the vibration energy from a car's engine needs to be isolated from the body of the car, fast convergence is required, since one does not want to wait (for example) several minutes (or hours) before the controller performs properly.

2.4 Multiple input multiple output active vibration isolation control systems

In this section the presentation is generalized and block diagrams are presented of MIMO feedforward and feedback SD-AVIC systems. Furthermore, the focus is on the discrete time domain.

This section is organized as follows. In section 2.4.1 block diagrams are considered of MIMO feedforward and feedback SD-AVIC systems. In section 2.4.2 it is shown how MIMO feedback SD-AVIC systems can be transformed to equivalent MIMO feedforward SD-AVIC systems by using the IMC principle (Elliott, 2000b).

2.4.1 Block diagrams of feedforward and feedback active vibration isolation control systems

Feedforward control. Figure 2.12 shows a block diagram of a MIMO feedforward SD-AVIC system with J reference inputs, I controller inputs, K actuator inputs and M error sensor outputs. \mathbf{P} represents⁶ the asymptotically stable⁷. J input M output transfer path from the reference inputs $\mathbf{x}(k) \in \mathbb{R}^J$ to the disturbance outputs $\mathbf{d}(k) \in \mathbb{R}^M$, i.e. the primary path. The signal $\mathbf{d}(k)$ is given by:

$$\mathbf{d}(k) = \mathbf{P}\mathbf{x}(k).$$

⁶Although the MIMO case is treated in this section, the same symbols are used for the plants and the controller as in the SISO case.

⁷All poles strictly inside the unit circle. This is usually the case for plants in active noise and vibration control (Fraanje, 2004).

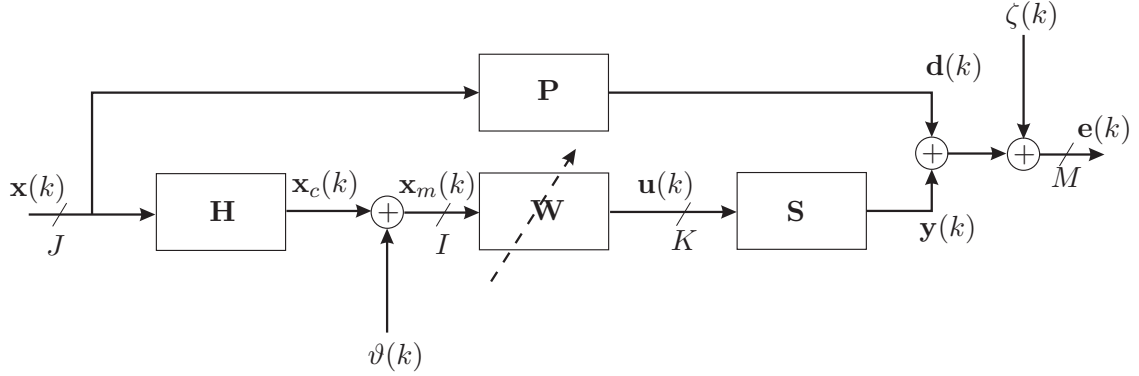


Figure 2.12: Block diagram representing a MIMO feedforward SD-AVIC system. Signals entering and leaving a plus sign ($+$) have the same dimension.

In the broadband control experiments described in this thesis it is assumed that the reference inputs $\mathbf{x}(k)$ are zero mean stochastic white noise sequences. \mathbf{S} is the asymptotically stable K input M output transfer path from the actuator inputs $\mathbf{u}(k) \in \mathbb{R}^K$ to the anti-disturbance outputs $\mathbf{y}(k) \in \mathbb{R}^M$, i.e. the secondary path. Therefore:

$$\mathbf{y}(k) = \mathbf{S}\mathbf{u}(k).$$

\mathbf{H} is the asymptotically stable J input I output transfer path from the reference inputs $\mathbf{x}(k) \in \mathbb{R}^J$ to the filtered reference inputs $\mathbf{x}_c(k) \in \mathbb{R}^I$, i.e. the primary detector path. Therefore, the signal $\mathbf{x}_c(k)$ is given by:

$$\mathbf{x}_c(k) = \mathbf{H}\mathbf{x}(k).$$

\mathbf{W} is the asymptotically stable I input K output feedforward controller from the measured reference inputs $\mathbf{x}_m(k) \in \mathbb{R}^I$ to the actuator inputs $\mathbf{u}(k)$, which means that:

$$\mathbf{u}(k) = \mathbf{W}\mathbf{x}_m(k),$$

where:

$$\mathbf{x}_m(k) = \mathbf{x}_c(k) + \vartheta(k),$$

with $\vartheta(k) \in \mathbb{R}^I$ the measurement noise, which can be colored but is uncorrelated to the reference inputs $\mathbf{x}(k)$. The error sensor outputs are given by $\mathbf{e}(k) \in \mathbb{R}^M$ and are a superposition⁸ of the disturbance outputs $\mathbf{d}(k)$, the anti-disturbance outputs $\mathbf{y}(k)$ and the measurement noise $\zeta(k)$:

$$\mathbf{e}(k) = \mathbf{d}(k) + \mathbf{y}(k) + \zeta(k). \quad (2.12)$$

The measurement noise $\zeta(k)$ is assumed to be uncorrelated to the reference inputs $\mathbf{x}(k)$ and the disturbance outputs $\mathbf{d}(k)$. The anti-disturbance outputs $\mathbf{y}(k)$ are generated by the feedforward controller \mathbf{W} . It is also assumed that $M \geq K, J$ and $K \geq I \geq J$.

The primary detector path \mathbf{H} finds its origin in the fact that usually a transfer path exists between the actual disturbance input and the one that is measured. In this thesis it is assumed

⁸Some authors use a minus, but here a plus is used to give $\mathbf{d}(k) + \mathbf{y}(k)$, since in real life signals are always added.

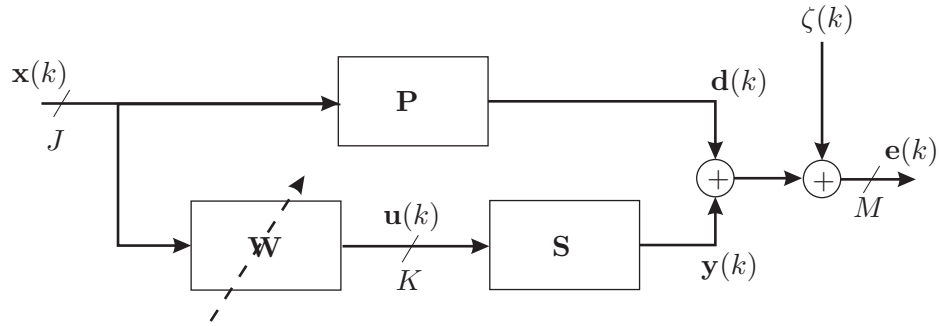


Figure 2.13: Block diagram representing a MIMO feedforward SD-AVIC system with a unity primary detector path \mathbf{H} .

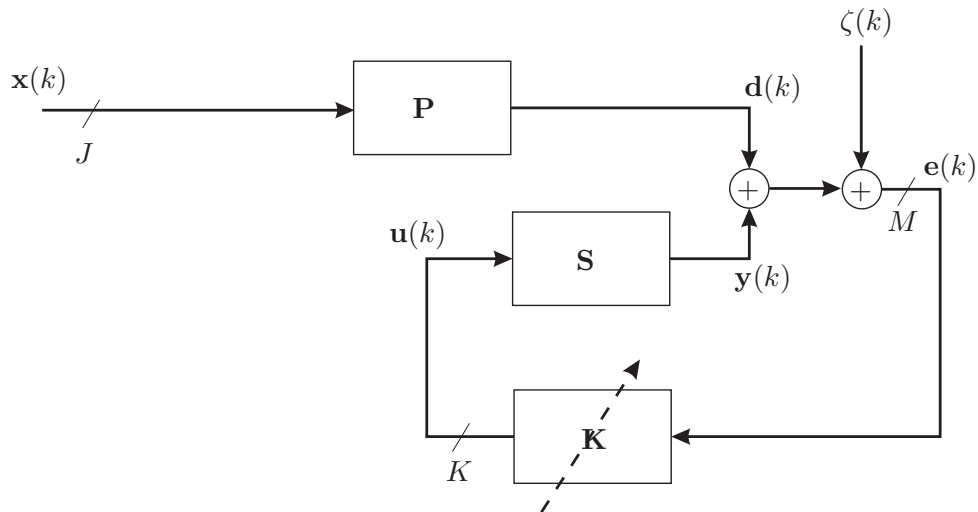


Figure 2.14: Block diagram representing a MIMO feedback SD-AVIC system.

that the primary detector path \mathbf{H} is a unity gain in the feedforward control approach. This is because a computer generated reference input $\mathbf{x}(k)$ is used and no measurement of $\mathbf{x}(k)$ is necessary. The block diagram shown in figure 2.12 then reduces to the block diagram depicted in figure 2.13. The main reason that the primary detector path \mathbf{H} and the measurement noise $\vartheta(k)$ are already included in figure 2.12 is that they become important in the feedback control approach when the IMC principle is used.

Feedback control. A block diagram of a MIMO feedback SD-AVIC system is shown in figure 2.14. The feedback controller \mathbf{K} is driven by the error sensor outputs $\mathbf{e}(k)$. Thus, it is true that:

$$\mathbf{u}(k) = \mathbf{K}\mathbf{e}(k).$$

The next section elaborates on the feedback control approach by means of the IMC principle.

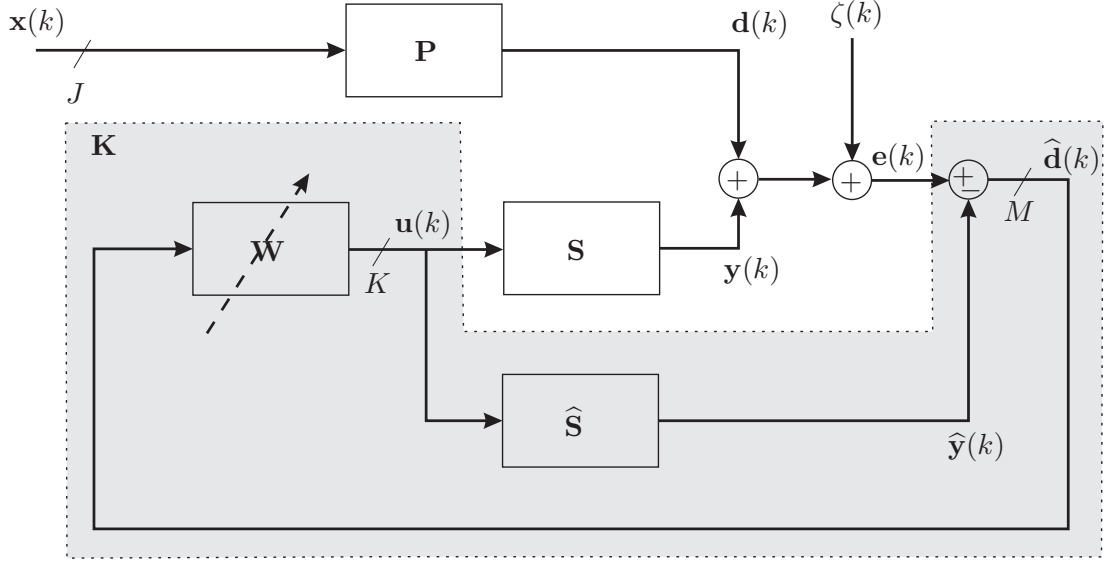


Figure 2.15: Block diagram representing a MIMO feedback SD-AVIC system making use of the IMC principle.

2.4.2 The internal model control principle

The IMC principle is explained by means of the block diagram shown in figure 2.15. The main idea behind IMC is to obtain an accurate estimate $\hat{\mathbf{d}}(k)$ of the primary disturbance $\mathbf{d}(k)$ to serve as an input for the controller \mathbf{W} which is embedded in the total closed loop controller \mathbf{K} . The estimate $\hat{\mathbf{d}}(k)$ is obtained as follows. $\hat{\mathbf{S}}$ is an appropriate model of the secondary path \mathbf{S} (this is the internal model to which the name IMC refers). The inputs to the model $\hat{\mathbf{S}}$ are the actuator inputs $\mathbf{u}(k)$. The outputs $\hat{\mathbf{y}}(k)$ of the model $\hat{\mathbf{S}}$ are estimates of the anti-disturbance outputs $\mathbf{y}(k)$:

$$\hat{\mathbf{y}}(k) = \hat{\mathbf{S}}\mathbf{u}(k).$$

Subsequently, the estimated anti-disturbances $\hat{\mathbf{y}}(k)$ are subtracted from the error sensor outputs $\mathbf{e}(k)$, which directly results in the desired estimates:

$$\hat{\mathbf{d}}(k) = \mathbf{e}(k) - \hat{\mathbf{y}}(k). \quad (2.13)$$

The controller \mathbf{W} is now driven by the estimates of the disturbance outputs $\hat{\mathbf{d}}(k)$:

$$\mathbf{u}(k) = \mathbf{W}\hat{\mathbf{d}}(k).$$

The complete feedback controller \mathbf{K} is given by the gray area that inputs $\mathbf{e}(k)$ and outputs $\mathbf{u}(k)$:

$$\mathbf{K} = \left(\mathbf{I}_K + \mathbf{W}\hat{\mathbf{S}} \right)^{-1} \mathbf{W}, \quad (2.14)$$

where \mathbf{I}_K is an identity matrix of size K . If \mathbf{W} is a fixed gain controller and \mathbf{S} is stable, \mathbf{K} in equation (2.14) is known as the Youla parametrization of all stabilizing controllers (Youla et al., 1976a,b). \mathbf{W} is denoted as the Youla parameter. To see that the feedback control

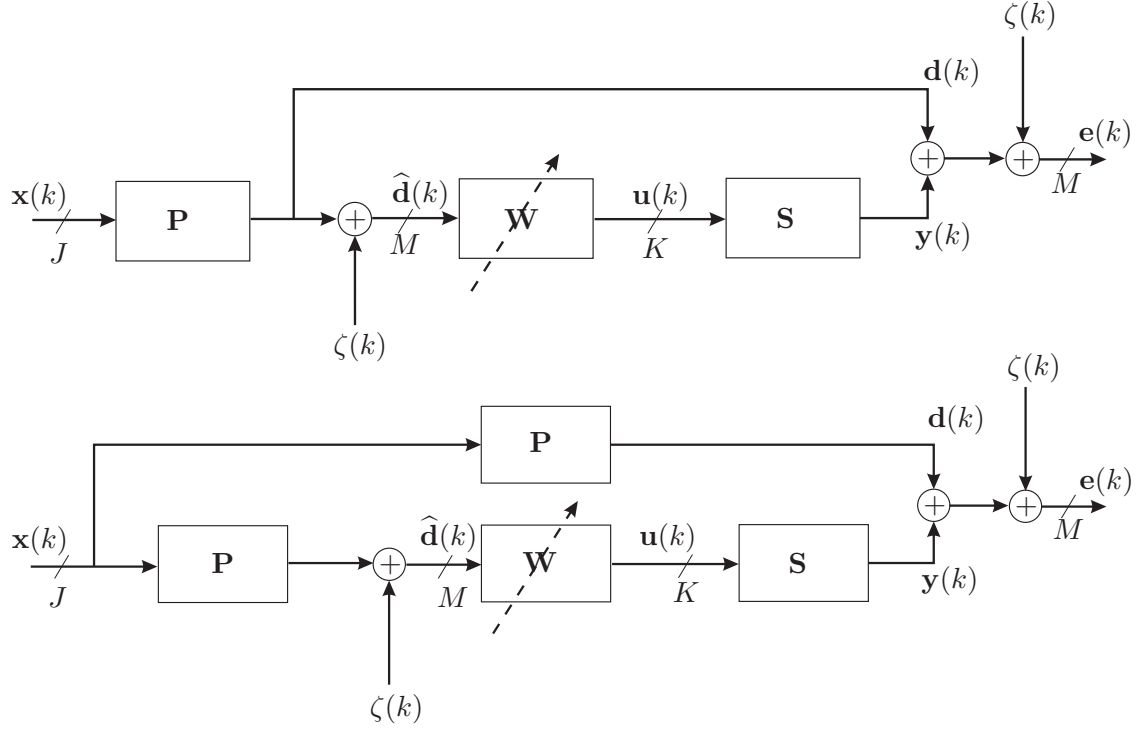


Figure 2.16: Upper: Block diagram representing a MIMO feedback SD-AVIC system making use of the IMC principle as shown in figure 2.15 if the model $\hat{\mathbf{S}}$ is exactly equal to \mathbf{S} . Lower: The same as the upper figure, but with the primary path \mathbf{P} shown in a different location.

problem can be transformed to an equivalent feedforward one, suppose the model $\hat{\mathbf{S}}$ is perfect and thus $\hat{\mathbf{S}} = \mathbf{S}$, then (see equation (2.13)):

$$\hat{\mathbf{d}}(k) = \mathbf{e}(k) - \hat{\mathbf{y}}(k) = \mathbf{d}(k) + \zeta(k) + \mathbf{y}(k) - \hat{\mathbf{y}}(k) = \mathbf{d}(k) + \zeta(k).$$

This result implies that the block diagram shown in figure 2.15 can be turned into the upper block diagram depicted in figure 2.16, which immediately shows the feedforward character of the feedback controller. The primary path \mathbf{P} can be put in a different position to make things more clear. This is shown in the lower picture in figure 2.16. The lower block diagram in figure 2.16 is identical to the block diagram shown in figure 2.12, with $\mathbf{H} = \mathbf{P}$, $\vartheta(k) = \zeta(k)$ and some different dimensions, of course. By using the block diagram of the MIMO feedforward SD-AVIC system in figure 2.12 and the block diagram of the MIMO feedback SD-AVIC system in figure 2.16, the actual basis is formed for the controller design presented in the next chapter. But first it is described how models of the plant under control are established.

2.5 Black-box identification of the plants

Figure 2.17 shows a block diagram in which the two main transfer paths are given that are present in a MIMO feedforward and feedback SD-AVIC system⁹. For control purposes,

⁹If figure 2.17 is compared with figure 2.12, one can conclude that the primary detector path \mathbf{H} is absent in figure 2.17. This is because the primary detector path \mathbf{H} is not needed in the feedforward control approach. In the feedback control approach, the primary detector path \mathbf{H} is equal to the primary path \mathbf{P} (see figure 2.16).

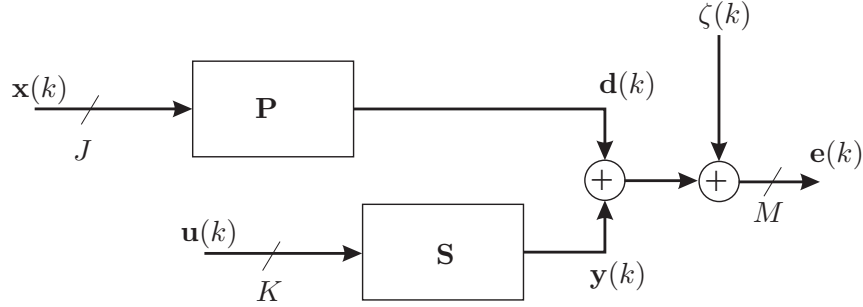


Figure 2.17: The two main transfer paths present in an SD-AVIC system.

models of these two plants are needed. The goal of this section is to clarify how models can be obtained of:

1. the primary path \mathbf{P} from the reference inputs $\mathbf{x}(k)$ to the disturbance outputs $\mathbf{d}(k)$.
2. the secondary path \mathbf{S} from the actuator inputs $\mathbf{u}(k)$ to the anti-disturbance outputs $\mathbf{y}(k)$.

In this section the basics of SMI are highlighted. See (Van Overschee and De Moor, 1994; Viberg, 1995; Verhaegen, 1993, 1994; Verhaegen and Dewilde, 1992; Haverkamp, 2001) for details on SMI.

This section is organized as follows. Section 2.5.1 describes the assumptions that are made concerning SMI. In section 2.5.2 it is explained how the quality of an identified model is measured. The measurements are used throughout the thesis and will become particularly useful in chapter 4 which describes the identification results. Finally, in section 2.5.3 guidelines are explained to obtain a proper data set for identification.

2.5.1 Subspace model identification

Assume that the input/output relationship from the inputs $\tilde{\mathbf{u}}(k) = [\mathbf{x}^T(k) \ \mathbf{u}^T(k)]^T \in \mathbb{R}^{J+K}$ to the error sensor outputs $\mathbf{e}(k)$ is given by the following state space description (see figure 2.17):

$$\chi \sim \begin{cases} \mathbf{z}(k+1) &= \mathbf{A}\mathbf{z}(k) + \mathbf{B}\tilde{\mathbf{u}}(k) + \psi(k) \\ \mathbf{e}(k) &= \mathbf{C}\mathbf{z}(k) + \mathbf{D}\tilde{\mathbf{u}}(k) + \iota(k) \end{cases}, \quad (2.15)$$

with $\mathbf{z}(k) \in \mathbb{R}^N$ the state of the system and $\mathbf{e}(k) \in \mathbb{R}^M$ the output of the system. The matrices \mathbf{A} , \mathbf{B} , \mathbf{C} and \mathbf{D} are called the system matrices and have appropriate dimension (see appendix A for more information on state space descriptions). $\psi(k) \in \mathbb{R}^N$ and $\iota(k) \in \mathbb{R}^M$ are the process noise and measurement noise, respectively¹⁰. The process noise $\psi(k)$ and measurement noise $\iota(k)$ are assumed to be zero mean stationary white noise sequences, statistically independent from the input $\tilde{\mathbf{u}}(k)$, with covariance matrix:

$$\mathbf{E} \left(\begin{bmatrix} \iota(k) \\ \psi(k) \end{bmatrix} \begin{bmatrix} \iota^T(l) & \psi^T(l) \end{bmatrix} \right) = \begin{bmatrix} \mathbf{R} & \mathbf{Z} \\ \mathbf{Z}^T & \mathbf{Q} \end{bmatrix} \delta(k-l), \quad (2.16)$$

¹⁰See (Fraanje, 2004) for an explanation of how the process noise $\psi(k)$ and measurement noise $\iota(k)$ in equation (2.15) can be ‘combined’ in the measurement noise $\zeta(k)$.

where δ is the unit operator which is one for $k = l$ (see appendix A). E denotes the statistical expectation operator (Leon-Garcia, 1994) and $[\cdot]^T$ represents the transpose. The matrices \mathbf{Q} , \mathbf{R} and \mathbf{Z} have appropriate dimension. See e.g. (Sayed, 2003, Section 1.3.3) for details on the covariance matrix.

The main objective of the identification procedure is to identify the:

1. order N of the system χ ;
2. system matrices \mathbf{A} , \mathbf{B} , \mathbf{C} and \mathbf{D} of the system χ up to a similarity transformation.

With SMI it is also possible to obtain estimates $\hat{\mathbf{Q}}$, $\hat{\mathbf{R}}$ and $\hat{\mathbf{Z}}$ of the covariance information \mathbf{Q} , \mathbf{R} and \mathbf{Z} as given in equation (2.16). Using the estimates $\hat{\mathbf{Q}}$, $\hat{\mathbf{R}}$ and $\hat{\mathbf{Z}}$, a stationary Kalman gain can be computed to develop an optimal minimum variance state-predictor (Kailath et al., 2000; Dorf and Bishop, 1998; Haverkamp, 2001). Although this is possible, the main interest here is in the two points mentioned above. The identification of the system χ is done based on a measured input/output data set. The actuator inputs $\tilde{\mathbf{u}}(k)$ are excited with a particular input sequence and the error sensor outputs $\mathbf{e}(k)$ are recorded. Based on the obtained data set, the required quantities are estimated.

For the identification the Fortran-based SLICOT routines are used (NICONET, 2000). Other software is available, however, that implements SMI routines, e.g. the standard Matlab system identification toolbox (Ljung, 1999) or the SMI toolbox (Haverkamp et al., 1997). Here the SLICOT routines (NICONET, 2000) are chosen since they are fast and numerically robust mex-based routines for Matlab (Redfern and Campbell, 1998).

2.5.2 Determining the model quality of an identified model

The quality of an identified model is tested by means of two model accuracy measures: the variance-accounted-for (VAF) and the signal-to-noise-ratio (SNR).

Variance-accounted-for. With the VAF, two matrices $\mathbf{Y} \in \mathbb{R}^{\mathcal{N} \times \mathcal{M}}$ and $\hat{\mathbf{Y}} \in \mathbb{R}^{\mathcal{N} \times \mathcal{M}}$ can be compared with each other. The VAF is expressed as a percentage (%): the VAF is 100% if the matrices \mathbf{Y} and $\hat{\mathbf{Y}}$ are identical. If the matrices are not identical, the VAF has a value between 0% and 100%. The VAF is defined as:

$$\text{VAF} = \frac{1}{\mathcal{M}} \sum_{m=1}^{\mathcal{M}} \left[\max \left\{ 1 - \frac{\|\mathbf{Y}_m - \hat{\mathbf{Y}}_m\|_2^2}{\|\mathbf{Y}_m\|_2^2}, 0 \right\} \times 100\% \right], \quad (2.17)$$

where ‘max’ denotes maximum value. The index $m = 1, \dots, \mathcal{M}$ refers to the m^{th} column of the corresponding matrix. In the case of system identification, \mathbf{Y} is often the validation output sequence of the system under consideration (\mathcal{N} samples long and \mathcal{M} outputs wide). The validation output sequence \mathbf{Y} is established by ‘feeding’ the true system with the validation input sequence. The estimate $\hat{\mathbf{Y}}$ is the estimated validation output sequence that is established by simulating the *model* with the validation input sequence. Obviously, if the model and the initial state of the model are perfect, the matrices \mathbf{Y} and $\hat{\mathbf{Y}}$ are identical and the VAF is 100%.

Signal-to-noise-ratio. The second method to test the model accuracy is the signal-to-noise-ratio (SNR) which is expressed in dB. The SNR is used in addition to the VAF since

more readers may be familiar with the SNR. Equivalent to the VAF in equation (2.17), with the SNR two matrices $\mathbf{Y} \in \mathbb{R}^{N \times \mathcal{M}}$ and $\hat{\mathbf{Y}} \in \mathbb{R}^{N \times \mathcal{M}}$ can be compared. There are different *expressions* of the SNR available for different applications, e.g. in communications (II, 1993) and electronics (Wissenburg, 1991). Here, the SNR is directly related to the VAF in equation (2.17) and is defined as:

$$\text{SNR} = \frac{1}{\mathcal{M}} \sum_{m=1}^{\mathcal{M}} \left[10 \log_{10} \left\{ \frac{\|\mathbf{Y}_m\|_2^2}{\|\mathbf{Y}_m - \hat{\mathbf{Y}}_m\|_2^2} \right\} \right] \text{dB}, \quad (2.18)$$

with \log_{10} the base 10 logarithm. The SNR is infinity (∞) if the matrices \mathbf{Y} and $\hat{\mathbf{Y}}$ are identical. More specifically, in that case the reciprocal of the SNR, the *noise-to-signal-ratio* is zero, i.e. there is no noise.

2.5.3 Guidelines to obtain a proper data set for identification

Four important guidelines are outlined below:

1. **Excitation.** With SMI it is possible to identify the system χ at one time. This is also called the one-shot approach, i.e. excite the system χ by using *all* actuator inputs $\tilde{\mathbf{u}}(k)$ simultaneously, record *all* error sensor outputs $\mathbf{e}(k)$, and use the obtained input/output data set to estimate a model $\hat{\chi}$. This approach is attractive, since in one single experiment a compact model can be estimated for both the primary path \mathbf{P} and secondary path \mathbf{S} .
2. **Input signal.** The input signal used for identification should excite the system properly in the frequency band of interest. Thus it is important that the input signal has proper spectral content at all desired frequencies. For example, assuming linearity, an input signal $\tilde{\mathbf{u}}(k)$ that contains one single sinusoid of normalized frequency $\omega_1 T$, only gives back information about the system χ at frequency $\omega_1 T$, i.e. an amplitude and a phase (Ljung and Glad, 1994). In SMI, $\tilde{\mathbf{u}}(k)$ should excite continuously. In general terms, this means that all relevant dynamics of the system that need to be identified should be excited.
3. **Sample frequency.** In principle it is favorable to choose a high sample frequency for the real-time system to operate at. However, sampling at a high frequency puts a burden on the computational performance. Therefore, an appropriate choice needs to be made between a suitable sample frequency and the number of computations that can be executed by the finite precision arithmetic. On the other hand, there is a lower bound with regards to the sample frequency that deals with the frequency area of interest. For example, in our case the frequency area of interest is 0-1 kHz. According to Nyquist's sampling theorem (Enden, 1989), sampling the real-time control system at at least 2 kHz is then necessary. Thus, a model needs to be obtained at a sample frequency of 2 kHz to be appropriate in the frequency region of interest.
4. **Number of samples.** In theory it is true that the more samples that are taken, the better an accurate model can be properly identified. However, choosing too many samples may increase the time to identify a proper model due to lack of computational

power. To be precise, the determination of the matrices $\hat{\mathbf{A}}$ and $\hat{\mathbf{C}}$ of the model $\hat{\chi}$ is based on solving a QR-factorization and a singular value decomposition (SVD). The SVD especially requires a lot of computational power. Furthermore, to find the initial state $\mathbf{z}(1)$ and the matrices $\hat{\mathbf{B}}$ and $\hat{\mathbf{D}}$, a linear least squares problem needs to be solved, which comes with a significant computational load (Kailath et al., 2000). Therefore, the more samples that are taken, the slower the estimation procedure. Although there are ways to shorten the estimation time, e.g. by using the fast QR-factorization instead of a normal QR-factorization (Golub and Loan, 1996), a workable balance needs to be found between the available computational power, the order of the model that needs to be estimated and the time available for estimating the model.

The interested reader is referred to appendix C for a representative example of SMI performed on the 1DOF MDS system.

Chapter 3

Design of fixed gain and adaptive controllers for narrowband and broadband disturbance rejection

3.1 Introduction

This chapter describes how fixed gain controllers and adaptive controllers can be designed for the control scenarios depicted in figures 2.12 and 2.15.

The main objectives of this chapter are to present derivations of the:

1. **Wiener controller.** The Wiener controller is a fixed gain controller that is suitable for suppressing both narrowband and broadband disturbance outputs. The Wiener controller is mainly used in the simulation and real-time experiments for benchmark purposes, to get insight into the optimum performance that can be obtained (in general, but by the adaptive controller in particular).
2. **Filtered-x least mean square (FxLMS) algorithm.** The FxLMS algorithm is used in this thesis to update the coefficients of an adaptive controller. Like the Wiener controller, an adaptive controller that is driven by the FxLMS algorithm is suitable for suppressing both narrowband and broadband disturbance outputs. Generally, whereas the Wiener controller is a fixed gain controller that is computed off-line, the FxLMS algorithm is an algorithm that is able to find the coefficients of the Wiener controller on a sample-by-sample basis. In this chapter a convergence analysis of the FxLMS algorithm is presented also, which is useful for analyzing the convergence behavior of the adaptive controller coefficients. It is explained how the convergence time of the adaptive controller coefficients can be decreased by post-conditioning the adaptive controller, i.e. by including prior information of the secondary path \mathbf{S} in the adaptive controller. A contribution found in this chapter is the introduction of hybrid algorithms for SD-AVIC in which some of the models in the adaptive control scheme are modelled by a state space filter (SSF) and the adaptive controller is modelled by a finite impulse response filter (FIRF).

3. **Principle component least mean square (PCLMS) algorithm.** The PCLMS algorithm is also used to update the coefficients of an adaptive controller. An adaptive controller that is driven by the PCLMS algorithm is only suitable for suppressing a narrowband disturbance output that consists of a single sine. However, the algorithm has a clear benefit in comparison with the FxLMS algorithm in the sense that the convergence speed of the adaptive controller coefficients is generally higher. This is due to a decoupling of the adaptive controller with respect to the secondary path. The PCLMS algorithm is implemented and validated in the simulation and real-time experiments later on and the results are compared with those obtained with the FxLMS algorithm.

For a better understanding of the theory, illustrative examples are given throughout this chapter. Most of the discussion presented in this chapter assumes perfect plant knowledge. Therefore, the symbol $\widehat{(\cdot)}$ is barely used. Furthermore, this chapter mainly deals with MIMO systems in theory. To explain some of the mathematical issues, SISO systems and single input multiple output (SIMO) systems are used in the examples, though.

A significant number of text books and articles have been written on the topics of the Wiener controller, the FxLMS algorithm and the PCLMS algorithm. See for example (Elliott, 2000*b*; Kuo and Morgan, 1996; Fuller et al., 1996; Haykin, 1996; Sayed, 2003; Haykin and Widrow, 2003) for a discussion of the Wiener controller and the FxLMS algorithm. A description of the PCLMS algorithm can be found in e.g. (Cabell, 1998; Elliott, 2000*b*).

This chapter is organized as follows. Section 3.2 describes how the Wiener controller can be derived. In section 3.3 the FxLMS algorithm derivation is presented, followed by a convergence analysis. In section 3.4 it is demonstrated how the convergence time of the controller coefficients can be decreased by post-conditioning the adaptive controller. Finally, in section 3.5 the PCLMS algorithm derivation is treated.

3.2 Wiener controller

This section is organized as follows. In section 3.2.1 the control objective that is pursued by the Wiener controller is stated. Section 3.2.2 explains what filter structure is chosen for the Wiener controller. The filter structure is an essential part of the controller and its choice plays an important role in the derivation. In section 3.2.3 the actual derivation of the Wiener controller is presented for both the feedforward and the feedback control scenario. This derivation is performed in the time-domain. However, a small section is dedicated to deriving the Wiener controller in the frequency domain for narrowband disturbances only.

3.2.1 Control objective

In figure 3.1 a block diagram of a MIMO feedforward SD-AVIC system is shown (see also figure 2.13). Ideally, the controller \mathbf{W} should be such that the transfer path $\mathbf{S}\mathbf{W}$ equals the transfer path \mathbf{P} . This implies that the controller \mathbf{W} can be described by the equation:

$$\mathbf{W} = -\mathbf{S}^{-1}\mathbf{P}, \quad (3.1)$$

assuming that \mathbf{S} can be inverted. In most control systems, the controller \mathbf{W} cannot be computed like in equation (3.1), since the secondary path \mathbf{S} has zeros that are outside the unit circle. The inverted system would thus be unstable. The main question then is how to

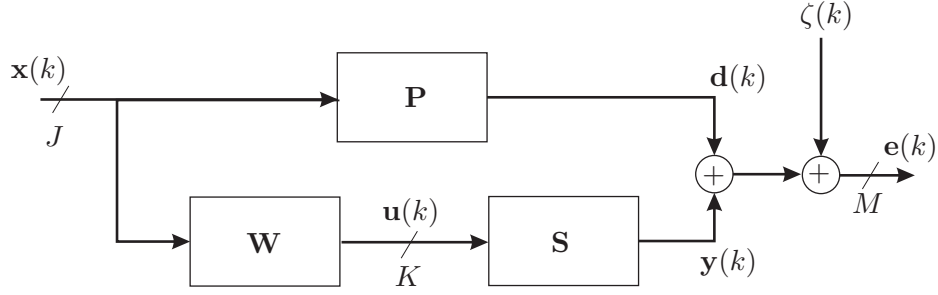


Figure 3.1: Block diagram of a MIMO feedforward SD-AVIC system with a fixed gain controller \mathbf{W} .

compute the controller \mathbf{W} . The answer is that the controller is not computed directly as in equation (3.1), but that the best suitable controller is sought that satisfies a certain criterion. Specifically, the computation of the controller is done by minimizing a cost function. Good performance is generally defined in terms of minimizing a mean square, or H_2 , type of cost function applied to the error sensor outputs (Elliott, 2000b). This is understandable as in physical terms this means minimizing the power of the error sensor outputs. This results in the following controller design problem.

Problem 3.2.1 (Fixed gain controller). *Given the asymptotically stable primary path \mathbf{P} and the asymptotically stable secondary path \mathbf{S} , find an asymptotically stable controller \mathbf{W} , so that:*

$$\mathbf{W} = \arg \min_{\mathbf{W}} J,$$

where the cost function is defined as:

$$J = E[\mathbf{e}^T(k)\mathbf{e}(k)], \quad (3.2)$$

with E the statistical expectation operator (Leon-Garcia, 1994).

The solution to this problem is given by the Wiener controller.

3.2.2 Filter structure

In this chapter the FIRF structure for the controller is used (see section A.3 in appendix A). The reason for this is that in combination with the applied H_2 cost function given in equation (3.2), the optimization procedure for finding the controller coefficients is *linear-in-the-coefficients*. This is explained in more detail later. Thus, the controller \mathbf{W} is defined as:

$$\mathbf{W} = \begin{bmatrix} \sum_{l=1}^L w_{l-1}^{11} q^{-l+1} & \cdots & \sum_{l=1}^L w_{l-1}^{1J} q^{-l+1} \\ \cdots & \ddots & \vdots \\ \sum_{l=1}^L w_{l-1}^{K1} q^{-l+1} & \cdots & \sum_{l=1}^L w_{l-1}^{KJ} q^{-l+1} \end{bmatrix}. \quad (3.3)$$

with w_l^{ni} , ($l = 0, \dots, L - 1$), ($n = 1, \dots, K$), ($i = 1, \dots, J$), the coefficients of the controller \mathbf{W} . q^{-1} is the backward shift operator. The L controller coefficients of each channel¹ in equation (3.3) are stacked in a column vector $\mathbf{w}_{ni} \in \mathbb{R}^L$:

$$\mathbf{w}_{ni} = [w_0^{ni} \ w_1^{ni} \ \dots \ w_{L-1}^{ni}]^T. \quad (3.4)$$

Subsequently, the controller coefficients from all inputs to a particular output are stacked in a new column vector $\mathbf{w}_n \in \mathbb{R}^{JL}$:

$$\mathbf{w}_n = [\mathbf{w}_{n1}^T \ \mathbf{w}_{n2}^T \ \dots \ \mathbf{w}_{nJ}^T]^T. \quad (3.5)$$

The latter vector collects all J vectors defined in equation (3.4). Finally, all vectors \mathbf{w}_n are stacked in a new column vector $\mathbf{w} \in \mathbb{R}^{JKL}$:

$$\mathbf{w} = [\mathbf{w}_1^T \ \mathbf{w}_2^T \ \dots \ \mathbf{w}_K^T]^T. \quad (3.6)$$

The latter vector collects all vectors defined in equation (3.5) and, as such, contains all controller coefficients of the controller \mathbf{W} defined by equation (3.3).

3.2.3 Feedforward control scenario

The derivation of the Wiener controller now follows from the cost function that is given in equation (3.2). The error sensor outputs $\mathbf{e}(k)$ can be described by² (see also equation (2.12)):

$$\begin{aligned} \mathbf{e}(k) &= \mathbf{d}(k) + \mathbf{y}(k) + \zeta(k) \\ &= \mathbf{d}(k) + \mathbf{S}\mathbf{u}(k) + \zeta(k). \end{aligned} \quad (3.9)$$

A crucial step in the derivation of the Wiener controller is that by using a Kronecker tensor product (indicated by ' \otimes ' (Brewer, 1978)), $\mathbf{y}(k) = \mathbf{S}\mathbf{u}(k)$ in equation (3.9) can be written as:

$$\mathbf{y}(k) = \mathbf{R}_x(k)\mathbf{w}, \quad (3.10)$$

¹The term *channel* denotes a controller from a particular input to a particular output such as e.g. the channel from the first input to the first output:

$$\sum_{l=1}^L w_{l-1}^{11} q^{-l+1},$$

in equation (3.3).

²For future reference, note that it is also possible to write:

$$\mathbf{e}(k) = \mathbf{d}(k) + \mathbf{S}[\mathbf{X}^T(k)\mathbf{w}] + \zeta(k), \quad (3.7)$$

where the part between square brackets $[\cdot]$ represents a *signal* instead of a system. The brackets are used here for reasons of clarity only and have no other function. The regression *matrix* $\mathbf{X}(k) \in \mathbb{R}^{JKL \times K}$ is defined as:

$$\mathbf{X}(k) = \begin{bmatrix} \bar{\mathbf{x}}(k) & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{x}}(k) & \dots & \vdots \\ \vdots & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \bar{\mathbf{x}}(k) \end{bmatrix}. \quad (3.8)$$

where $\mathbf{R}_x(k) \in \mathbb{R}^{M \times JKL}$ is defined as:

$$\mathbf{R}_x(k) = \mathbf{S} \otimes \bar{\mathbf{x}}^T(k). \quad (3.11)$$

$\bar{\mathbf{x}}(k) \in \mathbb{R}^{JL}$ is the regression column vector that is defined as³:

$$\bar{\mathbf{x}}(k) = [\bar{\mathbf{x}}_1^T(k) \ \bar{\mathbf{x}}_2^T(k) \ \cdots \ \bar{\mathbf{x}}_J^T(k)]^T,$$

with:

$$\bar{\mathbf{x}}_i(k) = [x_i(k) \ x_i(k-1) \ \cdots \ x_i(k-L+1)]^T, \quad (3.12)$$

the regression column vector of the i^{th} reference input for $(i = 1, \dots, J)$. For example, if $L = J = 2$ it holds that $\bar{\mathbf{x}}(k) = [\bar{\mathbf{x}}_1(k) \ \bar{\mathbf{x}}_2(k)]^T = [x_1(k) \ x_1(k-1) \ x_2(k) \ x_2(k-1)]^T \in \mathbb{R}^4$. That $\mathbf{y}(k)$ can be described by equation (3.10) is not evident. To illustrate this step, an example is used.

Example 3.2.1 (Kronecker product). *Suppose the controller \mathbf{W} is SISO with two parameters and the secondary path \mathbf{S} has one input and two outputs as well as a FIRF structure with two parameters:*

$$\mathbf{W} = w_1 + w_2q^{-1}, \quad \mathbf{S} = \begin{bmatrix} s_1^1 + s_2^1q^{-1} \\ s_1^2 + s_2^2q^{-1} \end{bmatrix}.$$

The regression vector $\bar{\mathbf{x}}(k)$ is defined as:

$$\bar{\mathbf{x}}(k) = [x(k) \ x(k-1)]^T.$$

It will now be demonstrated that equation (3.10) holds. Since $\mathbf{y}(k) = \mathbf{S}u(k)$, it is true that:

$$\mathbf{S}u(k) = (\mathbf{S} \otimes \bar{\mathbf{x}}^T(k)) \mathbf{w}. \quad (3.13)$$

$\mathbf{S}u(k)$ is given by:

$$\mathbf{S}u(k) = \begin{bmatrix} s_1^1 u(k) + s_2^1 u(k-1) \\ s_1^2 u(k) + s_2^2 u(k-1) \end{bmatrix}. \quad (3.14)$$

The output of the controller (i.e. $u(k)$) and the previous output of the controller (i.e. $u(k-1)$) are given by:

$$u(k) = w_1 x(k) + w_2 x(k-1), \quad u(k-1) = w_1 x(k-1) + w_2 x(k-2). \quad (3.15)$$

Substituting the results from equation (3.15) in equation (3.14), gives:

$$\mathbf{S}u(k) = \begin{bmatrix} s_1^1 (w_1 x(k) + w_2 x(k-1)) + s_2^1 (w_1 x(k-1) + w_2 x(k-2)) \\ s_1^2 (w_1 x(k) + w_2 x(k-1)) + s_2^2 (w_1 x(k-1) + w_2 x(k-2)) \end{bmatrix}.$$

³Note the horizontal line above $\bar{\mathbf{x}}$ to differentiate the regression vector from the reference inputs $\mathbf{x}(k)$ depicted in figure 3.1.

The latter equation can be worked out further:

$$\begin{aligned}
 \mathbf{S}u(k) &= \begin{bmatrix} s_1^1 w_1 x(k) + s_1^1 w_2 x(k-1) + s_2^1 w_1 x(k-1) + s_2^1 w_2 x(k-2) \\ s_1^2 w_1 x(k) + s_1^2 w_2 x(k-1) + s_2^2 w_1 x(k-1) + s_2^2 w_2 x(k-2) \end{bmatrix} \\
 &= \begin{bmatrix} (s_1^1 x(k) + s_2^1 x(k-1)) w_1 + (s_1^1 x(k-1) + s_2^1 x(k-2)) w_2 \\ (s_1^2 x(k) + s_2^2 x(k-1)) w_1 + (s_1^2 x(k-1) + s_2^2 x(k-2)) w_2 \end{bmatrix} \\
 &= \begin{bmatrix} s_1^1 x(k) + s_2^1 x(k-1) & s_1^1 x(k-1) + s_2^1 x(k-2) \\ s_1^2 x(k) + s_2^2 x(k-1) & s_1^2 x(k-1) + s_2^2 x(k-2) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \\
 &= \begin{bmatrix} (s_1^1 + s_2^1 q^{-1}) x(k) & (s_1^1 + s_2^1 q^{-1}) x(k-1) \\ (s_1^2 + s_2^2 q^{-1}) x(k) & (s_1^2 + s_2^2 q^{-1}) x(k-1) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \\
 &= \left(\begin{bmatrix} s_1^1 + s_2^1 q^{-1} \\ s_1^2 + s_2^2 q^{-1} \end{bmatrix} \otimes [x(k) \ x(k-1)] \right) \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}.
 \end{aligned} \tag{3.16}$$

This formula shows the validity of equation (3.10). Note that the following five steps are applied above: omission of the brackets, collecting terms corresponding to a particular controller coefficient, putting the controller coefficients in a vector, putting the equation in system form by using the unit delay operator q^{-1} and using the Kronecker product to rewrite the equation.

Substituting the first expression from equation (3.9) in equation (3.2) and using equation (3.10), gives:

$$\begin{aligned}
 J &= \mathbb{E} \left[\{ \mathbf{d}(k) + \zeta(k) + \mathbf{R}_x(k) \mathbf{w} \}^T \{ \mathbf{d}(k) + \zeta(k) + \mathbf{R}_x(k) \mathbf{w} \} \right] \\
 &= \mathbb{E} \left[\{ \mathbf{d}^T(k) \mathbf{d}(k) \} + \{ \zeta^T(k) \zeta(k) \} + \{ \mathbf{w}^T \mathbf{R}_x^T(k) \mathbf{R}_x(k) \mathbf{w} \} + \{ 2 \mathbf{d}^T(k) \mathbf{R}_x(k) \mathbf{w} \} + \right. \\
 &\quad \left. + \{ 2 \zeta^T(k) \mathbf{R}_x(k) \mathbf{w} \} + \{ 2 \zeta^T(k) \mathbf{d}(k) \} \right] \\
 &= \mathbb{E} \{ \mathbf{d}^T(k) \mathbf{d}(k) \} + \mathbb{E} \{ \zeta^T(k) \zeta(k) \} + \mathbf{w}^T \mathbb{E} \{ \mathbf{R}_x^T(k) \mathbf{R}_x(k) \} \mathbf{w} + 2 \mathbb{E} \{ \mathbf{d}^T(k) \mathbf{R}_x(k) \} \mathbf{w}.
 \end{aligned} \tag{3.17}$$

In the transition from the second to the third expression, use is made of the fact that $\mathbb{E} \{ 2 \zeta^T(k) \mathbf{R}_x(k) \mathbf{w} \} = \mathbb{E} \{ 2 \zeta^T(k) \mathbf{d}(k) \} = 0$, i.e. the expected value between the filtered reference inputs and the measurement noise and the expected value between the measurement noise and the disturbance outputs is equal to zero. This is true since in section 2.4 it is assumed that the measurement noise $\zeta(k)$ is uncorrelated to the reference inputs $\mathbf{x}(k)$ and the disturbance outputs $\mathbf{d}(k)$. The optimum controller coefficients \mathbf{w}_{opt} of the Wiener controller can now be computed by setting the gradient of the cost function J in the last expression in equation (3.17) to zero with respect to the controller coefficients \mathbf{w} :

$$\frac{\partial J}{\partial \mathbf{w}} = 0. \tag{3.18}$$

This results in:

$$2 \mathbb{E} \{ \mathbf{R}_x^T(k) \mathbf{R}_x(k) \} \mathbf{w} = -2 \mathbb{E} \{ \mathbf{R}_x^T(k) \mathbf{d}(k) \}, \tag{3.19}$$

which gives the optimum controller coefficients:

$$\mathbf{w}_{opt} = - [\mathbb{E} \{ \mathbf{R}_x^T(k) \mathbf{R}_x(k) \}]^{-1} \mathbb{E} \{ \mathbf{R}_x^T(k) \mathbf{d}(k) \}, \tag{3.20}$$

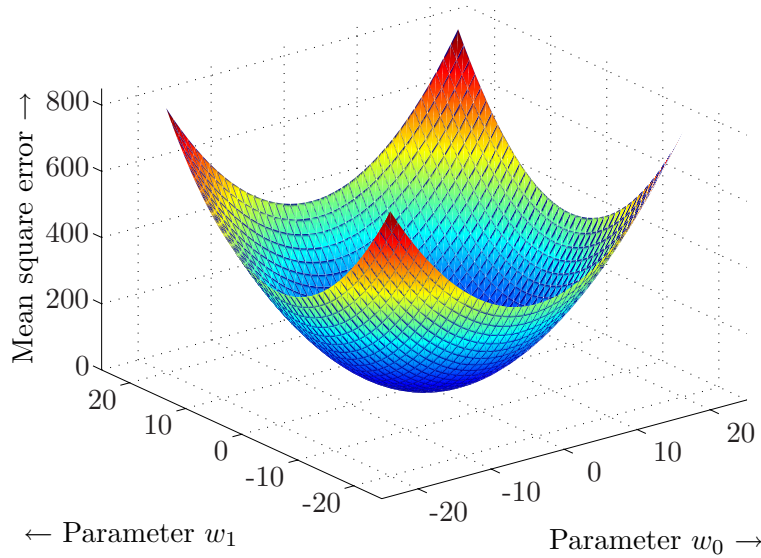


Figure 3.2: Cost function J that has a global minimum only which is present in $(w_0, w_1) = (0, 0)$.

assuming that $(\det [\mathbf{E} \{ \mathbf{R}_x^T(k) \mathbf{R}_x(k) \}] \neq 0)$. The latter is the case if the reference inputs $\mathbf{x}(k)$ are persistently exciting (Regalia, 1995). From equation (3.20) it is clear that the controller coefficients \mathbf{w}_{opt} do not depend on the measurement noise $\zeta(k)$. That is a well-known result in feedforward control (Elliott, 2000b) and straightforward to understand: it obviously is not possible to suppress measurement noise $\zeta(k)$ with a feedforward controller that has no information about the measurement noise. Substituting the controller coefficients of equation (3.20) in equation (3.3) gives the Wiener controller \mathbf{W} .

Remark: It is also possible to derive the Wiener controller as a SSF, in terms of identified state space models of the primary path \mathbf{P} and the secondary path \mathbf{S} . For an example of this, see for instance (Fraanje, 2004; Elliott, 2000b). The derivation itself is beyond the scope of this thesis, but is summarized in appendix B.

Finding the controller coefficients is a *linear-in-the-coefficients* problem. To explain this, see the last expression in equation (3.17). This expression is quadratic in the controller coefficients \mathbf{w} and the gradient - which is the derivative of the cost function J with respect to \mathbf{w} - in equation (3.19) is therefore *linear-in-the-coefficients* \mathbf{w} . This means that a unique solution exists for the controller parameters \mathbf{w} . In figure 3.2 an example of such a cost function J is given in case the Wiener controller consists of two coefficients. The cost function is quadratic and has a global minimum only, which is situated in $(w_0, w_1) = (0, 0)$. More importantly, there are no local minimums.

The Wiener controller in case the reference input is tonal

If the reference inputs $\mathbf{x}(k)$ only contain a single frequency, the optimum controller coefficients can be computed more easily in the *frequency domain* as follows. The response of the

secondary path \mathbf{S} for a single frequency ω_1 rad/s is given by:

$$\mathbf{S}(e^{j\omega_1 T}) \in \mathbb{C}^{M \times K}.$$

The response of the primary path \mathbf{P} for a single frequency ω_1 rad/s is given by:

$$\mathbf{P}(e^{j\omega_1 T}) \in \mathbb{C}^{M \times J}.$$

The goal of the adaptive control scheme depicted in figure 3.1 is to find a controller $\mathbf{W}(e^{j\omega_1 T}) \in \mathbb{C}^{K \times J}$ so that:

$$\mathbf{P}(e^{j\omega_1 T}) + \mathbf{S}(e^{j\omega_1 T}) \mathbf{W}(e^{j\omega_1 T}) = \mathbf{0}_{M \times J}. \quad (3.21)$$

This means that:

$$\mathbf{S}(e^{j\omega_1 T}) \mathbf{W}(e^{j\omega_1 T}) = -\mathbf{P}(e^{j\omega_1 T}).$$

Thus, the optimum controller $\mathbf{W}_{opt}(e^{j\omega_1 T})$ is given by:

$$\mathbf{W}_{opt}(e^{j\omega_1 T}) = -\mathbf{S}^\dagger(e^{j\omega_1 T}) \mathbf{P}(e^{j\omega_1 T}), \quad (3.22)$$

with $\mathbf{S}^\dagger(e^{j\omega_1 T})$ the pseudo-inverse⁴ of $\mathbf{S}(e^{j\omega_1 T})$ (Golub and Loan, 1996). Equation (3.22) gives the optimum controller $\mathbf{W}_{opt}(e^{j\omega_1 T})$ in the frequency domain as $K \times J$ complex coefficients. Every complex coefficient can be transformed to a two-coefficient FIRF to obtain the optimum controller \mathbf{W}_{opt} in the time domain. How a complex coefficient can be transformed to a two-coefficient FIRF, is explained in appendix D.

3.2.4 Improving the robustness of the controller with regularization

In the case of many practical problems, it is useful to include a regularization term in the cost function, to increase the robustness of the controller. For example:

- the *numerical* robustness needs to be increased if the inverse in equation (3.20) cannot be computed due to ill-conditioning of the matrix $\mathbf{E}\{\mathbf{R}_x^T(k)\mathbf{R}_x(k)\}$.
- the robustness needs to be increased if the voltages that need to be applied to the actuators cannot be handled by the digital-to-analogue converters of the real-time control system.
- in the feedback control scenario regularization is also used to increase the robustness of the controller to model uncertainties in the secondary path \mathbf{S} . Section 3.2.5 elaborates on that.

The regularized cost function is given by (Elliott, 2000b):

$$\check{J} = \mathbf{E}[\mathbf{e}^T(k)\mathbf{e}(k)] + \rho \mathbf{w}^T \mathbf{w}, \quad (3.23)$$

with $\rho \in \mathbb{R}^+$ a small positive constant. In addition to the expected value of the square error sensor outputs $\mathbf{e}(k)$, this cost function also weighs the controller parameters \mathbf{w} . Compare

⁴For example, if $M \geq K$:

$$\mathbf{s}^\dagger(e^{j\omega_1 T}) = \left\{ \mathbf{s}(e^{j\omega_1 T})^H \mathbf{s}(e^{j\omega_1 T}) \right\}^{-1} \mathbf{s}^H(e^{j\omega_1 T}).$$

this cost function with the cost function described by equation 3.2⁵. This can be interpreted as that the gain of the controller is reduced over the entire frequency band (Elliott, 2000b). One can see that due to the regularization in equation (3.23), an extra term is added to the cost function in equation (3.17). Thus, the regularized cost function is given by:

$$\check{J} = E \{ \mathbf{d}^T(k) \mathbf{d}(k) \} + E \{ \zeta^T(k) \zeta(k) \} + \mathbf{w}^T E \{ \mathbf{R}_x^T(k) \mathbf{R}_x(k) \} \mathbf{w} + 2E \{ \mathbf{d}^T(k) \mathbf{R}_x(k) \} \mathbf{w} + \rho \mathbf{w}^T \mathbf{w}.$$

Collecting terms gives:

$$\begin{aligned} \check{J} &= E \{ \mathbf{d}^T(k) \mathbf{d}(k) \} + E \{ \zeta^T(k) \zeta(k) \} + \mathbf{w}^T (E \{ \mathbf{R}_x^T(k) \mathbf{R}_x(k) \} + \rho \mathbf{I}_{JKL}) \mathbf{w} \\ &+ 2E \{ \mathbf{d}^T(k) \mathbf{R}_x(k) \} \mathbf{w}. \end{aligned} \quad (3.24)$$

The regularized controller coefficients $\check{\mathbf{w}}_{opt}$ can now be computed by setting the gradient of the cost function \check{J} to zero with respect to the controller coefficients \mathbf{w} :

$$\frac{\partial \check{J}}{\partial \mathbf{w}} = 0,$$

which results in:

$$2 (E \{ \mathbf{R}_x^T(k) \mathbf{R}_x(k) \} + \rho \mathbf{I}_{JKL}) \mathbf{w} = -2E \{ \mathbf{R}_x^T(k) \mathbf{d}(k) \}. \quad (3.25)$$

This gives:

$$\check{\mathbf{w}}_{opt} = - [E \{ \mathbf{R}_x^T(k) \mathbf{R}_x(k) \} + \rho \mathbf{I}_{JKL}]^{-1} E \{ \mathbf{R}_x^T(k) \mathbf{d}(k) \}. \quad (3.26)$$

Due to the regularization, an extra term $\rho \mathbf{I}_{JKL}$ is added to the matrix $E \{ \mathbf{R}_x^T(k) \mathbf{R}_x(k) \}$. Compare this equation with equation (3.20). If this matrix is ill-conditioned and cannot be inverted, a regularization term $\rho > 0$ will have a positive effect on the inversion. If there is no regularization (i.e. $\rho = 0$), the solution of equation (3.26) coincides with the solution of equation (3.20). This form of regularization is a simple but effective method to increase the robustness, but it will go at the expense of finding the optimum controller parameters \mathbf{w}_{opt} . Only a *regularized* solution $\check{\mathbf{w}}_{opt}$ is found.

Remark: *It is also possible to derive the regularized Wiener controller as a SSF in terms of identified state space models of the primary path \mathbf{P} and the secondary path \mathbf{S} . A summary of this derivation is given in Appendix B.*

3.2.5 Feedback control scenario

Although the derivation of the Wiener controller for the feedback control scenario is almost equal to the derivation for the feedforward control scenario, there are some differences. For example, as will be shown here, the controller coefficients are dependent on the measurement noise $\zeta(k)$. Furthermore, model uncertainties in the secondary path \mathbf{S} play a role. In this section those aspects are analyzed. First, the scenario without model uncertainty is treated and the sole focus is on how measurement noise plays a role. Later, the effect of model uncertainty is investigated.

No model uncertainty. Figure 3.3 shows a block diagram of a MIMO feedback SD-AVIC system. This figure is the same as figure 2.16, with the exception that the dashed arrow

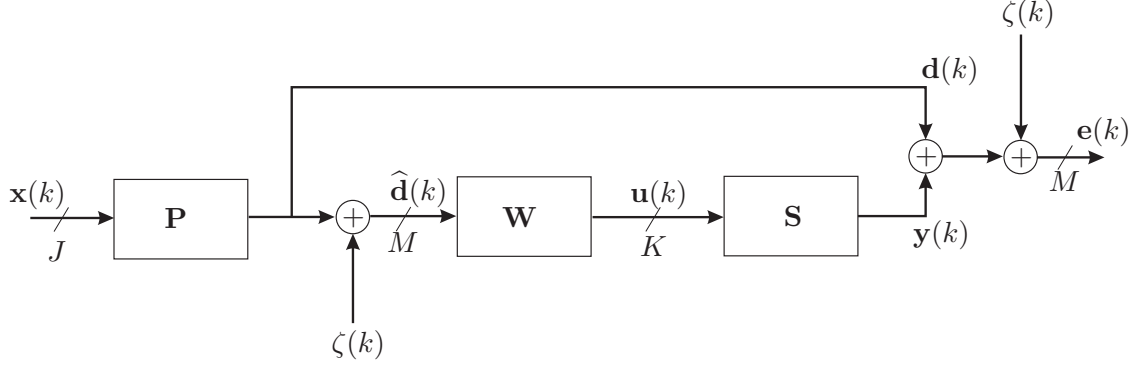


Figure 3.3: Block diagram of a IMC based MIMO feedback SD-AVIC system with a fixed gain controller \mathbf{W} , assuming perfect plant knowledge of the secondary path \mathbf{S} .

through \mathbf{W} is now omitted since it is clear that it concerns a fixed gain controller. From the figure it is clear that the error sensor outputs $\mathbf{e}(k)$ can be written as:

$$\begin{aligned} \mathbf{e}(k) &= \mathbf{d}(k) + \zeta(k) + \mathbf{y}(k) \\ &= \mathbf{d}(k) + \zeta(k) + \mathbf{R}_d(k)\mathbf{w}, \end{aligned} \quad (3.27)$$

with:

$$\mathbf{R}_d(k) = \mathbf{S} \otimes \overline{\mathbf{d}}^T(k).$$

The regression column vector $\overline{\mathbf{d}}(k) \in \mathbb{R}^{ML}$ is given by:

$$\overline{\mathbf{d}}(k) = \begin{bmatrix} \overline{\mathbf{d}}_1^T(k) & \overline{\mathbf{d}}_2^T(k) & \cdots & \overline{\mathbf{d}}_M^T(k) \end{bmatrix}^T, \quad (3.28)$$

with

$$\overline{\mathbf{d}}_i(k) = \begin{bmatrix} \widehat{d}_i(k-1) & \widehat{d}_i(k-2) & \cdots & \widehat{d}_i(k-L+1) \end{bmatrix}^T,$$

the regression vector of the i^{th} estimated disturbance output for $(i = 1, \dots, M)$. Just as in the feedforward control scenario, in the feedback control scenario the cost function described by equation (3.2) needs to be minimized. Equation (3.27) can be substituted in equation (3.2), which gives:

$$\begin{aligned} J &= \text{E} \left[\{ \mathbf{d}(k) + \zeta(k) + \mathbf{R}_d(k)\mathbf{w} \}^T \{ \mathbf{d}(k) + \zeta(k) + \mathbf{R}_d(k)\mathbf{w} \} \right] \\ &= \text{E} \left[\{ \widehat{\mathbf{d}}(k) + \mathbf{R}_d(k)\mathbf{w} \}^T \{ \widehat{\mathbf{d}}(k) + \mathbf{R}_d(k)\mathbf{w} \} \right] \\ &= \text{E} \left[\{ \widehat{\mathbf{d}}^T(k)\widehat{\mathbf{d}}(k) \} + \{ \mathbf{w}^T \mathbf{R}_d^T(k) \mathbf{R}_d(k) \mathbf{w} \} + \{ 2\widehat{\mathbf{d}}^T(k) \mathbf{R}_d(k) \mathbf{w} \} \right] \\ &= \text{E} \left\{ \widehat{\mathbf{d}}^T(k)\widehat{\mathbf{d}}(k) \right\} + \mathbf{w}^T \text{E} \left\{ \mathbf{R}_d^T(k) \mathbf{R}_d(k) \right\} \mathbf{w} + 2\text{E} \left\{ \widehat{\mathbf{d}}^T(k) \mathbf{R}_d(k) \right\} \mathbf{w}. \end{aligned} \quad (3.29)$$

⁵In this thesis, regularized cost functions are indicated by a breve above J .

The optimum feedback controller coefficients \mathbf{w}_{opt} (the same symbol for the controller coefficients is used since basically a feedforward control problem is solved) can be computed by setting the gradient of the cost function J in the last expression to zero with respect to the controller coefficients \mathbf{w} (see equation (3.18)). This results in:

$$2\mathbb{E} \{ \mathbf{R}_d^T(k) \mathbf{R}_d(k) \} \mathbf{w} = -2\mathbb{E} \{ \mathbf{R}_d^T(k) \hat{\mathbf{d}}(k) \}. \quad (3.30)$$

This gives:

$$\mathbf{w}_{opt} = - [\mathbb{E} \{ \mathbf{R}_d^T(k) \mathbf{R}_d(k) \}]^{-1} \mathbb{E} \{ \mathbf{R}_d^T(k) \hat{\mathbf{d}}(k) \}. \quad (3.31)$$

The solution of this equation is similar to the solution of equation (3.20) with \mathbf{R}_d instead of \mathbf{R}_x and $\hat{\mathbf{d}}(k)$ instead of $\mathbf{d}(k)$. However, the controller coefficients of the feedforward controller in equation (3.20) do not depend on the measurement noise $\zeta(k)$. The controller coefficients of the feedback controller do depend on the measurement noise $\zeta(k)$, i.e. the measurement noise is embedded in the matrix $\mathbf{R}_d(k)$ and the vector $\hat{\mathbf{d}}(k)$. If the coefficients \mathbf{w}_{opt} for the controller are substituted in equation (3.3), the Wiener controller \mathbf{W} is found for the feedback control scenario. The complete feedback controller \mathbf{K} is given by the following Youla parametrization (Youla et al., 1976a,b):

$$\mathbf{K} = (\mathbf{I}_K + \mathbf{W}\mathbf{S})^{-1} \mathbf{W}. \quad (3.32)$$

For this, see also equation (2.14) and figure 2.15. The controller (internally) stabilizes the closed loop, since perfect plant knowledge of the secondary path \mathbf{S} is assumed, see e.g. (Elliott, 2000b; Fraanje, 2004). Note: if the secondary path \mathbf{S} can be stably inverted, the controller $\mathbf{W} = -\mathbf{S}^{-1}$ yields an infinite gain over the entire frequency band and therefore optimum performance is guaranteed over the entire frequency band. That is what ideally would be the case, but cannot be accomplished in practice since the secondary path \mathbf{S} cannot be stably inverted. Furthermore, due to model uncertainties in the secondary path, model inversion is not an option, since the feedback controller would never yield a stable closed loop.

Dealing with model uncertainty. Consider the feedback problem depicted in figure 3.4, which is a repeat of figure 2.15. Assume that the secondary path model $\hat{\mathbf{S}}$ is not identical to the secondary path \mathbf{S} , but that there are model uncertainties $\Delta\mathbf{S}$:

$$\mathbf{S} = \hat{\mathbf{S}} + \Delta\mathbf{S}.$$

The feedback controller \mathbf{K} again is given by equation (3.32) (with $\hat{\mathbf{S}}$ substituted for \mathbf{S}). However, now the controller \mathbf{K} does not necessarily internally stabilize the closed loop. The presence of the uncertainty $\Delta\mathbf{S}$ complicates the feedback controller design. In order to see this, rewrite the feedback control solution in figure 3.4 into an equivalent feedforward one. This gives rise to an additional loop around the controller due to the uncertainty $\Delta\mathbf{S}$, which is depicted in figure 3.5. It can be deduced (even intuitively) that the gain of the controller \mathbf{W} has to be reduced at the locations in the frequency domain where the uncertainty $\Delta\mathbf{S}$ is large. To accomplish that, in this thesis the cost function given in equation (3.23) is solved with respect to \mathbf{w} . This approach is proposed in (Elliott, 2000b, Chapter 6), to which one is referred for a more in-depth discussion on this topic. Using this approach gives the following solution for the controller parameters $\check{\mathbf{w}}_{opt}$:

$$\check{\mathbf{w}}_{opt} = - [\mathbb{E} \{ \mathbf{R}_d^T(k) \mathbf{R}_d(k) \} + \rho \mathbf{I}_{MKL}]^{-1} \mathbb{E} \{ \mathbf{R}_d^T(k) \hat{\mathbf{d}}(k) \}. \quad (3.33)$$

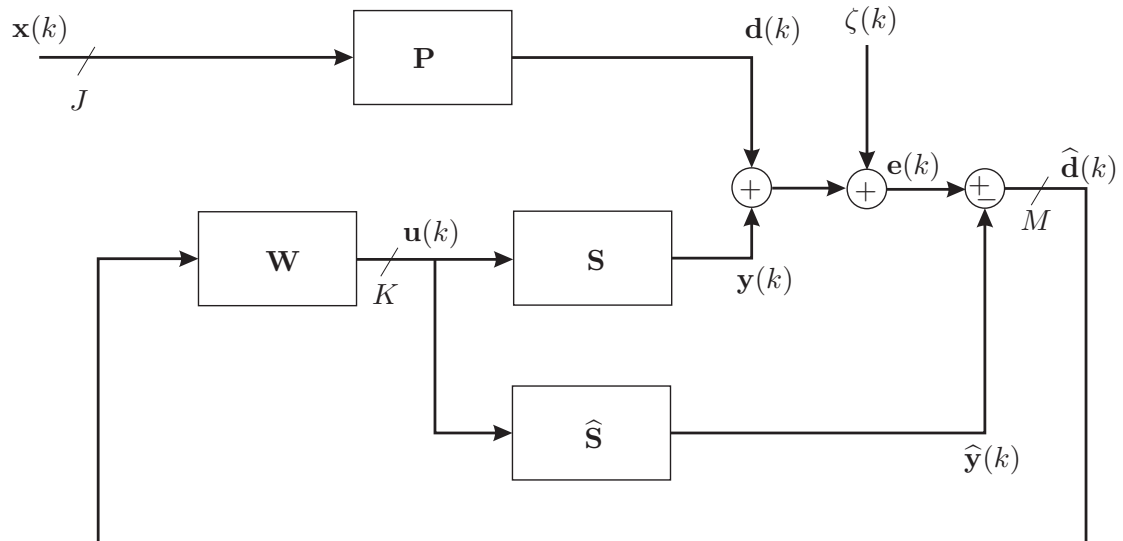


Figure 3.4: Block diagram of a MIMO feedback SD-AVIC system with a fixed gain controller W .

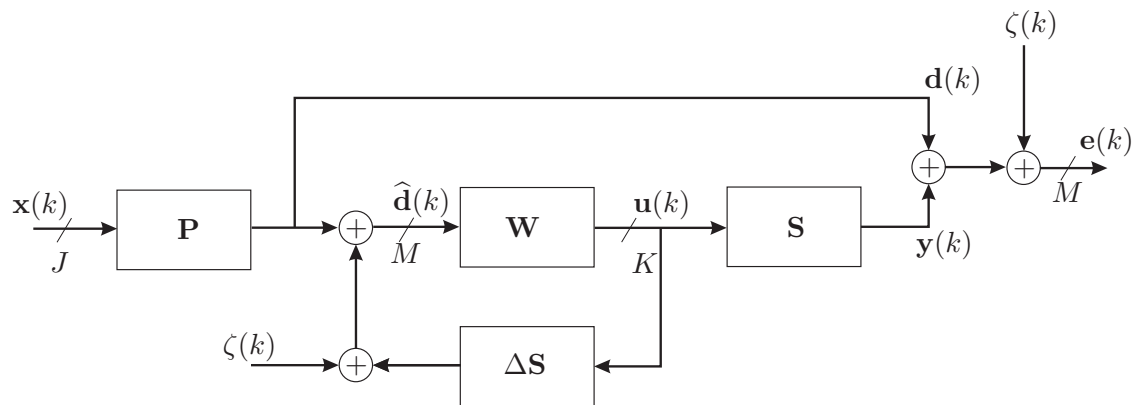


Figure 3.5: Block diagram of a MIMO feedback SD-AVIC system with a fixed gain controller W and assuming model uncertainty ΔS in the secondary path model.

Comparing this equation to equation (3.31) shows that an extra term $\rho \mathbf{I}_{MKL}$ is added. See also page 41 on which this is explained in more detail for the feedforward control scenario.

Other cost functions are available though for accomplishing robust stability. However, looking ahead, it can be said that the cost function given in equation (3.23) is suitable for the application described in this thesis and other cost functions are not treated. See e.g. (Elliott, 2000*b*; Fraanje, 2004; Morari and Zafiriou, 1989) for a description of other cost functions.

Up to this point feedforward and feedback fixed gain controllers have been discussed that are not able to respond to changes in e.g. plants and signals. In the next section the FxLMS algorithm is described that is able to drive the coefficients of an adaptive controller (see also section 2.3).

3.3 Filtered-x least mean square algorithm

For performance purposes, it would be beneficial if the controller had the ability to adapt to small changes. One could argue that it might be a solution to recompute the Wiener controller on-line, if a change would occur. However, this is difficult to do, since:

1. a change needs to be detected, which requires an extra mechanism that needs to be incorporated in the control scheme. Furthermore, it can be difficult to determine exactly when a change has taken place (Gustafsson, 2000; Basseville and Nikiforov, 1993).
2. the Wiener controller has to be estimated from the time histories of the data e.g. in the feedforward control scenario, this means based on the auto-correlation matrix of the filtered reference inputs and the cross-correlation vector between the filtered reference inputs and the disturbance outputs (see equation (3.20)). For accurate computation - in real-time problems - this may require a significant amount of data samples. The main problem with that is, first of all, that these data samples need to be stored, which requires memory. Second, in order to be able to collect the data samples the real-time control system cannot be in operation but has to be shut down.
3. e.g. equations (3.26) and (3.33) show that a matrix inversion needs to be done to compute the controller coefficients. Although there are ways to do the matrix inversion efficiently, the computational load is still proportional to the square of the number of coefficients in the controller (Elliott, 2000*b*; Kailath and Sayed, 1999). In the case of large scale problems that deal with a controller that has many coefficients, a fast real-time computer is necessary to compute the inverse.

These points imply that if the controller must be able to respond to small changes, another control strategy is desirable. An approach that has proven its usefulness in practice is to make the controller coefficients adaptive. Instead of computing the controller coefficients in advance, they are adapted on-line every time a new data sample comes in. This is what is called an adaptive control strategy instead of a fixed gain control strategy. An algorithm that is widely used for adapting the coefficients of the adaptive controller is the FxLMS algorithm, which is explained in this section⁶.

⁶In this section the FxLMS algorithm is only explained for the feedforward control scenario. Using the IMC-principle, the FxLMS algorithm for the feedback control scenario can easily be derived from the feedforward control scenario (with and without model uncertainty in the secondary path)

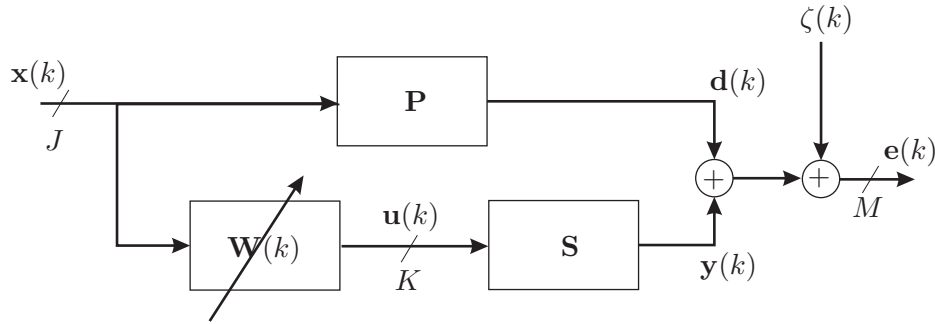


Figure 3.6: Block diagram of a MIMO feedforward SD-AVIC system with an adaptive controller $\mathbf{W}(k)$.

This section is organized as follows. Section 3.3.1 gives the derivation of the FxLMS algorithm. In section 3.3.2 a convergence analysis of the FxLMS algorithm is presented. In section 3.3.3 a new hybrid algorithm is introduced. Finally, section 3.3.4 explains how the robustness of the adaptive controller can be improved with regularization.

3.3.1 Derivation of the algorithm

Figure 3.6 shows a block diagram of a MIMO adaptive feedforward SD-AVIC system. In the adaptive control scheme, instead of minimizing the expected value of the square of the error sensor outputs as done in equation (3.2), the *instantaneous* value of the squared error sensor outputs is used. Thus, the time-varying cost function is defined as:

$$J(k) = \mathbf{e}^T(k)\mathbf{e}(k). \quad (3.34)$$

Also, in the adaptive control scheme a FIRF structure for the controller is used to create a linear-in-the-coefficients problem. This means that the definitions given in equations (3.3), (3.4), (3.5) and (3.6) hold, but that there now is a time dependency of the controller coefficients on the sample instant k . For example, all controller coefficients are now grouped in a vector $\mathbf{w}(k) \in \mathbb{R}^{JKL}$ (compare this with equation (3.6)).

At the core of the FxLMS algorithm is the least mean square (LMS) algorithm. This algorithm is described in many articles and text books and is given by:

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \frac{\mu}{2} \underbrace{\frac{\partial J(k)}{\partial \mathbf{w}(k)}}_{\text{Gradient}}, \quad (3.35)$$

where the step-size μ needs to be chosen by the user (Rupp and Sayed, 1998). See also e.g. (Widrow and Walach, 1996; Sayed, 2003; Haykin and Widrow, 2003; Kuo and Morgan, 1996). The interpretation of this algorithm is that for every sample instant the old controller coefficients $\mathbf{w}(k)$ are adjusted by a small amount ($\mu/2$) in the direction of the negative gradient of the cost function $J(k)$, to obtain the new controller coefficients $\mathbf{w}(k+1)$. As an illustration: in the two-dimensional case depicted in figure 3.2, the goal is to find the (global) minimum of the paraboloid by adapting the controller coefficients - which are initialized at a random point on the paraboloid - for every new sample.

The FxLMS algorithm follows logically from the LMS algorithm. Only the gradient in equation (3.35) has to be worked out to see the *filtered-x* part in the name FxLMS. For *deriving* the gradient the assumption needs to be made that the controller $\mathbf{W}(k)$ changes slowly in time, so that the controller $\mathbf{W}(k)$ and the secondary path \mathbf{S} can be interchanged (or swapped, as it is often called)⁷. The interchange assumption is well-known and is discussed in literature in, for example, (Elliott, 2000*b*; Kuo and Morgan, 1996; Kim et al., 1994). Making this assumption, equation (3.7) can be written as:

$$\mathbf{e}(k) = \mathbf{d}(k) + \zeta(k) + \mathbf{R}_x(k)\mathbf{w}(k). \quad (3.36)$$

Substituting this expression for $\mathbf{e}(k)$ in equation (3.34) gives:

$$\begin{aligned} J(k) &= [\mathbf{d}(k) + \zeta(k) + \mathbf{R}_x(k)\mathbf{w}(k)]^T [\mathbf{d}(k) + \zeta(k) + \mathbf{R}_x(k)\mathbf{w}(k)] \\ &= \mathbf{d}^T(k)\mathbf{d}(k) + \zeta^T(k)\zeta(k) + 2\mathbf{d}^T(k)\zeta(k) + \mathbf{w}^T(k)\mathbf{R}_x^T(k)\mathbf{R}_x(k)\mathbf{w}(k) + \\ &\quad + 2\mathbf{w}^T(k)\mathbf{R}_x^T(k)\mathbf{d}(k) + 2\mathbf{w}^T(k)\mathbf{R}_x^T(k)\zeta(k). \end{aligned} \quad (3.37)$$

Using the latter expression, the gradient $\partial J(k)/\partial \mathbf{w}(k)$ in equation (3.35) can be written as:

$$\begin{aligned} \frac{\partial J(k)}{\partial \mathbf{w}(k)} &= 2 [\mathbf{R}_x^T(k)\mathbf{d}(k) + \mathbf{R}_x^T(k)\zeta(k) + \mathbf{R}_x^T(k)\mathbf{R}_x(k)\mathbf{w}(k)] \\ &= 2\mathbf{R}_x^T(k)\mathbf{e}(k). \end{aligned}$$

Substituting this result in equation (3.35) gives the FxLMS algorithm:

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{w}(k) - \mu \mathbf{R}_x^T(k)\mathbf{e}(k) \\ &= \mathbf{w}(k) - \mu (\mathbf{S}^T \otimes \bar{\mathbf{x}}(k)) \mathbf{e}(k), \end{aligned} \quad (3.38)$$

The reason why this is called the *filtered-x* LMS algorithm is clear from the second expression in equation (3.38) and has to do with the matrix $(\mathbf{S}^T \otimes \bar{\mathbf{x}}(k))$. This matrix demonstrates that the reference signal $\mathbf{x}(k)$ that is contained in the regression vector $\bar{\mathbf{x}}(k)$, is *filtered* with the secondary path⁸ \mathbf{S} in the sense of the Kronecker tensor product. A block diagram of the FxLMS based MIMO feedforward adaptive SD-AVIC system is shown in figure 3.7. For more information, see also (Rupp and Sayed, 1998) in which robust versions are proposed and (Douglas, 1997, 1999) in which fast implementations of the FxLMS algorithm are considered. Consult (Hassibi et al., 1996) for information on as to why the FxLMS algorithm is robust. Now follows an example of the FxLMS algorithm.

⁷In general, two linear time invariant plants in series can be interchanged while keeping the same input/output relationship. A linear time varying plant, such as the controller $\mathbf{W}(k)$, however, might not be interchangeable with a linear time invariant plant \mathbf{S} , because in that case the same input/output relationship is not maintained. However, if the controller $\mathbf{W}(k)$ changes slowly in time, it is safe to interchange the plants.

⁸In a real-time situation the true secondary path \mathbf{S} is not known and a model $\hat{\mathbf{S}}$ needs to be used, so that:

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{w}(k) - \mu \hat{\mathbf{R}}_{xx}^T(k)\mathbf{e}(k) \\ &= \mathbf{w}(k) - \mu (\hat{\mathbf{S}}^T \otimes \bar{\mathbf{x}}(k)) \mathbf{e}(k). \end{aligned} \quad (3.39)$$

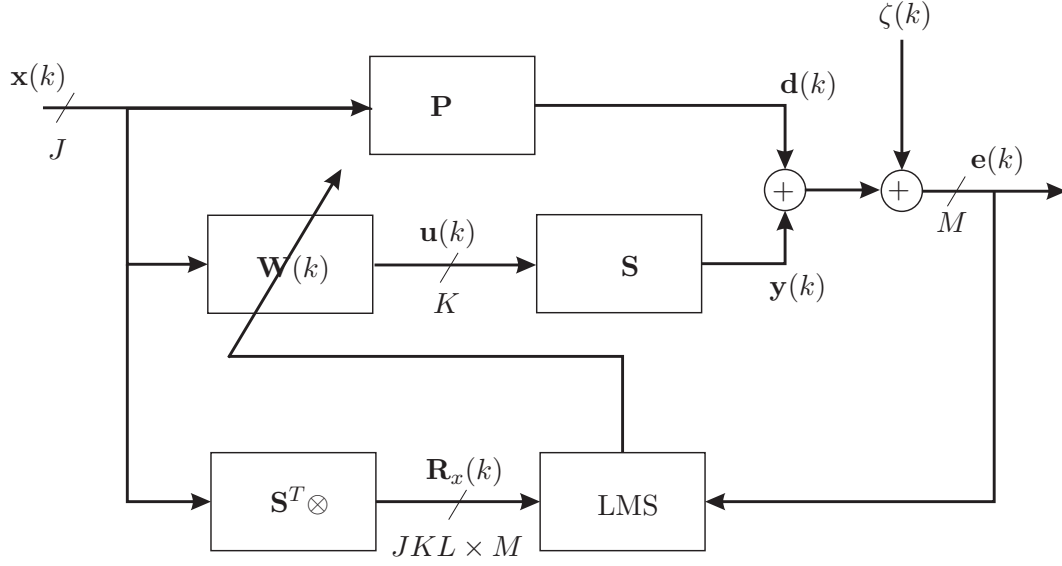


Figure 3.7: Block diagram of a MIMO feedforward SD-AVIC system with an adaptive controller $\mathbf{W}(k)$ of which the coefficients are updated with the FxLMS algorithm.

Example 3.3.1 (FxLMS algorithm). Consider the controller \mathbf{W} as described in example 3.2.1 on page 37 to now be dependent on time and indicated by $\mathbf{W}(k)$, and the secondary path \mathbf{S} as in example 3.2.1. In the example, it is observed that:

$$(\mathbf{S} \otimes \bar{\mathbf{x}}^T(k)) = \left(\left[\begin{array}{c} s_1^1 + s_2^1 q^{-1} \\ s_1^2 + s_2^2 q^{-1} \end{array} \right] \otimes [x(k) \ x(k-1)] \right).$$

Therefore:

$$\begin{aligned} (\mathbf{S}^T \otimes \bar{\mathbf{x}}(k)) &= \left(\left[\begin{array}{cc} s_1^1 + s_2^1 q^{-1} & s_1^2 + s_2^2 q^{-1} \end{array} \right] \otimes \left[\begin{array}{c} x(k) \\ x(k-1) \end{array} \right] \right) \\ &= \left[\begin{array}{cc} (s_1^1 + s_2^1 q^{-1}) x(k) & (s_1^2 + s_2^2 q^{-1}) x(k) \\ (s_1^1 + s_2^1 q^{-1}) x(k-1) & (s_1^2 + s_2^2 q^{-1}) x(k-1) \end{array} \right] \\ &= \left[\begin{array}{cc} s_1^1 x(k) + s_2^1 x(k-1) & s_1^2 x(k) + s_2^2 x(k-1) \\ s_1^1 x(k-1) + s_2^1 x(k-2) & s_1^2 x(k-1) + s_2^2 x(k-2) \end{array} \right]. \end{aligned}$$

Substituting that result in the second expression in equation (3.38), in this particular example gives the following FxLMS algorithm:

$$\begin{aligned} \begin{bmatrix} w_0(k+1) \\ w_1(k+1) \end{bmatrix} &= \begin{bmatrix} w_0(k) \\ w_1(k) \end{bmatrix} \\ &+ \mu \begin{bmatrix} s_1^1 x(k) + s_2^1 x(k-1) & s_1^2 x(k) + s_2^2 x(k-1) \\ s_1^1 x(k-1) + s_2^1 x(k-2) & s_1^2 x(k-1) + s_2^2 x(k-2) \end{bmatrix} \begin{bmatrix} e_1(k) \\ e_2(k) \end{bmatrix}. \end{aligned}$$

From this equation it is clear that the FxLMS algorithm is straightforward to implement. For every sample instant, only a correction needs to be made to the old controller coefficients at

sample k to obtain the new controller coefficients at sample $k+1$. This correction is completely defined in terms of additions and multiplications. There are no inversions involved, which are computationally intensive.

3.3.2 Convergence analysis of the adaptive controller coefficients

Optimum performance of the adaptive controller is only obtained when the controller coefficients are converged. Convergence can require a significant amount of time. To get more insight into the convergence time, it is useful to analyze the FxLMS algorithm. That is what is described in this section. See also (Elliott, 2000b; Kuo and Morgan, 1996; Bjarnason, 1995; Wang and Ren, 1999; Rupp and Sayed, 1996) for more details on convergence analysis. Basically, the main interest is the time it takes for the controller coefficients to converge to the optimum controller coefficients in the mean or the sense of the statistical expectation. Applying the statistical expectation to the left hand side and the right hand side of equation (3.38) gives:

$$\begin{aligned} E\{\mathbf{w}(k+1)\} &= E\{\mathbf{w}(k)\} - \mu E\{\mathbf{R}_x^T(k)[\mathbf{d}(k) + \zeta(k) + \mathbf{R}_x(k)\mathbf{w}(k)]\} \\ &= E\{\mathbf{w}(k)\} - \mu E\{\mathbf{R}_x^T(k)\mathbf{d}(k) + \mathbf{R}_x^T(k)\zeta(k) + \mathbf{R}_x^T(k)\mathbf{R}_x(k)\mathbf{w}(k)\} \\ &= E\{\mathbf{w}(k)\} - \mu E\{\mathbf{R}_x^T(k)\mathbf{d}(k)\} - E\{\mathbf{R}_x^T(k)\zeta(k)\} \\ &\quad - \mu E\{\mathbf{R}_x^T(k)\mathbf{R}_x(k)\mathbf{w}(k)\}. \end{aligned}$$

Assuming that the filter coefficients $\mathbf{w}(k)$ only change slowly in time, and assuming that the reference inputs $\mathbf{x}(k)$ are uncorrelated to the measurement noise $\zeta(k)$ (so that $E\{\mathbf{R}_x^T(k)\zeta(k)\} = \mathbf{0}$), the last expression in the equation above can be written as:

$$\begin{aligned} E\{\mathbf{w}(k+1)\} &= E\{\mathbf{w}(k)\} - \mu E\{\mathbf{R}_x^T(k)\mathbf{d}(k)\} - \mu E\{\mathbf{R}_x^T(k)\mathbf{R}_x(k)\} E\{\mathbf{w}(k)\} \\ &= E\{\mathbf{w}(k)\} - \mu E\{\mathbf{R}_x^T(k)\mathbf{R}_x(k)\} \mathbf{w}_{opt} - \mu E\{\mathbf{R}_x^T(k)\mathbf{R}_x(k)\} E\{\mathbf{w}(k)\}. \end{aligned} \quad (3.40)$$

Here, the transition from the first to the second expression is accomplished using equation (3.25), i.e. $E\{\mathbf{R}_x^T(k)\mathbf{d}(k)\} = E\{\mathbf{R}_x^T(k)\mathbf{R}_x(k)\} \mathbf{w}_{opt}$. Subtracting the optimum solution \mathbf{w}_{opt} from both the left hand side and the right hand side of the last expression of equation (3.40) and collecting terms, gives:

$$E\{\mathbf{w}(k+1)\} - \mathbf{w}_{opt} = [\mathbf{I}_{JKL} - \mu E\{\mathbf{R}_x^T(k)\mathbf{R}_x(k)\}] (E\{\mathbf{w}(k)\} - \mathbf{w}_{opt}). \quad (3.41)$$

Defining $\mathbf{v}(k) = E\{\mathbf{w}(k)\} - \mathbf{w}_{opt}$, equation (3.41) can be rewritten as:

$$\mathbf{v}(k+1) = (\mathbf{I}_{JKL} - \mu E\{\mathbf{R}_x^T(k)\mathbf{R}_x(k)\}) \mathbf{v}(k). \quad (3.42)$$

The auto-correlation matrix $E\{\mathbf{R}_x^T(k)\mathbf{R}_x(k)\}$ can be decomposed in a diagonal matrix Λ and a symmetric unitary matrix \mathbf{Q} so that (Strang, 1988; Golub and Loan, 1996):

$$E\{\mathbf{R}_x^T(k)\mathbf{R}_x(k)\} = \mathbf{Q}\Lambda\mathbf{Q}^T, \quad (3.43)$$

where $\mathbf{Q}\mathbf{Q}^T = \mathbf{Q}^T\mathbf{Q} = \mathbf{I}$. Substituting equation (3.43) in equation (3.42) and defining the vector $\underline{\mathbf{v}}(k) = \mathbf{Q}^T\mathbf{v}(k)$, gives:

$$\underline{\mathbf{v}}(k+1) = (\mathbf{I} - \mu\Lambda) \underline{\mathbf{v}}(k).$$

Since Λ is a diagonal matrix, the components of $\underline{v}(k)$ can be further decoupled as:

$$\underline{v}_n(k+1) = (1 - \mu\lambda_n)\underline{v}_n(k). \quad (3.44)$$

Specifically, every eigenvalue λ_n corresponds to a so-called convergence mode. The so-called *convergence time constant* is defined as the time that is required for any transient to decay to $1/e$ of its initial value (Kuo and Morgan, 1996). Obviously, it is true that:

$$\underline{v}_n(k+1) = (1 - \mu\lambda_n)^k \underline{v}_n(0), \quad (3.45)$$

with $(n = 1, \dots, JKL)$. This equation is the basis for estimating how much time it takes before the adaptive controller coefficients have reached the optimum controller coefficients (in the mean). In order to have a stable adaptation process, it has to be true that:

$$|1 - \mu\lambda_n| < 1.$$

In order for this constraint to be met, the step-size μ has to satisfy:

$$0 < \mu < \frac{2}{\lambda_{max}}. \quad (3.46)$$

For a small μ ($\mu\lambda_n \ll 1$), equation (3.44) can also be written as (Kuo and Morgan, 1996):

$$\underline{v}_n(k+1) = e^{-\mu\lambda_n k} \underline{v}_n(0).$$

The time required for $\underline{v}_n(k)$ to decay from $\underline{v}_n(0)$ to $\underline{v}_n(0)/e$ can now be computed as follows. Note that:

$$\underline{v}_n(0)/e = e^{-\mu\lambda_n k} \underline{v}_n(0).$$

By dividing both the left hand side and the right hand side by $\underline{v}_n(0)$, this equals:

$$e^{-1} = e^{-\mu\lambda_n k}.$$

Equalizing the exponents, gives:

$$-\mu\lambda_n k = -1.$$

Thus, it can be concluded that:

$$k = \frac{1}{\mu\lambda_n}.$$

The interpretation of this equation is that it requires k samples for the convergence mode that corresponds to λ_n , to reach its $1/e$ value. Both the left hand side and the right hand side of the equation have to be multiplied with the sample time T to obtain a time constant in seconds:

$$\tau_n = \frac{T}{\mu\lambda_n} \text{ s.} \quad (3.47)$$

From this expression a minimum time constant τ_{min} can be derived:

$$\tau_{min} = \frac{T}{\mu\lambda_{max}}, \quad (3.48)$$

that corresponds to the maximum eigenvalue λ_{max} and a maximum time constant τ_{max} :

$$\tau_{max} = \frac{T}{\mu\lambda_{min}},$$

that corresponds to the minimum eigenvalue λ_{min} . The step-size defined in equation (3.46) minimizes τ_{min} . Fastest convergence is guaranteed if $\lambda_{max} = \lambda_{min}$ or when τ_{min} and τ_{max} coincide. The ratio between the minimum time constant and the maximum time constant is given by:

$$\Theta = \frac{\tau_{max}}{\tau_{min}} = \frac{\frac{T}{\mu\lambda_{min}}}{\frac{T}{\mu\lambda_{max}}} = \frac{\lambda_{max}}{\lambda_{min}}, \quad (3.49)$$

which is called the eigenvalue-spread. Optimum convergence is guaranteed if the eigenvalue spread is equal to unity. This can be established by ensuring that the auto-correlation matrix in equation (3.43) equals for example the identity matrix so that all eigenvalues λ_n are identical. This means that it has to be made sure that:

$$\mathbb{E} \{ \mathbf{R}_x^T(k) \mathbf{R}_x(k) \} = \mathbb{E} \{ (\mathbf{S}^T \otimes \bar{\mathbf{x}}(k)) (\mathbf{S} \otimes \bar{\mathbf{x}}^T(k)) \} = \mathbf{I}_{JKL},$$

which can be established if the secondary path is an identity gain $\mathbf{I}_{M \times K}$, so that:

$$\mathbb{E} \{ \mathbf{R}_x^T(k) \mathbf{R}_x(k) \} = \mathbb{E} \{ \mathbf{X}(k) \mathbf{X}^T(k) \} = \mathbf{I}_{JKL}. \quad (3.50)$$

That is accomplished if the reference inputs $\mathbf{x}(k)$ are unit variance uncorrelated white noise sequences. In general the problem is, however, that the secondary path \mathbf{S} is not an identity gain. This implies that even if the reference inputs $\mathbf{x}(k)$ are unit variance uncorrelated white noise sequences⁹, the auto-correlation matrix in equation (3.43) will not be equal to the identity matrix due to influence of the secondary path \mathbf{S} . In other words: the influence of the secondary path \mathbf{S} hampers the performance speed of the adaptive controller coefficients $\mathbf{w}(k)$. This is a common problem, which is described in, for example, (Elliott, 2000b; Kuo and Morgan, 1996; Widrow and Walach, 1996). Later on in this thesis, a solution to this problem is proposed by using post-conditioning.

3.3.3 Incorporation of state space models

Traditionally, the model $\hat{\mathbf{S}}$ of the secondary path \mathbf{S} used for filtering the reference signal, has a FIRF structure and is defined as:

$$\hat{\mathbf{S}} = \begin{bmatrix} \hat{\mathbf{s}}_{(1,1)} & \hat{\mathbf{s}}_{(1,2)} & \cdots & \hat{\mathbf{s}}_{(1,K)} \\ \hat{\mathbf{s}}_{(2,1)} & \hat{\mathbf{s}}_{(2,2)} & \cdots & \hat{\mathbf{s}}_{(2,K)} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\mathbf{s}}_{(M,1)} & \hat{\mathbf{s}}_{(M,2)} & \cdots & \hat{\mathbf{s}}_{(M,K)} \end{bmatrix}, \quad (3.51)$$

where the impulse response $\hat{\mathbf{s}}_{(m,n)}$ from the n^{th} actuator input to the m^{th} error sensor output is defined as:

$$\hat{\mathbf{s}}_{(m,n)}(q^{-1}) = \hat{\mathbf{s}}_{(m,n)}^{(1)} + \hat{\mathbf{s}}_{(m,n)}^{(2)} q^{-1} + \cdots + \hat{\mathbf{s}}_{(m,n)}^{(-F+1)} q^{F-2} + \hat{\mathbf{s}}_{(m,n)}^{(F)} q^{-F+1},$$

with F the length of the impulse response. However, in this thesis a state space model (SSM) of the secondary path is identified. This SSM needs to be incorporated in the FxLMS algorithm, which gives rise to a new hybrid algorithm. See (Nijssse et al., 1999) for details.

⁹Thus meeting the condition for optimal convergence of the LMS algorithm (Widrow and Walach, 1996).

The update equation per individual filter $\mathbf{w}_{ni}(k) \in \mathbb{R}^L$ is given by:

$$\mathbf{w}_{ni}(k+1) = \mathbf{w}_{ni}(k) + \mu \sum_{m=1}^M [\widehat{\mathbf{s}}_{(m,n)} \mathbf{x}_i(k)] e_m(n).$$

This update equation can also be written as:

$$\begin{bmatrix} w_1^{ni}(k+1) \\ w_2^{ni}(k+1) \\ \vdots \\ w_L^{ni}(k+1) \end{bmatrix} = \begin{bmatrix} w_1^{ni}(k) \\ w_2^{ni}(k) \\ \vdots \\ w_L^{ni}(k) \end{bmatrix} + \begin{bmatrix} \mu \sum_{m=1}^M [\widehat{\mathbf{s}}_{(m,n)} x_j(k)] e_m(k) \\ \mu \sum_{m=1}^M [\widehat{\mathbf{s}}_{(m,n)} x_j(k-1)] e_m(k) \\ \vdots \\ \mu \sum_{m=1}^M [\widehat{\mathbf{s}}_{(m,n)} x_j(k-L+1)] e_m(k) \end{bmatrix}. \quad (3.52)$$

$e_m(k)$ represents the m^{th} error sensor output. Now focus on the computationally complex part of equation (3.52), i.e. the summation and the convolution. More specifically, focus on the top element of the second part of equation (3.52), which corresponds to $w_1^{ni}(k+1)$ and which will be denoted by $p_{ni}(k)$:

$$p_{ni}(k) = \sum_{m=1}^M [\widehat{\mathbf{s}}_{(m,n)}(k) x_j(k)] e_m(k).$$

This can be written as:

$$p_{ni}(k) = [\widehat{\mathbf{s}}_{(1,n)} x_j(k)] e_1(k) + [\widehat{\mathbf{s}}_{(2,n)} x_j(k)] e_2(k) + \cdots + [\widehat{\mathbf{s}}_{(M,n)} x_j(k)] e_M(k).$$

This expression can be written in a vector-vector multiplication as follows:

$$p_{ni}(k) = \underbrace{\begin{bmatrix} e_1(k) & e_2(k) & \cdots & e_M(k) \end{bmatrix}}_{\text{Error: } \mathbf{e}^T(k)} \underbrace{\begin{bmatrix} \widehat{\mathbf{s}}_{(1,n)} x_i(k) \\ \widehat{\mathbf{s}}_{(2,n)} x_i(k) \\ \vdots \\ \widehat{\mathbf{s}}_{(M,n)} x_i(k) \end{bmatrix}}_{\mathbf{o}_{ni}(k)}. \quad (3.53)$$

The vector $\mathbf{o}_{ni}(k)$ is produced by the subset of the secondary path from the n^{th} input to all error sensor outputs, with input $x_i(k)$. Thus, the vector $\mathbf{o}_{ni}(k)$ can be written as a SIMO SSF:

$$\begin{aligned} \mathbf{z}_{ni}(k+1) &= \mathbf{A} \mathbf{z}_{ni}(k) + \mathbf{b} x_i(k) \\ \mathbf{o}_{ni}(k) &= \mathbf{C} \mathbf{z}_{ni}(k) + \mathbf{d} x_i(k) \end{aligned} \quad (3.54)$$

where $\mathbf{A} \in \mathbb{R}^{N \times N}$, $\mathbf{b} \in \mathbb{R}^{N \times 1}$, $\mathbf{C} \in \mathbb{R}^{M \times N}$ and $\mathbf{d} \in \mathbb{R}^{M \times 1}$, with N the state dimension. Since there are J reference inputs and K actuator inputs, this means that to be able to compute the quantity defined in equation (3.53) for all ni filters, a total of $K \times J$ SSFs has to be computed. Using this approach may diminish the number of computations needed: see (Nijssen et al., 1999) for details. Note that the KJ SSFs can further be compressed in a new SSF, with J inputs and KM outputs.

3.3.4 Improving the robustness of the adaptive controller with regularization

A regularization term can be added to the cost function in equation (3.34) to increase the robustness of the adaptive controller (similar to equation (3.23)):

$$\check{J}(k) = \mathbf{e}^T(k)\mathbf{e}(k) + \rho \mathbf{w}^T(k)\mathbf{w}(k). \quad (3.55)$$

with $\rho \in \mathbb{R}^+$ a small positive constant. Applying this cost function, the regularized FxLMS algorithm becomes:

$$\check{\mathbf{w}}(k+1) = \check{\mathbf{w}}(k) - \frac{\mu}{2} \frac{\partial \check{J}(k)}{\partial \mathbf{w}(k)}. \quad (3.56)$$

Here, the reasoning can go along similar lines as with the derivation of the FxLMS algorithm. Specifically, it can be written that:

$$\begin{aligned} \check{J}(k) &= [\mathbf{d}(k) + \zeta(k) + \mathbf{R}_x(k)\mathbf{w}(k)]^T [\mathbf{d}(k) + \zeta(k) + \mathbf{R}_x(k)\mathbf{w}(k)] + \rho \mathbf{w}^T(k)\mathbf{w}(k) \\ &= \mathbf{d}^T(k)\mathbf{d}(k) + \zeta^T(k)\zeta(k) + \mathbf{w}^T(k)\mathbf{R}_x^T(k)\mathbf{R}_x(k)\mathbf{w}(k) + 2\mathbf{w}^T(k)\mathbf{R}_x^T(k)\mathbf{d}(k) \\ &\quad + 2\mathbf{w}^T(k)\mathbf{R}_x^T(k)\zeta(k) + \rho \mathbf{w}^T(k)\mathbf{w}(k). \end{aligned}$$

Using this expression, $\partial \check{J}(k)/\partial \mathbf{w}(k)$ in equation (3.56) can be written as:

$$\begin{aligned} \frac{\partial \check{J}(k)}{\partial \mathbf{w}(k)} &= 2 [\mathbf{R}_x^T(k)\mathbf{d}(k) + \mathbf{R}_x^T(k)\zeta(k) + \mathbf{R}_x^T(k)\mathbf{R}_x(k)\mathbf{w}(k) + \rho \mathbf{w}(k)] \\ &= 2 [\mathbf{R}_x^T(k)\mathbf{e}(k) + \rho \mathbf{w}(k)]. \end{aligned}$$

Substituting this result in the lower expression of equation (3.56) gives the regularized FxLMS algorithm:

$$\check{\mathbf{w}}(k+1) = (1 - \mu\rho)\check{\mathbf{w}}(k) - \mu\mathbf{R}_x^T(k)\mathbf{e}(k). \quad (3.57)$$

This algorithm is also called the leaky FxLMS algorithm, since the coefficients leak away due to the regularization term ρ (i.e. assuming $\mu \neq 0$ and the gradient is zero).

3.4 Post-conditioning the adaptive controller for faster convergence

The convergence time of the adaptive controller coefficients $\mathbf{w}(k)$ can be slow due to the Kronecker filtering with the secondary path \mathbf{S} (see equation (3.38)). In this section it is explained how the convergence time can be increased by post-conditioning the controller, i.e. by incorporating advance knowledge of the secondary path \mathbf{S} in the control scheme. In section 3.4.1 it is discussed how post-conditioning can be applied to the FxLMS algorithm. Section 3.4.2 describes how post-conditioning can be applied to the filtered-error LMS algorithm. In this algorithm, instead of the reference inputs the error sensor outputs are filtered (Wan, 1996).

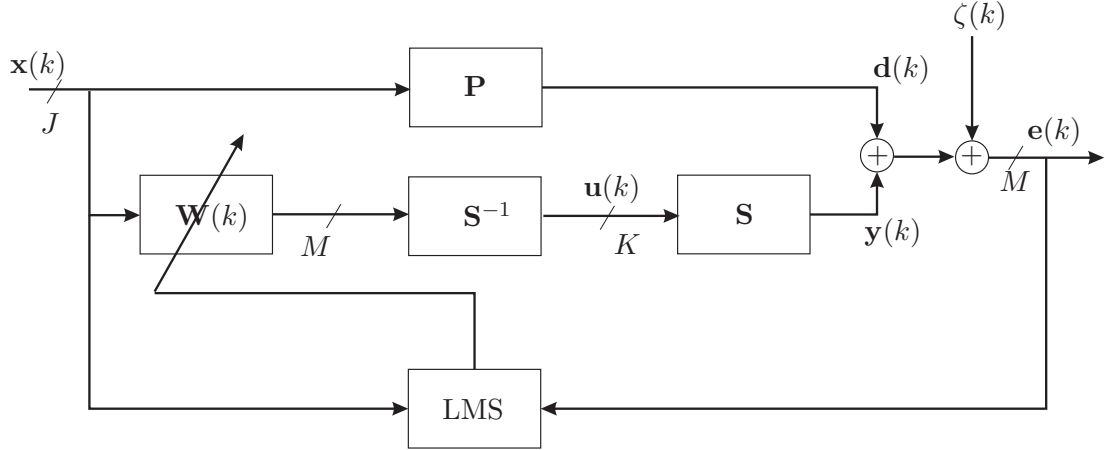


Figure 3.8: Block diagram of a MIMO feedforward SD-AVIC system with an FxLMS based adaptive controller and secondary path equalization by incorporating the inverse \mathbf{S}^{-1} .

3.4.1 Post-conditioning in the filtered-x least mean square algorithm

Due to the impact of the secondary path \mathbf{S} on the convergence speed it can take several hours, or even days, before the controller coefficients are converged. Although in some applications the convergence speed is less critical, slow convergence is not desirable in general and thus a way needs to be found to cancel out the impact of the secondary path on the convergence speed of the adaptive controller coefficients $\mathbf{w}(k)$. Ideally, the inverse \mathbf{S}^{-1} of the secondary path \mathbf{S} needs to be incorporated in the control scheme, as depicted in figure 3.8. However, the inverse \mathbf{S}^{-1} cannot be computed in a practical situation, since the secondary path \mathbf{S} has commonly zeros outside the unit circle. For this, see the remark on page 34 and also (Nelson et al., 1995; Miyoshi and Kaneda, 1988; Tokuno et al., 1997). In (Elliott, 1999) it is proposed to split up the secondary path \mathbf{S} in an inner factor \mathbf{S}_i and an outer factor \mathbf{S}_o by performing an inner/outer factorization of the secondary path \mathbf{S} (Vidyasagar, 1985; Oarâ and Varga, 1999):

$$\mathbf{S} = \mathbf{S}_i \mathbf{S}_o, \quad (3.58)$$

with the M output K input inner factor \mathbf{S}_i all pass, which means that:

$$\mathbf{S}_i^* \mathbf{S}_i = \mathbf{I}_K,$$

with \mathbf{S}_i^* the adjoint and at the same time¹⁰ left inverse of \mathbf{S}_i . See appendix A for details on the adjoint. The K input K output outer factor \mathbf{S}_o has the property that it can be stably inverted. In general, the main idea of the inner/outer factorization is to factorize a stable system in a minimum phase part that extracts all poles and all zeros that are inside the unit circle (i.e. \mathbf{S}_o) and an all-pass part that contains all zeros that are outside the unit circle (i.e. \mathbf{S}_i). An all-pass plant has unity magnitude in the frequency domain and only invokes a

¹⁰In this case the adjoint and the inverse coincide. In general, the inverse is not identical to the adjoint.

phase-shift upon a signal input to the plant. Due to the fact that \mathbf{S}_i is all-pass, it holds that:

$$\begin{aligned}\mathbf{S}^*\mathbf{S} &= \mathbf{S}_o^* \overbrace{\mathbf{S}_i^* \mathbf{S}_i}^{\mathbf{I}_K} \mathbf{S}_o \\ &= \mathbf{S}_o^* \mathbf{S}_o,\end{aligned}\tag{3.59}$$

which will prove helpful later on. Before it is explained how this inner/outer factorization can be used, an example is given.

Example 3.4.1 (Inner/outer factorization). *Given is the following secondary path:*

$$\mathbf{S} = \frac{(2 - 3q^{-1})(1 - 0.6q^{-1})}{(1 - 0.5q^{-1})(1 - 0.8q^{-1})},$$

which has two zeros (at 1.5 and at 0.6) and two poles (at 0.5 and 0.8). The secondary path \mathbf{S} is stable, but cannot be inverted stably due to the zero that is outside the unit circle. An inner/outer factorization can be computed, though, and the outer factor can be inverted as a best approximation. For this SISO second order secondary path \mathbf{S} the inner/outer factorization can be done manually in three steps:

1. Factorize the secondary path \mathbf{S} in two parts, extracting the zero that is outside the unit circle:

$$\mathbf{S} = \underbrace{(2 - 3q^{-1})}_{\text{Part I}} \underbrace{\frac{(1 - 0.6q^{-1})}{(1 - .5q^{-1})(1 - .8q^{-1})}}_{\text{Part II}}.$$

The subsystem indicated by Part I is not all-pass and the subsystem indicated by Part II is not causal (see section A.4 for an explanation of this). Therefore, both subsystems by themselves are not very useful.

2. Introduce an extra pole/zero pair into the system which is indicated below by Part III. The location of the pole and the zero are set to the inverse of the zero, i.e. to $1/(3/2) = 2/3$:

$$\mathbf{S} = \underbrace{(2 - 3q^{-1})}_{\text{Part I}} \underbrace{\frac{(1 - 0.6q^{-1})}{(1 - .5q^{-1})(1 - .8q^{-1})}}_{\text{Part II}} \underbrace{\frac{(3 - 2q^{-1})}{(3 - 2q^{-1})}}_{\text{Part III}}.$$

Part III is in itself a simple pole/zero cancellation and equal to unity. Thus, it does not change the input/output relationship of \mathbf{S} . However, the pole-zero cancellation can be used conveniently.

3. Take the denominator of Part III and cascade it with Part I to make it an all-pass system. Furthermore, take the numerator of Part III and shift it to Part II to make that system causal:

$$\begin{aligned}\mathbf{S} &= \underbrace{(2 - 3q^{-1})}_{\text{Part I}} \underbrace{\frac{(3 - 2q^{-1}) \rightarrow}{\leftarrow (3 - 2q^{-1})}}_{\text{Part III}} \underbrace{\frac{(1 - 0.6q^{-1})}{(1 - .5q^{-1})(1 - .8q^{-1})}}_{\text{Part II}} \\ &= \frac{(2 - 3q^{-1})(3 - 2q^{-1})(1 - 0.6q^{-1})}{(3 - 2q^{-1})(1 - .5q^{-1})(1 - .8q^{-1})}.\end{aligned}$$

The result is that the system is split up into an inner factor \mathbf{S}_i and an outer factor \mathbf{S}_o :

$$\mathbf{S}_i = \frac{2 - 3q^{-1}}{3 - 2q^{-1}}, \quad \mathbf{S}_o = \frac{(3 - 2q^{-1})(1 - 0.6q^{-1})}{(1 - .5q^{-1})(1 - .8q^{-1})}.$$

It holds that¹¹ $\mathbf{S}_i^* \mathbf{S}_i = 1$ and the outer factor \mathbf{S}_o can be stably inverted since now the two zeros and the two poles are inside the unit circle. From the example it is clear that the order of the inner factor \mathbf{S}_i is identical to the number of zeros that are outside the unit circle. The order of the outer factor \mathbf{S}_o is identical to the order of the secondary path \mathbf{S} . This is also true for general MIMO plants (Oarâ and Varga, 1999).

Using the result of the inner/outer factorization, a new adaptive control scheme is suggested in (Elliott, 1999) which is based on the FxLMS algorithm. Instead of minimizing the cost function, as the FxLMS algorithm does:

$$\begin{aligned} J(k) &= \mathbf{e}^T(k) \mathbf{e}(k) \\ &= \|\mathbf{e}(k)\|_2^2 \\ &= \|[\mathbf{P} + \mathbf{S}\mathbf{W}(k)] \mathbf{x}(k)\|_2^2, \end{aligned} \tag{3.60}$$

it is proposed in (Elliott, 1999) to include the advance information of the inverse of the outer factor, i.e. \mathbf{S}_o^{-1} , in the adaptive control scheme and to minimize:

$$J_o(k) = \|[\mathbf{P} + \mathbf{S}\mathbf{S}_o^{-1} \mathbf{W}_o(k)] \mathbf{x}(k)\|_2^2, \tag{3.61}$$

with respect to the adaptive controller coefficients $\mathbf{w}_o(k)$ of $\mathbf{W}_o(k)$, which are defined equivalent to the controller coefficients $\mathbf{w}(k)$ of $\mathbf{W}(k)$. Since $\mathbf{S}\mathbf{S}_o^{-1} = \mathbf{S}_i$, the cost function in equation (3.61) can be written as:

$$J_o(k) = \|[\mathbf{P} + \mathbf{S}_i \mathbf{W}_o(k)] \mathbf{x}(k)\|_2^2.$$

So, unlike the controller $\mathbf{W}(k)$ that acts on the secondary path \mathbf{S} as described by equation (3.60), the controller $\mathbf{W}_o(k)$ acts on the inner factor \mathbf{S}_i only. This gives the following update rule for the FxLMS algorithm, which is denoted as the IOFxLMS algorithm (compare this to e.g. equation (3.38)):

$$\mathbf{w}_o(k+1) = \mathbf{w}_o(k) - \mu \mathbf{R}_{xx}^i \mathbf{e}(k),$$

with $\mathbf{R}_{xx}^i = (\mathbf{S}_i^T \otimes \bar{\mathbf{x}}(k))$. Of course, in a real-time application a model $\hat{\mathbf{S}}_i$ has to be used, since the *true* inner factor \mathbf{S}_i is not known. With regularization the algorithm becomes:

$$\check{\mathbf{w}}_o(k+1) = (1 - \mu\rho) \check{\mathbf{w}}_o(k) - \mu \mathbf{R}_{xx}^i \mathbf{e}(k). \tag{3.62}$$

¹¹

$$\begin{aligned} \mathbf{S}_i^* \mathbf{S}_i &= \left(\frac{2 - 3q^{-1}}{3 - 2q^{-1}} \right)^* \left(\frac{2 - 3q^{-1}}{3 - 2q^{-1}} \right) = \left(\frac{2 - 3q}{3 - 2q} \right) \left(\frac{2 - 3q^{-1}}{3 - 2q^{-1}} \right) \\ &= \left(\frac{2q^{-1} - 3}{3q^{-1} - 2} \right) \left(\frac{2 - 3q^{-1}}{3 - 2q^{-1}} \right) = \left(\frac{3 - 2q^{-1}}{2 - 3q^{-1}} \right) \left(\frac{2 - 3q^{-1}}{3 - 2q^{-1}} \right) \\ &= 1. \end{aligned}$$

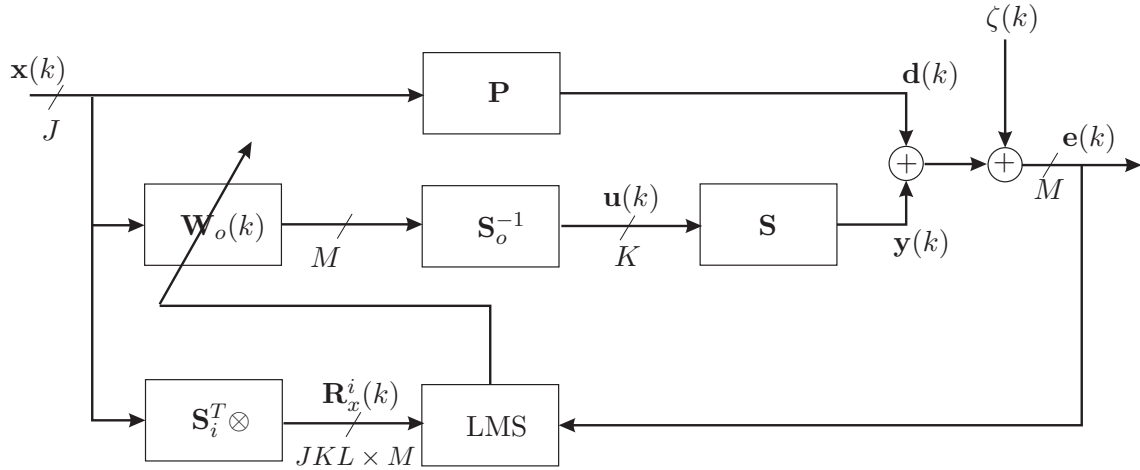


Figure 3.9: Block diagram of a MIMO feedforward SD-AVIC system with a FxLMS-based adaptive controller and the inverse of the outer factor included.

In (Nijse et al., 2001b) it is proposed to model the inverse outer factor by a state space model. In state space, the inner/outer factorization can be computed numerically stable. This gives a new hybrid algorithm, which may need less computations, especially for resonant plants.

Incorporating the inverse of the outer factor in the algorithm, enables faster convergence since the eigenvalue spread in equation (3.49) is now brought back to unity, since \mathbf{S}_i is all-pass. That is, if the reference inputs $\mathbf{x}(k)$ are e.g. assumed to be unit variance uncorrelated white noise sequences. For mathematical proof that faster convergence is established, see (van der Hoeven, 2002). A block diagram of the IOFxLMS based MIMO feedforward adaptive SD-AVIC system is shown in figure 3.9. The complete adaptive feedforward controller is given by¹²:

$$\check{\mathbf{W}}(k) = \mathbf{S}_o^{-1} \check{\mathbf{W}}_o(k), \quad (3.63)$$

and corresponds to the following regularized fixed gain controller:

$$\check{\mathbf{W}} = \mathbf{S}_o^{-1} \check{\mathbf{W}}_o. \quad (3.64)$$

The coefficients of the fixed gain controller $\check{\mathbf{W}}_o$ can be computed using equation (3.26). See figure 3.10 for a depiction of this, which is a slightly modified version of the scheme shown in figure 3.1.

Regularizing the inverse outer factor for stricter regularization

To prevent actuator saturation and to increase the robustness of the adaptive controller, besides the regularization parameter ρ included in equation (3.57), it is sometimes necessary to regularize the inverse outer factor \mathbf{S}_o^{-1} for stricter regularization. This new approach is called the two-step regularization approach (see also appendix F). The outer factor is determined so that it holds true that:

$$\check{\mathbf{S}}_o^* \check{\mathbf{S}}_o = \mathbf{S}_o^* \mathbf{S}_o + \varpi \mathbf{I}_K, \quad (3.65)$$

¹²Although \mathbf{S}_o^{-1} is not adaptive, $\check{\mathbf{W}}_o(k)$ is, and thus, $\check{\mathbf{W}}(k)$ can be considered to be adaptive.

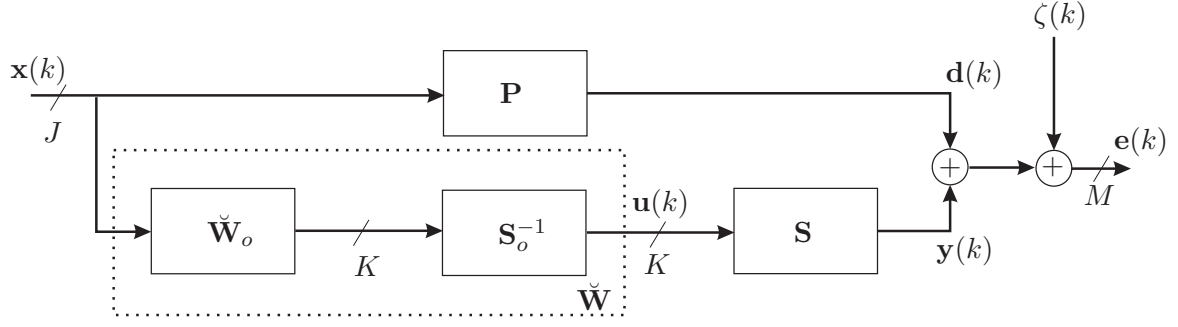


Figure 3.10: Block diagram of a MIMO feedforward SD-AVIC system with a fixed gain controller $\check{\mathbf{W}}$ that consists of an inverse outer factor $\check{\mathbf{S}}_o^{-1}$ and a regularized controller $\check{\mathbf{W}}_o$.

with $\varpi \in \mathbb{R}^+$ a small positive constant. Thus, \mathbf{S}_o^{-1} in figure 3.9 needs to be replaced by its regularized counterpart $\check{\mathbf{S}}_o^{-1}$. The ‘regularized’ inner factor $\check{\mathbf{S}}_i$ is given by:

$$\check{\mathbf{S}}_i = \mathbf{S}\check{\mathbf{S}}_o^{-1}. \quad (3.66)$$

Due to the regularized $\check{\mathbf{S}}_o^{-1}$ it is no longer true that $\check{\mathbf{S}}_i^* \check{\mathbf{S}}_i = \mathbf{1}$. The regularized FxLMS update from equation (3.57) now becomes:

$$\check{\mathbf{w}}_o(k+1) = (1 - \mu\rho)\check{\mathbf{w}}_o(k) - \mu\check{\mathbf{R}}_{xx}^i \mathbf{e}(k), \quad (3.67)$$

with $\check{\mathbf{R}}_{xx}^i = \left(\check{\mathbf{S}}_i^T \otimes \bar{\mathbf{x}}(k) \right)$, and the complete adaptive feedforward controller is now given by:

$$\check{\mathbf{W}}(k) = \check{\mathbf{S}}_o^{-1} \check{\mathbf{W}}_o(k). \quad (3.68)$$

The corresponding regularized fixed gain controller, which corresponds to the adaptive controller in equation (3.68), can be determined again (see figure 3.11) and is given by:

$$\check{\mathbf{W}} = \check{\mathbf{S}}_o^{-1} \check{\mathbf{W}}_o. \quad (3.69)$$

3.4.2 Application to the filtered-error least mean square algorithm

Another control scheme that can be used is depicted in figure 3.12. To compensate for the remaining inner factor \mathbf{S}_i , instead of filtering the reference signal with the inner factor \mathbf{S}_i in the Kronecker sense (see figure 3.9), the error sensor outputs $\mathbf{e}(k)$ are filtered with the adjoint \mathbf{S}_i^* of \mathbf{S}_i . This is computationally more attractive since the Kronecker filtering may require many computations, especially for large-order models (Nijssse et al., 1999). Assuming $\mathbf{d}(k) = \zeta(k) = \mathbf{0}$, the transfer function from the output of the controller $\mathbf{W}_o(k)$ to the filtered error sensor outputs $\tilde{\mathbf{e}}(k)$, is equal to identity:

$$\mathbf{S}_i^* \mathbf{S}_o^{-1} = \mathbf{I}_K,$$

so that the effect of the secondary path \mathbf{S} in the filter-update is completely compensated for. A problem is that \mathbf{S}_i^* is a SSF which is anti-causal¹³ and cannot be implemented in real-time.

¹³The output of the SSF depends on future inputs.

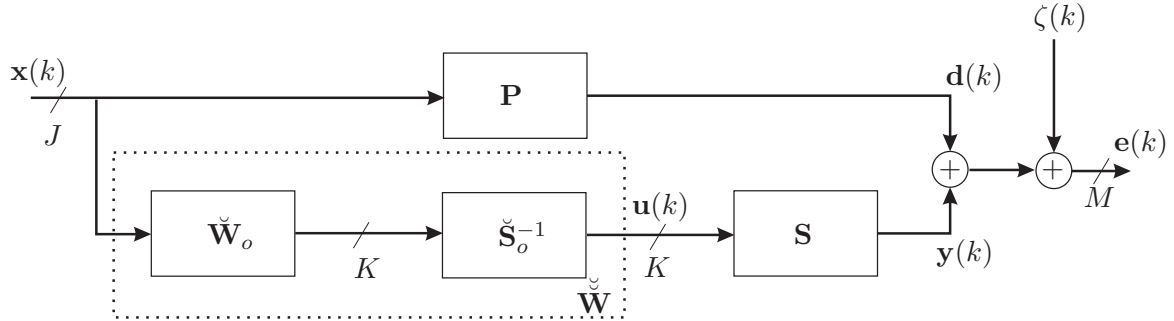


Figure 3.11: Block diagram of a MIMO feedforward SD-AVIC system with a fixed gain controller $\check{\check{W}}$ that consists of a regularized inverse outer factor $\check{\check{S}}_o^{-1}$ and a regularized controller $\check{\check{W}}_o$.

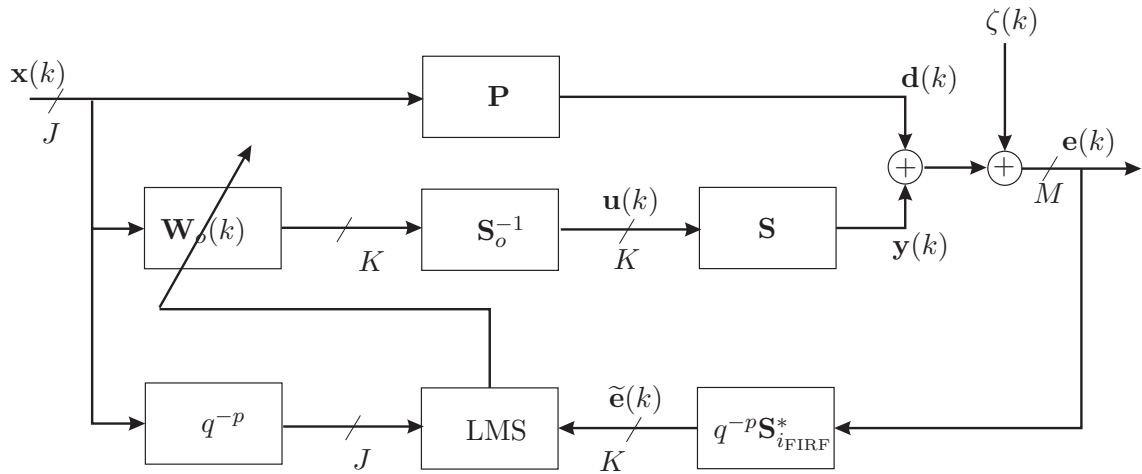


Figure 3.12: Block diagram of a MIMO feedforward SD-AVIC system with an adaptive controller and the inner/outer factorization.

To solve this problem, \mathbf{S}_i^* is converted to a FIRF of finite length p (indicated by $\mathbf{S}_{i_{\text{FIRF}}}^*$). The FIRF is still anti-causal, but this can now easily be solved by incorporating a delay of p samples. That makes the FIRF causal and thus implementable. An example is given to explain that.

Example 3.4.2 (Delayed causal FIRF). *Given are the following N -dimensional causal SISO SSF for \mathbf{S}_i , with state $\mathbf{z}(k)$, input $u(k)$ and output $y(k)$ and corresponding system matrices:*

$$\begin{aligned} \mathbf{z}(k+1) &= \mathbf{A}\mathbf{z}(k) + \mathbf{b}u(k) \\ y(k) &= \mathbf{c}\mathbf{z}(k) + \mathbf{d}u(k) \end{aligned} \quad .$$

The adjoint \mathbf{S}_i^* of this SSF is given by:

$$\begin{aligned} \mathbf{z}(k-1) &= \mathbf{A}^T\mathbf{z}(k) + \mathbf{c}^T u(k) \\ u(k) &= \mathbf{b}^T\mathbf{z}(k) + \mathbf{d}y(k) \end{aligned} \quad .$$

Obviously, the adjoint is not implementable, since the state \mathbf{z} at sample instant $k-1$ depends on the state \mathbf{z} at time instant k and the input \mathbf{u} at time instant k . The adjoint SSF can be converted to a FIRF of length p :

$$\mathbf{S}_{i_{\text{FIRF}}}^*(q^{-1}) = s_0^i + s_1^i q + s_2^i q^2 + \dots + s_{p-1}^i q^{p-1} + s_p^i q^p.$$

$\mathbf{S}_{i_{\text{FIRF}}}^*(q^{-1})$ is also anti-causal and cannot be implemented in real-time since it assumes knowledge concerning the future, i.e. q contains positive powers of p . Therefore a delay of p samples is applied to the adjoint $\mathbf{S}_{i_{\text{FIRF}}}^*(q^{-1})$, so that a causal and implementable filter is obtained:

$$q^{-p}\mathbf{S}_{i_{\text{FIRF}}}^*(q^{-1}) = s_p^i + s_{p-1}^i q^{-1} + \dots + s_2^i q^{-p+2} + s_1^i q^{-p+1} + s_0^i q^{-p}.$$

The FIRF $q^{-p}\mathbf{S}_{i_{\text{FIRF}}}^*$ is used to filter the error sensor outputs $\mathbf{e}(k)$. The number p needs to be chosen with care. Making p too small may result in performance degradation¹⁴. This results in the following filtered error sensor outputs $\tilde{\mathbf{e}}(k)$ that are actually used in the LMS update (see figure 3.12):

$$\tilde{\mathbf{e}}(k) = q^{-p}\mathbf{S}_{i_{\text{FIRF}}}^* \mathbf{e}(k). \quad (3.70)$$

Since the error sensor outputs $\mathbf{e}(k)$ are delayed by p samples, the reference inputs also need to be delayed by p samples. The resulting LMS algorithm is a modification of the traditional adjoint LMS algorithm (Wan, 1996) and is mentioned in e.g. (Elliott, 2000b):

$$\check{\mathbf{w}}_o(k+1) = (1 - \mu\rho)\check{\mathbf{w}}_o(k) - \mu\mathbf{X}(k-p)\tilde{\mathbf{e}}(k),$$

with $\mathbf{X}(k-p)$ a delayed version of $\mathbf{X}(k)$ as defined in equation (3.8). This algorithm is referred to as the inner/outer based adjoint LMS algorithm (IOALMS).

¹⁴In general, $\mathbf{S}_{i_{\text{FIRF}}}^*$ is obtained from \mathbf{S}_i as follows:

- Determine the impulse response of \mathbf{S}_i with a length of p samples (that need to be chosen by the user);
- The coefficients of this impulse response are equal to the coefficients of the FIRF $\mathbf{S}_{i_{\text{FIRF}}}$;
- $\mathbf{S}_{i_{\text{FIRF}}}^*$ can be obtained by taking the p coefficients of $\mathbf{S}_{i_{\text{FIRF}}}$ in reverse order (the first element of $\mathbf{S}_{i_{\text{FIRF}}}^*$ is the last of $\mathbf{S}_{i_{\text{FIRF}}}$, and so on).

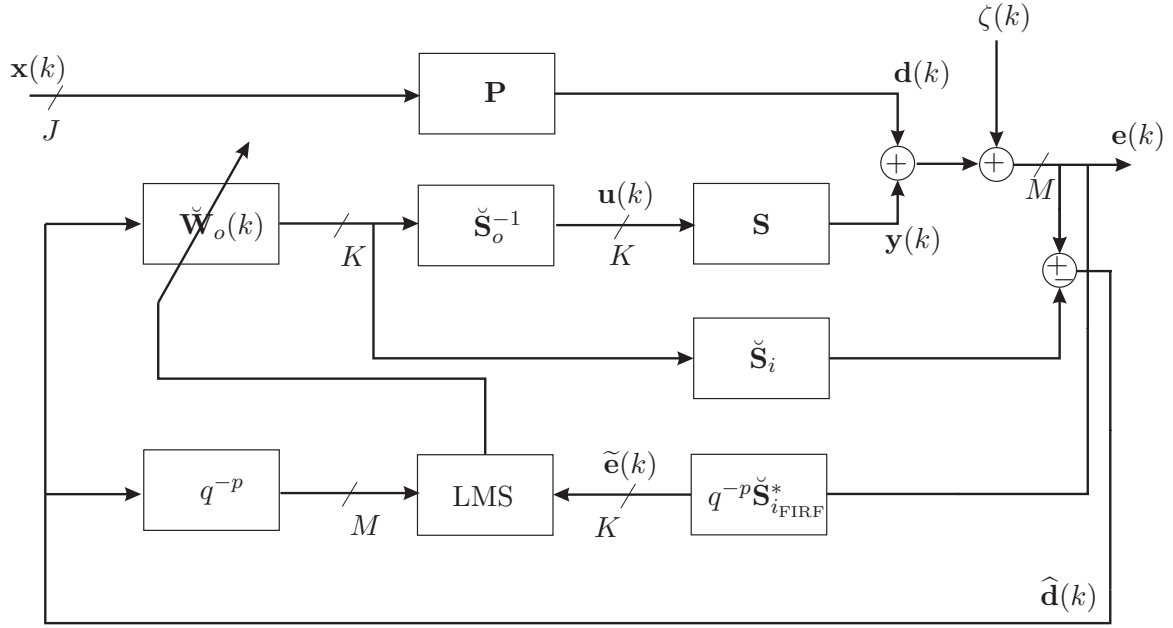


Figure 3.13: Block diagram of a MIMO feedback SD-AVIC system with an adaptive controller and a regularized inverse outer factor model.

Sometimes it is necessary to regularize the inverse outer factor, as indicated in equation (3.65). In that case in figure 3.12, \mathbf{S}_o^{-1} needs to be replaced by its regularized counterpart $\check{\mathbf{S}}_o^{-1}$ and \mathbf{S}_i^* by its regularized counterpart $\check{\mathbf{S}}_i^*$. The IOALMS algorithm then becomes:

$$\check{\mathbf{w}}_o(k+1) = (1 - \mu\rho)\check{\mathbf{w}}_o(k) - \mu\mathbf{X}(k-p)\tilde{\mathbf{e}}(k), \quad (3.71)$$

and the new adaptive controller is given by equation (3.68). The corresponding fixed gain controller is (still) given by equation (3.69).

The IOALMS algorithm in a feedback control scenario

Using the IMC principle discussed in section 2.4.2, the IOALMS algorithm can also be used in a feedback control scenario (see figure 3.13). Here the solution is treated in which the inverse outer factor is regularized. In the feedback control scenario, the filter-update in equation (3.71) then changes to:

$$\check{\mathbf{w}}_o(k+1) = (1 - \mu\rho)\check{\mathbf{w}}_o(k) - \mu\hat{\mathbf{D}}(k-p)\tilde{\mathbf{e}}(k), \quad (3.72)$$

where $\hat{\mathbf{D}}(k-p)$ is a delayed version of $\hat{\mathbf{D}}(k) \in \mathbb{R}^{MKL \times M}$, which is defined similarly to $\mathbf{X}(k)$ in equation (3.8), but using $\hat{\mathbf{d}}(k)$ described by equation (3.28) instead of $\bar{\mathbf{x}}(k)$ (and appropriate dimensions). Thus, the complete adaptive controller $\check{\mathbf{K}}(k)$ is given by (compare this to equation (3.32)):

$$\check{\mathbf{K}}(k) = \check{\mathbf{S}}_o^{-1} \left\{ \mathbf{I}_K + \check{\mathbf{W}}_o(k)\check{\mathbf{S}}_i \right\}^{-1} \check{\mathbf{W}}_o(k). \quad (3.73)$$

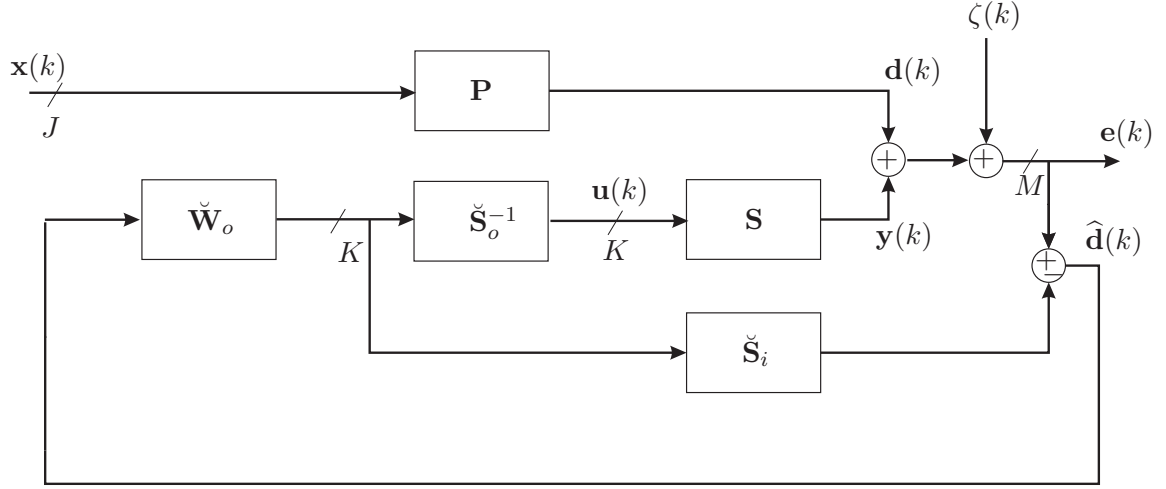


Figure 3.14: Block diagram of a MIMO feedback SD-AVIC system with a regularized fixed gain controller and a regularized inverse outer factor model.

The regularized fixed gain controller from the error sensor outputs $\mathbf{e}(k)$ to the actuator inputs $\mathbf{u}(k)$ is given by:

$$\check{\mathbf{K}} = \check{\mathbf{S}}_o^{-1} \left\{ \mathbf{I}_K + \check{\mathbf{W}}_o \check{\mathbf{S}}_i \right\}^{-1} \check{\mathbf{W}}_o, \quad (3.74)$$

where the coefficients of $\check{\mathbf{W}}_o$ can be determined using equation (3.33). See figure 3.14 for a depiction of the fixed gain control scheme (which is derived from figure 3.13).

3.5 Principle component least mean square algorithm

In this section it is assumed that the reference input $x(k)$ is scalar and not J -dimensional. This assumption has to be made in the derivation of the PCLMS algorithm.

The FxLMS algorithm can be used for driving an arbitrary number of controller coefficients, L . Very often though, the disturbance outputs $\mathbf{d}(k)$ that need to be suppressed only contain a single sine of ω_1 rad/s. To cancel such a disturbance signal, the FxLMS algorithm can be used with an adaptive controller that has a length of two coefficients¹⁵ ($L = 2$). However, the FxLMS algorithm has the shortcoming that the convergence speed of the controller parameters $\mathbf{w}(k)$ is dependent on the eigenvalue spread $\lambda_{max}/\lambda_{min}$. The eigenvalue spread can be large due to the influence of the secondary path \mathbf{S} . In contrast to the FxLMS algorithm, the PCLMS algorithm decouples the secondary path \mathbf{S} at the disturbance frequency ω_1 . Decoupling enables faster convergence of the adaptive controller coefficients.

For the derivation of the PCLMS algorithm, a switch is made from the time domain to the frequency domain. Implementation of the PCLMS algorithm is done in the time domain, though. The *complex* error sensor outputs at sample k , i.e. $\mathbf{e}(k, e^{j\omega_1 T})$, can be written as (Elliott, 2000b):

$$\mathbf{e}(k, e^{j\omega_1 T}) = \mathbf{d}(e^{j\omega_1 T}) + \mathbf{S}(e^{j\omega_1 T}) \mathbf{W}(k, e^{j\omega_1 T}) x(e^{j\omega_1 T}), \quad (3.75)$$

¹⁵Two coefficients are sufficient since for cancelling a single sine only a phase and an amplitude need to be matched.

where $\mathbf{e}(k, e^{j\omega_1 T}) \in \mathbb{C}^M$, $\mathbf{d}(e^{j\omega_1 T}) \in \mathbb{C}^M$, $x(e^{j\omega_1 T}) \in \mathbb{C}^K$ and finally the secondary path $\mathbf{S}(e^{j\omega_1 T}) \in \mathbb{C}^{M \times K}$, provide the amplitude and phase information of the error sensor outputs $\mathbf{e}(k)$, the disturbance outputs \mathbf{d} , the reference input x , the controller $\mathbf{W}(k)$ and the secondary path \mathbf{S} at normalized frequency $\omega_1 T$. The disturbance outputs $\mathbf{d}(e^{j\omega_1 T})$ and the reference input $x(e^{j\omega_1 T})$ do not depend on the sample number 'k' since it is assumed that they contain a stationary single frequency and thus are constant in the frequency domain. If a singular value decomposition of $\mathbf{S}(e^{j\omega_1 T})$ is computed, this gives (Golub and Loan, 1996):

$$\mathbf{S}(e^{j\omega_1 T}) = \mathbf{U}(e^{j\omega_1 T}) \mathbf{\Sigma}(e^{j\omega_1 T}) \mathbf{V}^H(e^{j\omega_1 T}), \quad (3.76)$$

with $(\cdot)^H$ the Hermitian transpose, $\mathbf{U}(e^{j\omega_1 T}) \in \mathbb{C}^{M \times M}$, $\mathbf{\Sigma}(e^{j\omega_1 T}) \in \mathbb{R}^{M \times K}$ and $\mathbf{V}(e^{j\omega_1 T}) \in \mathbb{C}^{K \times K}$. This solution of the singular value decomposition¹⁶ can be substituted in equation (3.75) to obtain (assuming $x = 1$):

$$\mathbf{e}(k, e^{j\omega_1 T}) = \mathbf{d}(e^{j\omega_1 T}) + \mathbf{U}(e^{j\omega_1 T}) \mathbf{\Sigma}(e^{j\omega_1 T}) \mathbf{V}^H(e^{j\omega_1 T}) \mathbf{W}(k, e^{j\omega_1 T}). \quad (3.78)$$

Pre-multiplying the left hand and right hand side of equation (3.78) with $\mathbf{U}^H(e^{j\omega_1 T})$ gives¹⁷:

$$\begin{aligned} \mathbf{U}^H(e^{j\omega_1 T}) \mathbf{e}(k, e^{j\omega_1 T}) &= \mathbf{U}^H(e^{j\omega_1 T}) \mathbf{d}(e^{j\omega_1 T}) \\ &+ \mathbf{U}^H(e^{j\omega_1 T}) \mathbf{U}(e^{j\omega_1 T}) \mathbf{\Sigma}(e^{j\omega_1 T}) \mathbf{V}^H(e^{j\omega_1 T}) \mathbf{W}(k, e^{j\omega_1 T}) \\ &= \mathbf{U}^H(e^{j\omega_1 T}) \mathbf{d}(e^{j\omega_1 T}) + \mathbf{\Sigma}(e^{j\omega_1 T}) \mathbf{V}^H(e^{j\omega_1 T}) \mathbf{W}(k, e^{j\omega_1 T}). \end{aligned} \quad (3.79)$$

The transition from the first to the second expression given in equation (3.79) is true since

$$\mathbf{U}^H(e^{j\omega_1 T}) \mathbf{U}(e^{j\omega_1 T}) = \mathbf{I}_M.$$

The last expression given in equation (3.79) can be simplified to:

$$\tilde{\mathbf{e}}(k, e^{j\omega_1 T}) = \tilde{\mathbf{d}}(e^{j\omega_1 T}) + \mathbf{\Sigma} \tilde{\mathbf{W}}(k, e^{j\omega_1 T}). \quad (3.80)$$

where $\tilde{\mathbf{e}}(k, e^{j\omega_1 T}) = \mathbf{U}^H(e^{j\omega_1 T}) \mathbf{e}(k, e^{j\omega_1 T})$, $\tilde{\mathbf{d}}(e^{j\omega_1 T}) = \mathbf{U}^H(e^{j\omega_1 T}) \mathbf{d}(e^{j\omega_1 T})$ and the controller $\tilde{\mathbf{W}}(k, e^{j\omega_1 T}) = \mathbf{V}^H(e^{j\omega_1 T}) \mathbf{W}(k, e^{j\omega_1 T})$. Equation (3.80) forms the basis of the PCLMS algorithm. From this equation it is clear that a decoupled control problem is created, since $\mathbf{\Sigma}(e^{j\omega_1 T})$ is a diagonal matrix containing the singular values σ_l of $\mathbf{S}(e^{j\omega_1 T})$ only, i.e. there are no cross-couplings present in the matrix $\mathbf{\Sigma}(e^{j\omega_1 T})$ as in the case of $\mathbf{S}(e^{j\omega_1 T})$. Rather than determining the controller $\mathbf{W}(k, e^{j\omega_1 T})$, as the FxLMS algorithm does, the PCLMS algorithm determines the decoupled controller $\tilde{\mathbf{W}}(k, e^{j\omega_1 T})$ given in equation (3.80)¹⁸. In order

¹⁶Note that the eigenvalue-spread for a tonal reference input is defined as:

$$\frac{\sigma_{max}^2}{\sigma_{min}^2} \quad (3.77)$$

with σ_{max} and σ_{min} the maximum and minimum element on the diagonal of the matrix $\mathbf{\Sigma}(e^{j\omega_1 T})$ in equation (3.76), respectively. This eigenvalue-spread is a frequency domain version of the eigenvalue-spread indicated in equation (3.49), but gives exactly the same outcome (see section 4.3.3 in (Elliott, 2000b)).

¹⁷In general, active vibration isolation control problems with $M > K$, $\mathbf{U}^H(e^{j\omega_1 T})$ can be replaced by $\mathbf{U}^H(e^{j\omega_1 T})$, with $\mathbf{U}^H(e^{j\omega_1 T})$ containing the first K rows of $\mathbf{U}^H(e^{j\omega_1 T})$ only (Cabell, 1998).

¹⁸In the time domain, the controller $\tilde{\mathbf{W}}(k, e^{j\omega_1 T})$ is referred to as $\tilde{\mathbf{W}}(k, q^{-1})$.

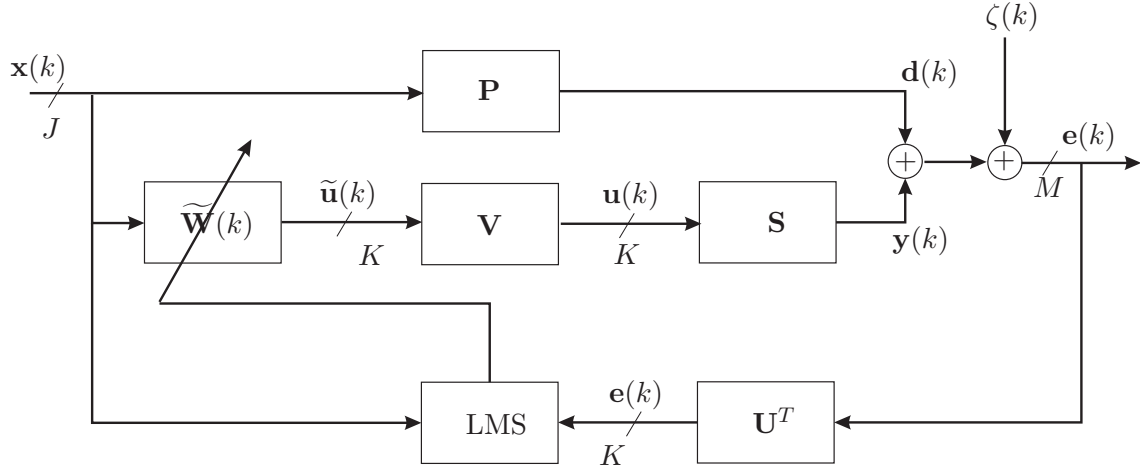


Figure 3.15: Block diagram of the single reference PCLMS control problem.

to implement the PCLMS algorithm it is, however, necessary to filter the error sensor outputs $\mathbf{e}(k)$ with $\mathbf{U}^T(q^{-1})$ and the controller output $\tilde{\mathbf{W}}(k, q^{-1})$ with $\mathbf{V}(q^{-1})$. The following cost function is minimized in the time domain to find a controller:

$$\check{J}(k) = \mathbf{e}^T(k)\mathbf{e}(k) + \rho\tilde{\mathbf{w}}^T(k)\tilde{\mathbf{w}}(k) \quad (3.81)$$

In figure 3.15 a block diagram is shown of the PCLMS algorithm¹⁹. Given the filtered error sensor outputs $\tilde{\mathbf{e}}(k)$, the filter update for the PCLMS algorithm is described by (Elliott, 2000b; Kuo and Morgan, 1996):

$$\begin{aligned} \check{\mathbf{w}}(k+1) &= (1 - \rho\mu)\check{\mathbf{w}}(k) + \frac{\mu}{2} \frac{\partial \check{J}(k)}{\partial \check{\mathbf{w}}(k)} \\ &= (1 - \rho\mu)\mathbf{w}(k) + \mu\mathbf{X}(k)\tilde{\mathbf{e}}(k). \end{aligned} \quad (3.82)$$

The main difference between implementing the PCLMS algorithm or the FxLMS algorithm is that the reference input $x(k)$ does not need to be filtered with \mathbf{S}^T . Instead, the filters \mathbf{U}^T and \mathbf{V} must be implemented that decouple the secondary path \mathbf{S} .

¹⁹With regards to figure 3.15: to create the filters \mathbf{U}^T and \mathbf{V} in the time domain, every individual element of the complex matrices \mathbf{U}^H and \mathbf{V} needs to be transformed to, for example, a FIRF of two coefficients.

Chapter 4

Subspace model identification of the transfer paths of the experimental vibration isolation setups

4.1 Introduction

The contribution of this chapter is the identification of the primary and secondary path of both the SR-SISO vibration isolation setup and the SR-MIMO vibration isolation setup. See section 1.4 for a description of the setups. A block diagram of the transfer paths is depicted in figure 2.17.

As described in section 2.5.1, for this identification the PO-MOESP SMI routine is used. SMI allows one to estimate a model based on measured input/output data from the actuator inputs to the error sensor outputs in a straightforward manner. In appendix C a basic example is given as to how SMI can be applied using the toolbox (NICONET, 2000), that implements the PO-MOESP SMI routine.

This chapter is organized as follows. Section 4.2 treats the identification of both the primary path \mathbf{P} and the secondary path \mathbf{S} of the SR-SISO vibration isolation setup. Section 4.3 discusses the identification of both the primary path \mathbf{P} and the secondary path \mathbf{S} of the SR-MIMO vibration isolation setup.

4.2 Identification of the transfer paths of the SR-SISO vibration isolation setup

In this section the system χ (see equation (2.15), with $J = K = M = 1$) of the SR-SISO vibration isolation setup is identified. The system χ comprises both the primary path \mathbf{P} and the secondary path \mathbf{S} . Thus, this section describes multiple¹ input single output (MISO) identification experiments. Section 4.2.1 describes what data are used for the identification experiment, based on the guidelines set out in section 2.5.3. In section 4.2.2 the actual identification is described.

¹Two, to be exact.

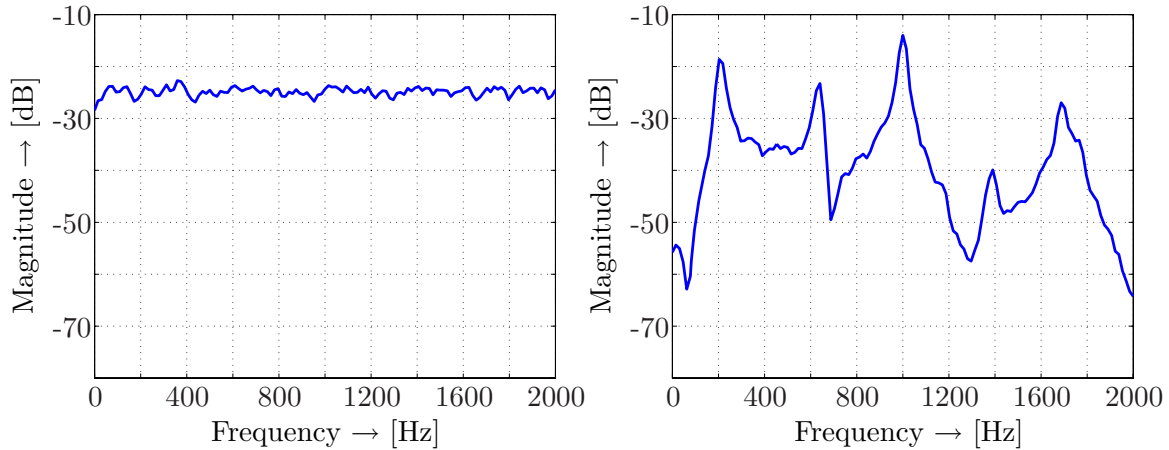


Figure 4.1: Spectrum of the actuator input (left) and the error sensor output (right).

4.2.1 Acquisition of a proper data set for identification

A data set of 16,000 samples (four seconds of data sampling at a frequency of 4 kHz) is obtained from the SR-SISO vibration isolation setup. The piezo-electric actuator and the shaker actuator are provided (or excited, as it is often referred to) with a stochastic white noise sequence and the error sensor output $e(k)$ is recorded. The data set is split in half to obtain a trial data set and a validation data set. Thus, there are two data sets of 2 inputs \times 1 output \times 8,000 samples.

4.2.2 Identification of the primary path and the secondary path

The input/output data are proper in the sense that they do not show undesired discrepancies such as spikes. The spectrum of the actuator input data² and the error sensor output data are shown in figure 4.1. The spectrum of the actuator input data is white as desired. The spectrum of the error sensor output data clearly shows the four main resonance peaks of the system, i.e. at 203 Hz, 640 Hz, 1 kHz, and 1688 Hz.

The presence of delays in the system χ is checked. It is desirable to remove delays from the data before identification takes place, to reduce the model order. The presence of delays is checked by estimating two finite impulse response models on the data, i.e. from every input to the error sensor output. This check shows that the system χ does not contain delays. A final action that is performed is removing a sample mean and a linear trend from the input/output data (Ljung and Glad, 1994). Subsequently, the actual identification is performed. The number of rows³ s in the estimated extended observability matrix is chosen to be 50 (Verhaegen, 1994). The first 40 singular values of the estimated extended observability matrix of the system χ are depicted in figure 4.2. The singular value plot indicates - by displaying gaps - that model orders 2, 4, 6, 8 and 28 might be appropriate. The model order

²The shaker actuator and the piezo-electric actuator are excited with a similar white noise sequence, i.e. the noise sequences have the same mean and variance. Therefore, only *one* representation of the input data is shown.

³The number of (block) rows s in the Hankel matrix is equivalent to the upper limit (minus one) of the order of the system that can be identified (Verhaegen, 1994).

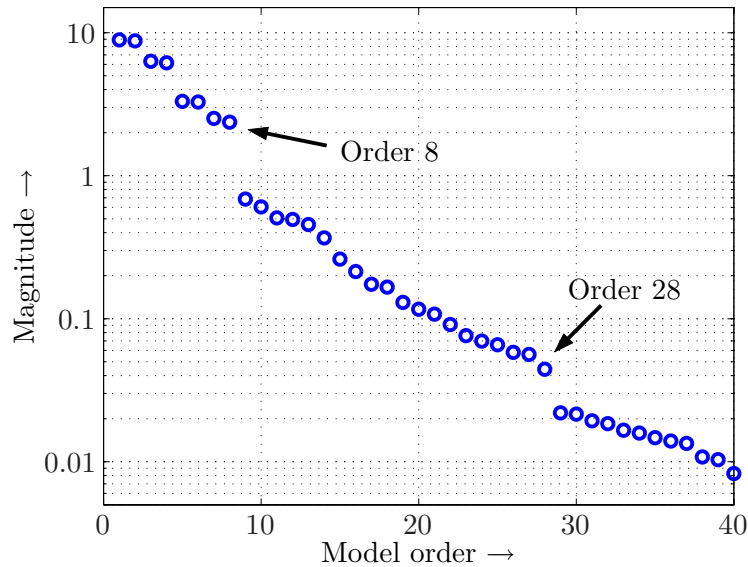


Figure 4.2: Identification of the system χ in equation (2.15): singular values of the estimated extended observability matrix.

is the dimension N of $\mathbf{z}(k)$ in equation (2.15). An initial ‘guess’ is model order 8, which gives a VAF value (equation (2.17)) of 94.8% on the validation data, i.e. an SNR value (equation (2.18)) of 12.9 dB. From the low VAF-value it is concluded that the accuracy of that particular model order is not sufficient. Therefore, a model is estimated with order 28, resulting in a VAF value of 99.7% on the validation data, i.e. an SNR value of 24.5 dB. These large numbers indicate that the system can very well be described by a linear model. To be on the safe side and to gain more insight concerning the proper model order, the VAF values were calculated for model orders⁴ 15-38. The results are shown in figure 4.3. Clearly, a model of order 28 is a good choice. The VAF values ‘saturate’ for this particular model order at the previously mentioned value of 99.7 %. In figure 4.4 the result of the identification procedure is depicted by means of spectra on the validation data output and the estimated validation data output. It can be concluded that the spectra accurately match. The lines overlap and no differences are visible. The peaks that are visible in the residual are largely due to measurement noise which is present in the background. In figure 4.5 a pole/zero map is shown of the estimated secondary path model $\hat{\mathbf{S}}$ (the pole/zero map of the estimated primary path model $\hat{\mathbf{P}}$ is omitted for brevity). In the pole/zero map of $\hat{\mathbf{S}}$, two zeros on the real axis are outside the axis of the plot: a zero at -3.69 and a zero at -51.88. These two zeros are not shown to avoid confusion. From the pole-zero map it can be deduced that the model $\hat{\mathbf{S}}$ is non-minimum phase, since there are zeros outside the unit circle. Also, the model $\hat{\mathbf{S}}$ has mostly undamped poles and zeros, indicating resonant behavior. Finally, in figure 4.6 a Bode diagram is presented of the estimated secondary path $\hat{\mathbf{S}}$. The overlapping spectra in figure 4.4 and the high VAF-value of 99.7% on the validation data lead to the conclusion that an accurate linear model has been found in a straightforward manner. Note that a model of order 28 seems significant for a

⁴The VAF values for model orders 1-14, 39 and 40 are not presented. Rather the close-up version of the VAF plot is given. These VAF values are omitted to obtain a good view of the VAF plot. For example, models with orders 1, 39 and 40 are unstable and resulted in a VAF value of 0%.

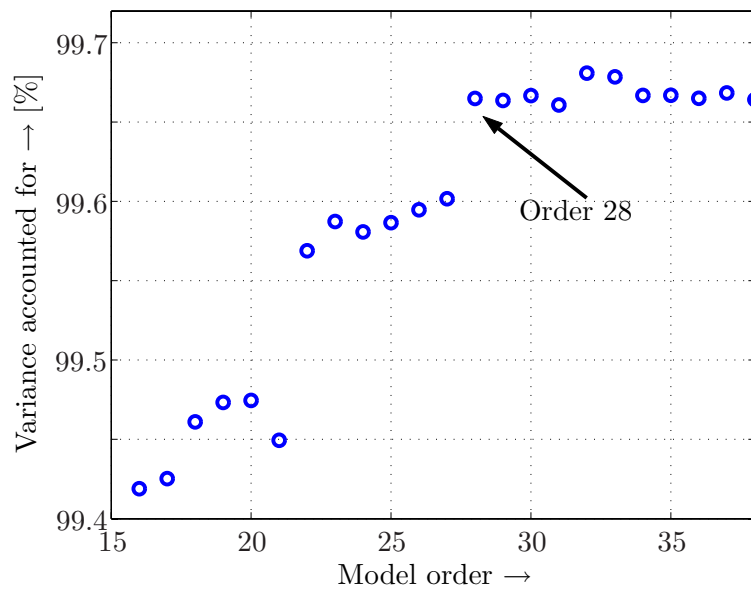


Figure 4.3: Identification of the system χ in equation (2.15): VAF values.

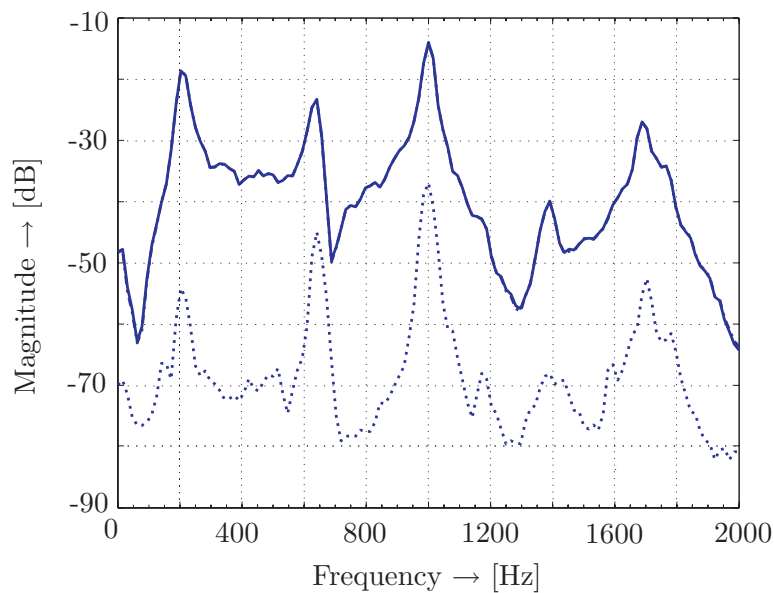


Figure 4.4: Spectrum of the validation data output (solid, [—]), the estimated validation data output (dashed, [- -]) (not visible) and the residual (dotted, [· · ·]). The solid and the dashed lines overlap, and therefore no differences are visible.

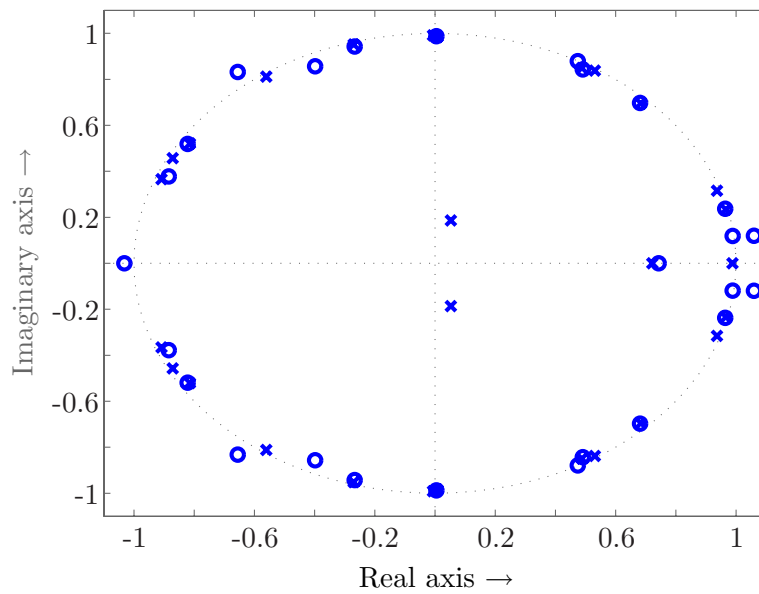


Figure 4.5: Pole/zero map of the estimated secondary path \hat{S} .

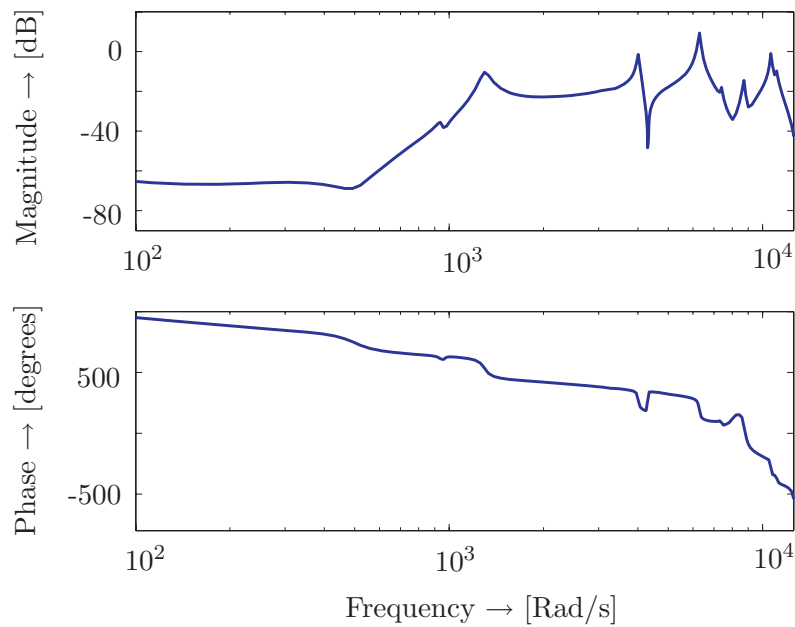


Figure 4.6: Bode diagram of the estimated secondary path \hat{S} .

‘simple’ system consisting of a shaker actuator and a piezo-electric actuator. However, the identified model not only comprises the dynamics of the shaker actuator and the piezo-electric actuator itself, but also includes amplifiers, reconstruction filters, anti-aliasing filters and so on. Moreover, the clamped metal plate (see figure 1.7) consists of a number of modes that are excited in the frequency band between 0 and 2 kHz. That is also considered in this model.

4.3 Identification of the transfer paths of the SR-MIMO vibration isolation setup

First, the primary path \mathbf{P} is identified. Subsequently, the secondary path \mathbf{S} is identified. Now $J = 1$ and $K = M = 6$ for the system χ in equation (2.15).

This section is organized as follows. Section 4.3.1 describes what data sets are obtained for the identification tasks. In section 4.3.2 the SIMO identification experiment of the primary path \mathbf{P} is discussed. In section 4.3.3 the MIMO identification of the secondary path \mathbf{S} is described⁵.

4.3.1 Acquisition of proper data sets for identification

Two data sets are obtained from the SR-MIMO vibration isolation setup. First, the reference input $x(k)$ is excited with a stochastic white noise sequence and the error sensor outputs $\mathbf{e}(k)$ are recorded. The actuator inputs $\mathbf{u}(k)$ are not excited in this first experiment. Second, the actuator inputs $\mathbf{u}(k)$ are excited with a stochastic white noise sequence and the error sensor outputs $\mathbf{e}(k)$ are recorded. In this second experiment, the reference input $x(k)$ is not excited. Based on these two separate data sets the primary path \mathbf{P} and the secondary path \mathbf{S} are identified (Nijssen et al., 2004). For both identification experiments ten seconds of data, i.e. 20,000 samples (ten seconds of data sampling at a frequency of 2 kHz), are generated. Thus, one data set is obtained of 1 inputs \times 6 outputs \times 20,000 samples from the shaker actuator input to the six error sensor outputs meant for identification of the primary path \mathbf{P} . Furthermore, one data set is obtained of 6 input \times 6 outputs \times 20,000 samples from the six actuator inputs to the six error sensor outputs meant for identification of the secondary path \mathbf{S} . Six seconds of data are used to test the models, while four seconds of data are used to validate the models. This gives rise to four data sets. From here on it is assumed that the sample mean and the linear trend have been removed from the data.

4.3.2 Identification of the primary path

Studying the input/output data, it can be concluded that these do not include spikes or other undesired discrepancies. The spectra of the shaker actuator input data and the first error sensor output data are shown in figure 4.7. The other five error sensor outputs are not depicted for reasons of conciseness. The figure shows that the spectrum of the shaker actuator input data is white, as desired. In the spectrum of the first error sensor output data a significant number of resonance peaks can be seen. Therefore, a high number of block rows s is chosen: $s = 140$. It is also checked whether the primary path \mathbf{P} contains delays by estimating six impulse responses on the data, i.e. by estimating an impulse response from

⁵Note that the secondary path \mathbf{S} is identified from all six inputs to all six outputs all at once. This is a benefit of SMI, which allows for identification of MIMO plants in one time.

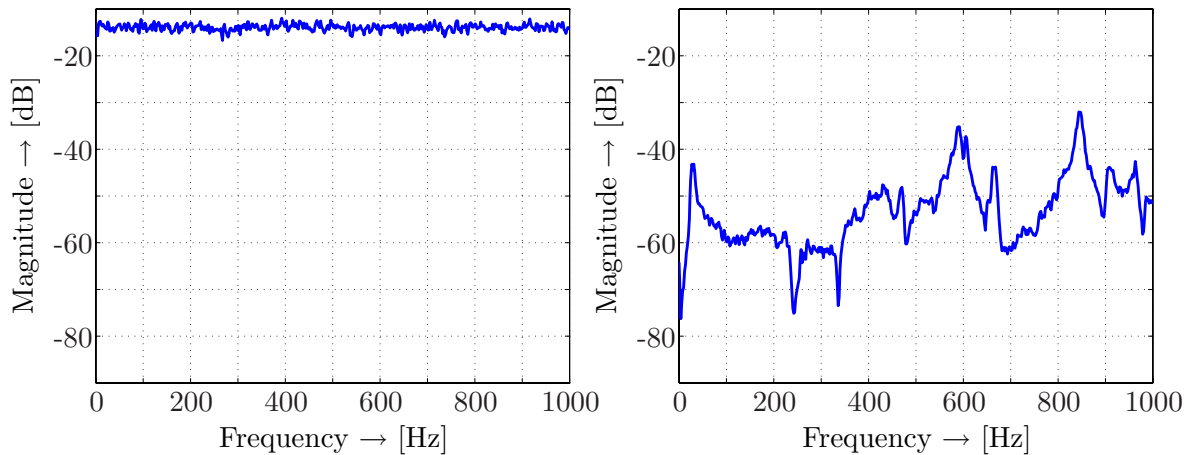


Figure 4.7: Spectrum of the shaker actuator input data (left) and the first error sensor output data (right).

the shaker actuator input to every error sensor output. Since all impulse responses have a direct feed-through term no delays need to be removed and it is concluded that the data are suitable for identification. Figure 4.8 shows a close-up version of the 840 singular values ($840 = 6 \text{ outputs} \times 140 \text{ block rows}$). The plot with the singular values indicates that several model orders might be appropriate, e.g. model orders 4, 14, 24 and 32. A model with order 32 gives a VAF of 97.8% (a SNR of 16.9 dB) on the validation data. Although these numbers seem to be high, it is concluded that a model with that order does not provide sufficient accuracy. That conclusion is drawn after careful analysis of the spectra on the validation data output and estimated validation data output. However, at higher values than model order 32 it is difficult to draw a good conclusion about the model order from the singular values. This is due to conflicts with the basic assumptions made in section 2.5.1, i.e. assumptions on linearity, process and measurement noise (Visser et al., 1996; Verhaegen et al., 1994). Note that even with $s = 300$ and more input/output data, the desired model order information, i.e. a significant gap in the singular values, is not visible. Though there are several gaps in the plot, it does not reveal the order information as clearly as for the system χ for the SR-SISO vibration isolation setup. However, for complex real-life systems this is not uncommon. Although the singular value plot can be a handy tool, the choice of the model order from the singular value plot is heuristic (van Overschee and de Moor, 1996). To obtain a better insight in the model order, figure 4.9 gives the VAF values ranging from model orders 60 to 105. An important conclusion can be drawn from the figure: the VAF values continuously increase up till model order 105. However:

1. although the VAF values do not saturate, they increase slowly. For example, the difference between selecting a model with order 100 and one with order 105 is only 0.03%.
2. for the development of real-time adaptive controllers, the primary path \mathbf{P} is not needed at all.
3. most models with orders higher than 106 are unstable models, i.e. they have pole(s) outside the unit circle. This can be explained by the method of SMI which does not

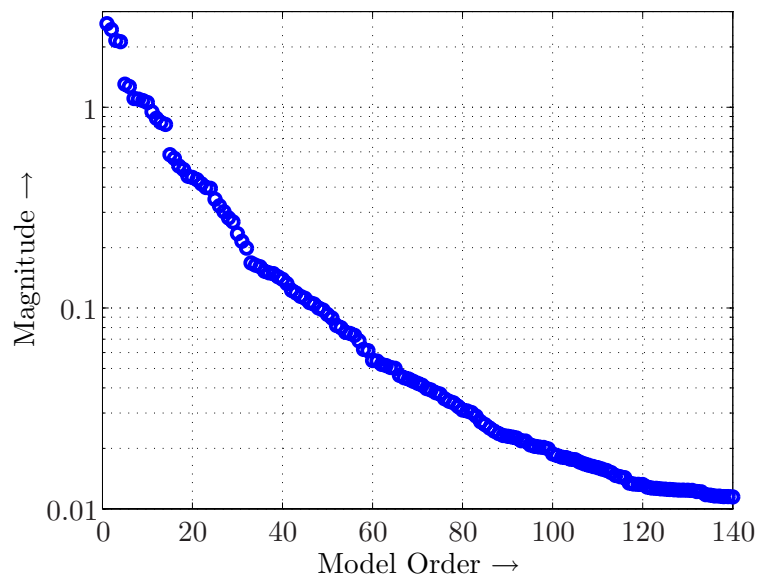


Figure 4.8: Singular values stemming from the identification of the primary path \mathbf{P} of the SR-MIMO vibration isolation setup.

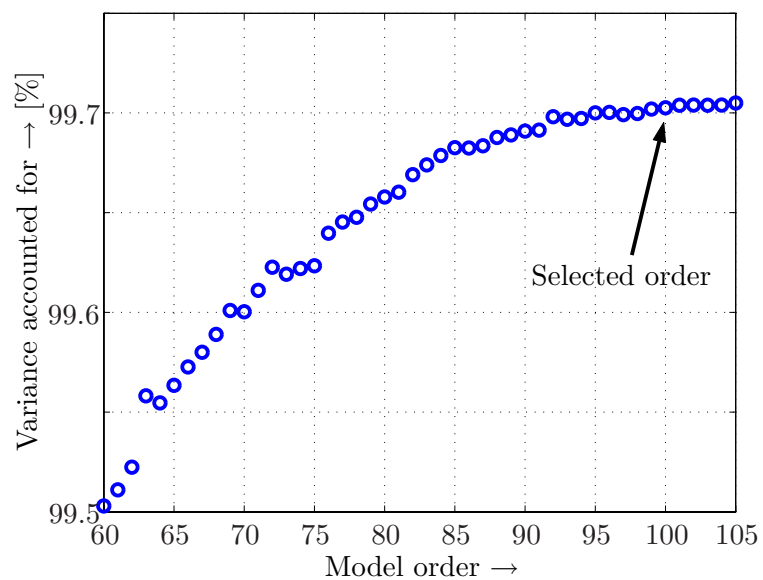
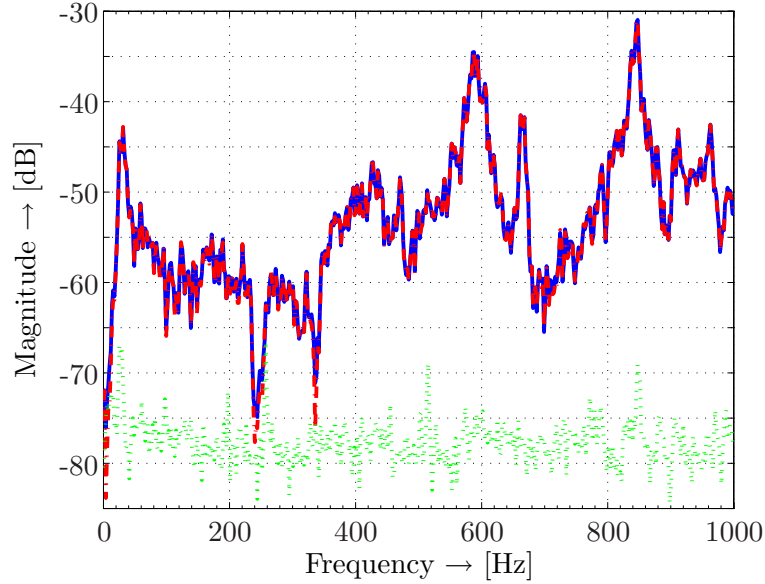


Figure 4.9: VAF values stemming from the identification of the primary path \mathbf{P} of the SR-MIMO vibration isolation setup.

Table 4.1: VAF values and SNR values stemming from the identification of the primary \mathbf{P} of the SR-MIMO vibration isolation setup.

Output	1	2	3	4	5	6	Average
VAF (%)	99.9	99.9	99.6	99.5	99.6	99.7	99.7
SNR (dB)	30.1	30.3	24.2	22.9	24.1	25.0	26.1

**Figure 4.10:** Spectra of the first validation data output (solid, [—]), the estimated first validation data output (dash-dotted, [· · ·]) and the residual (dotted, [· · ·]). The spectra overlap, so no differences are visible.

guarantee a stable model: the system matrix \mathbf{A} is computed by a QR-decomposition and a singular value decomposition (Golub and Loan, 1996; Strang, 1988). The compositions do not constrain the pole locations to be within the unit circle. Routines can be applied to ‘stabilize’ the model, but that is not necessary for our application (van Overschee and de Moor, 1996).

For these reasons, the goal here is not to get the best linear model possible. A model of order 100 is chosen, which gives rise to a VAF value of 99.7% on the validation data, i.e. an SNR value of 26.1 dB. The six separate VAF and SNR values on which these averages of 99.7% and 26.1 dB are based, are shown in table 4.1. To further demonstrate the accuracy of the model, figure 4.10 shows the spectra of the first validation data output, the estimated first validation data outputs and the corresponding residual. The spectra for the other five sensors are omitted, since they show a similar behavior. From the spectra it can be concluded that an accurate fit is obtained over the complete frequency band. Some anti-resonances do not fit very well, which can be explained by the fact that the magnitude of the anti-resonance is of the same order as the magnitude of the measurement noise level. It is concluded that an accurate linear model can be identified between the shaker actuator input and the six

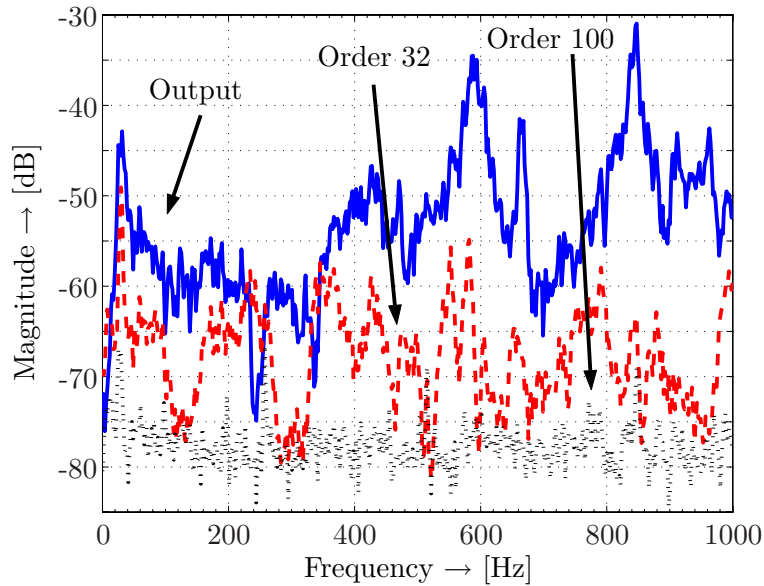


Figure 4.11: Spectrum of the first validation data output (solid, $[-]$), the residual for a model with order 32 (dashed, $[- -]$) and the residual for a model with order 100 (dotted, $[\cdot\cdot\cdot]$).

error sensor outputs. For reasons of comparison, in figure 4.11 the fit on the first error sensor output is given on the validation data for a model with order 32 and for the selected model of order 100. Comparing the fit of the 32nd order model on the validation data output with the fit of the 100th order model, leads to the conclusion that the 100th order model is indeed much better.

4.3.3 Identification of the secondary path

For reasons of conciseness, the investigation of the obtained input/output data is not described in further detail (note: no delays needed to be removed from the data). Instead the focus is put on the actual identification with the number of block rows $s = 140$. In figure 4.12 a close-up version of the $6 \times 140 = 840$ singular values is shown. The singular value plot clearly indicates that several model orders might be appropriate, e.g. model orders 2, 4, 62, 72 and (approximately) 84. Here, a model of order 84 is chosen, which gives a VAF of 99.8% (26.6 dB) on the validation data. Since this is a high value, a model with order 84 seems to be a logical choice. However, as with the identification of the primary path \mathbf{P} it is difficult to draw a concrete conclusion about the model order from the singular values. Figure 4.13 shows the VAF values on the validation data output and the estimated validation data output, for model orders 60-105. From this figure it is clear that a model with order 84 is a good choice, since the VAF values approximately saturate for that model order. Note that the VAF values keep on increasing slowly, even for model orders higher than 130, but that this increase is relatively small. This is to be expected, since it also happened with the identification of the primary path \mathbf{P} . The six separate VAF values and SNR values concerning the model with order 84 are given in table 4.2. Figure 4.14 depicts the fit of the model on the validation data output for the first sensor. At frequencies below approximately 150 Hz, the fit of the model is not good. This is due to:

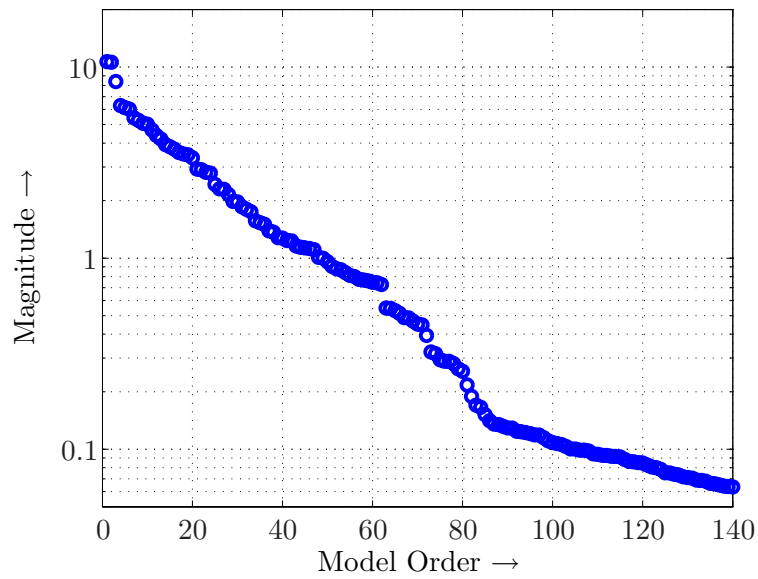


Figure 4.12: Singular values stemming from the identification of the secondary path \mathbf{S} of the SR-MIMO vibration isolation setup.

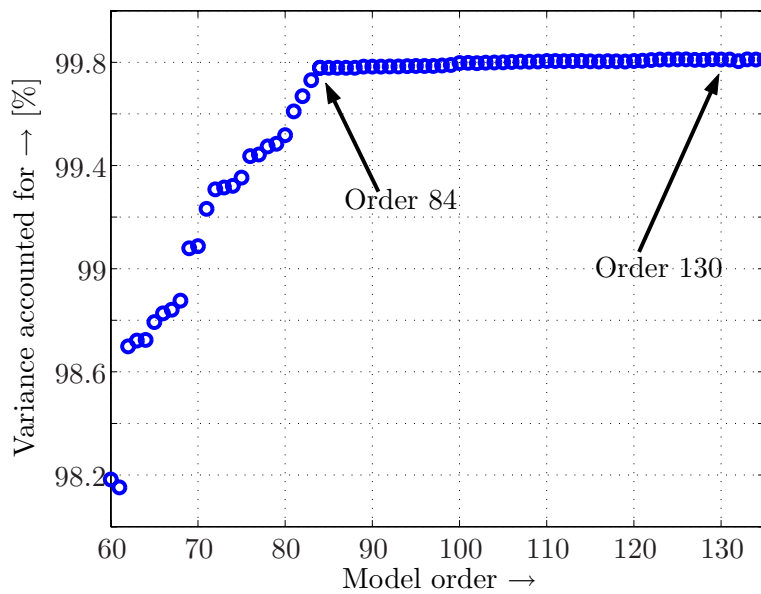


Figure 4.13: VAF values stemming from the identification of the secondary path \mathbf{S} of the SR-MIMO vibration isolation setup.

Table 4.2: VAF values and SNR values stemming from the identification of the secondary path \mathbf{S} of the SR-MIMO vibration isolation setup (model order 84) .

Output	1	2	3	4	5	6	Average
VAF (%)	99.8	99.8	99.8	99.8	99.8	99.8	99.8
SNR (dB)	26.7	26.0	26.4	27.0	26.4	26.8	26.6

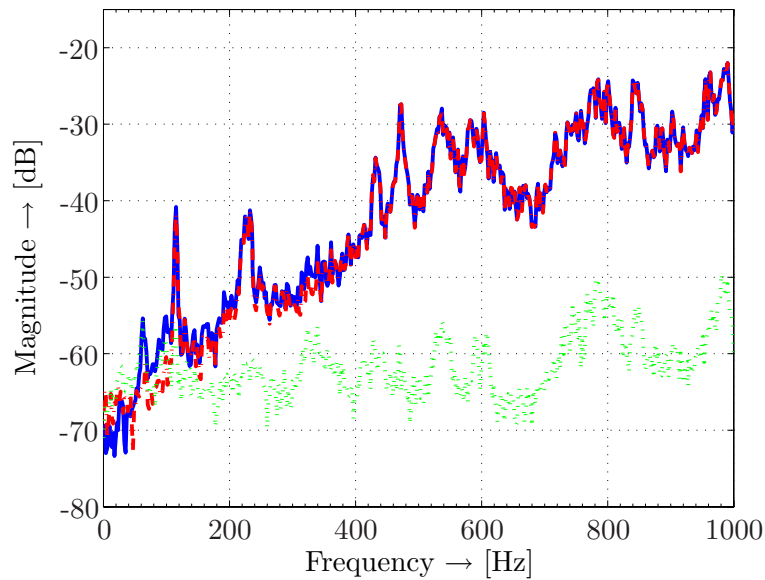


Figure 4.14: Spectra of the first validation data output (solid, $[-]$), the estimated first validation data output (dash-dotted, $[-\cdot-]$) and the residual (dotted, $[\cdot\cdot\cdot]$) with a model with order 84. The spectra mainly overlap, so no differences are visible.

Table 4.3: VAF values and SNR values stemming from the identification of the secondary path \mathbf{S} of the SR-MIMO vibration isolation setup (model order 130).

Output	1	2	3	4	5	6	Average
VAF (%)	99.8	99.8	99.8	99.8	99.8	99.8	99.8
SNR (dB)	27.3	26.4	27.2	27.6	27.4	27.5	27.2

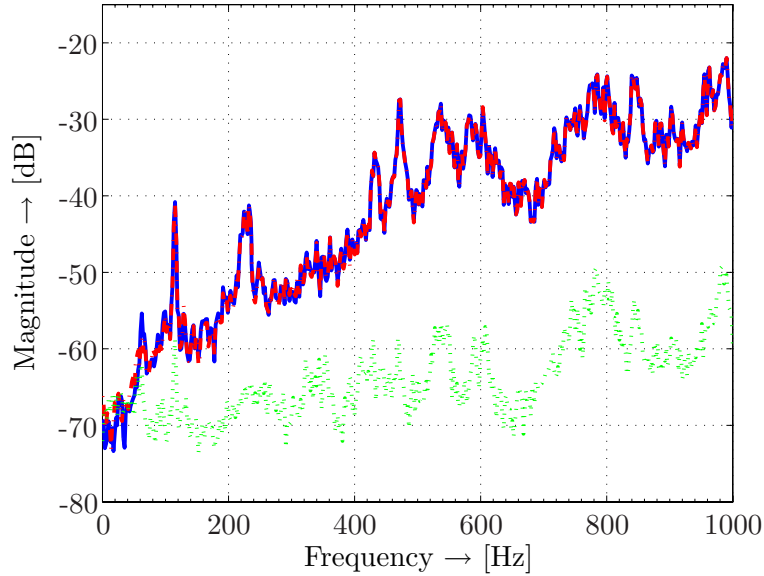


Figure 4.15: Spectra of the first validation data output (solid, $[-]$), the estimated first validation data output (dash-dotted, $[-\cdot-]$) and the residual (dotted, $[\cdot\cdot\cdot]$) with a model with order 130. The spectra mainly overlap, so no differences are visible.

- the properties of the setup (Super, 2006): the secondary path \mathbf{S} has a low gain for low frequencies. Due to this low gain, there is a low signal-to-noise ratio, hampering the identification procedure.
- a model with order 84 is a compromise between the accuracy and the number of parameters used in the model. Selecting a higher model order will improve the fit at lower frequencies. This is demonstrated hereafter for a model with order 130.

For reasons of comparison, table 4.3 shows the VAF and SNR values for a model with order 130. From these values it can be concluded that a model with order 130 is more accurate than a model with order 84 (this is especially clear from the SNR values). In figure 4.15 the spectra corresponding to the first (estimated) validation data output is shown. The improved fit in the lower frequency area is obvious. It is concluded that the secondary path of the SR-MIMO vibration isolation setup can be accurately modelled by a linear model.

Chapter 5

Control of narrowband disturbances on the SR-MIMO vibration isolation set-up

5.1 Introduction

Taking the identified models as a starting point, this chapter describes narrowband adaptive feedforward control experiments that are performed in simulation and real-time on the SR-MIMO vibration isolation set-up. To be precise, in the control experiments, narrowband disturbances stemming from the source structure are isolated from the receiver structure by steering the six ceramic piezo-electric actuators with an adaptive controller.

The two main goals of the experiments described in this chapter are:

- to implement and validate two control algorithms for controlling narrowband disturbances. Specifically, this concerns the traditional filtered-x least mean squares (FxLMS) algorithm (see section 3.3) and the principle component least mean squares (PCLMS) algorithm (see section 3.5). The PCLMS algorithm is especially suitable for suppressing narrowband disturbances and - in general - exhibits faster convergence speed for the adaptive controller coefficients than the FxLMS algorithm (due to decoupling of the adaptive controller).
- to provide a comparison between the two algorithms on the SR-MIMO vibration isolation set-up. In particular, single frequency experiments and a multiple frequency experiment are described. In the single frequency experiments it is attempted to eliminate a disturbance output that contains a single frequency, i.e. a single tone. In the multiple frequency experiment it is attempted to eliminate a disturbance output that contains two frequencies, i.e. two tones. The real-time experiments are performed on the dSPACE 1005 system (see section E.2.2 in appendix E).

Note that frequent use of models is made in this chapter, and thus frequent use of the symbol $\widehat{(\cdot)}$.

The chapter is organized as follows. Section 5.2 describes the experiments in which a disturbance output that contains one frequency is suppressed. Section 5.3 describes the experiment in which a disturbance output that contains two frequencies is suppressed.

Table 5.1: Eigenvalue-spreads and step-sizes in the two single frequency experiments for both the FxLMS algorithm and the PCLMS algorithm. The same step-size μ is maintained in the PCLMS and FxLMS algorithm.

Frequency	528 Hz	565 Hz
Eigenvalue-spread	3094	162
Step-size μ	0.10	0.09

5.2 Eliminating a single frequency disturbance output

This section is organized as follows. Section 5.2.2 discusses the first single frequency experiment. Section 5.2.3 discusses the second single frequency experiment. Before describing the experiments, the preliminaries, such as the step-size μ that is chosen in the algorithms and the amount of measurement noise that is added in the simulation, are discussed in section 5.2.1.

5.2.1 Preliminaries concerning the experiments

A block diagram of the adaptive control scheme is shown in figure 3.7 for the FxLMS algorithm and in figure 3.15 for the PCLMS algorithm. The FxLMS algorithm updates the coefficients of the adaptive controller $\mathbf{W}(k)$, while the PCLMS algorithm updates the coefficients of the (decoupled) adaptive controller $\widetilde{\mathbf{W}}(k)$. Both adaptive single input six outputs controllers have a FIRF structure and every channel in the controller contains two coefficients in the FIRF. See also section A.3 in the appendix for an explanation on what having two coefficients in the FIRF entails.

Two single frequency experiments are performed. In the first experiment, the reference input $x(k)$ consists of a single frequency of 528 Hz. Thus, the disturbance output $\mathbf{d}(k)$ also contains a single frequency disturbance of 528 Hz. In the second experiment, the reference input $x(k)$ consists of a single frequency of 565 Hz. These two frequencies are chosen with regard to the eigenvalue-spread in the secondary path \mathbf{S} (see equation (3.77)). At 528 Hz there is a significant eigenvalue-spread. At 565 Hz the eigenvalue-spread is smaller. Table 5.1 shows the actual values of the eigenvalue-spread (the presence of the step-size μ in the table will be explained later). Therefore, a slower convergence of the adaptive controller coefficients is expected when the FxLMS algorithm is used in controlling the 528 Hz disturbance as opposed to the 565 Hz disturbance¹. The PCLMS algorithm on the other hand, has the ability to overcome the slow convergence problem due to the decoupling of the adaptive controller $\widetilde{\mathbf{W}}(k)$ and a more rapid convergence of the adaptive controller coefficients is expected when the PCLMS algorithm is used. For both single frequency experiments, in both algorithms, a step-size μ has to be chosen. See equation (3.39) and equation (3.82) in which the step-size μ is present. After careful investigation, the step-sizes μ are set at the values mentioned in table 5.1.

¹The fact that the convergence is slower in the case of a larger eigenvalue-spread is a well-known result that is described in e.g. (Elliott, 2000b). Specifically, the convergence speed of the adaptive controller parameters is dependent on the eigenvalue-spread Θ present in the secondary path \mathbf{S} at the disturbance frequency under consideration (see section 3.3.2).

In the simulation experiments, the measurement noise $\zeta(k)$ is maintained at approximately 25 dB SNR with respect to the disturbance output $\mathbf{d}(k)$. Take a look at e.g. figure 3.7 to see where the measurement noise $\zeta(k)$ enters the system. In both the simulation experiments and the real-time experiments the adaptive controller is switched on after $t = 10$ seconds, i.e. the adaptation of the adaptive controller coefficients starts after ten seconds. In the simulation and real-time experiments the adaptive controller coefficients are initialized with zeros before the adaptation starts. Every single frequency experiment takes 140 seconds.

5.2.2 Eliminating a disturbance output containing 528 Hz

The FxLMS algorithm and the PCLMS algorithm are implemented in Matlab. The models $\hat{\mathbf{P}}$ and $\hat{\mathbf{S}}$ of the primary path and secondary path, respectively, are used to represent the true plants in the simulation. For the FxLMS algorithm, for filtering the reference signal $x(k)$ in the sense of the Kronecker tensor product as in equation (3.39), a FIRF $\hat{\mathbf{S}}_{\text{FIRF}}$ is extracted from the secondary path model $\hat{\mathbf{S}}$ at 528 Hz². Every channel in the FIRF has two coefficients. The FIRF $\hat{\mathbf{S}}_{\text{FIRF}}$ is derived from the secondary path model $\hat{\mathbf{S}}$ as follows:

1. The complex transfer $\hat{\mathbf{S}}(e^{j2\pi 528T})$ of the secondary path model $\hat{\mathbf{S}}$ at the disturbance frequency of 528 Hz is determined from the estimated SSM. This can easily be done in Matlab by computing the frequency response at that particular frequency. For the six input six output secondary path model, $\hat{\mathbf{S}}(e^{j2\pi 528T})$ is a matrix of 36 complex coefficients, with each coefficient consisting of a real and an imaginary part.
2. Every complex coefficient can be transformed into a two-coefficient FIRF. The procedure is described in appendix D.
3. All 36 FIRFs are collected in a new transfer matrix $\hat{\mathbf{S}}_{\text{FIRF}}$, that gives the desired result.

For the PCLMS algorithm, the transfers $\hat{\mathbf{V}}(e^{j2\pi 528T})$ and $\hat{\mathbf{U}}^H(e^{j2\pi 528T})$ are determined to decouple the adaptive controller $\widetilde{\mathbf{W}}(k)$. That is done as follows. From the complex valued matrix $\hat{\mathbf{S}}(e^{j2\pi 528T})$, that is determined in the first step just described, a SVD is computed as in equation (3.76). This immediately gives the complex matrices $\hat{\mathbf{V}}(e^{j2\pi 528T})$ and $\hat{\mathbf{U}}^H(e^{j2\pi 528T})$, each having 36 coefficients also. These two matrices can be converted to the FIRFs $\hat{\mathbf{V}}(q^{-1})$ and $\hat{\mathbf{U}}^H(q^{-1})$ in a manner equivalent to what is done to determine the two-coefficient secondary path model $\hat{\mathbf{S}}_{\text{FIRF}}$. The reference signal $x(k)$ does not need to be filtered in the case of the PCLMS algorithm, so no other filters need to be implemented.

The results that are obtained are shown in figure 5.1. The ones for the FxLMS algorithm are presented on the left and those for the PCLMS algorithm are shown on the right. The spectra depicted are obtained using the last ten seconds of the experiment. The figure shows the controlled and the uncontrolled spectra that result from averaging the six error sensor outputs. That is, for each error sensor output the spectrum is determined, then all the spectra are summed and divided by six. Clearly, at 528 Hz there is a peak of -13 dB in the uncontrolled outputs. From the results shown in the figure it can be concluded that the peak at 528 Hz is reduced to approximately the noise level for both algorithms, indicating that both adaptive controllers perform well. An average reduction of approximately 61 dB is obtained for both algorithms.

²To replace the full order SSM.

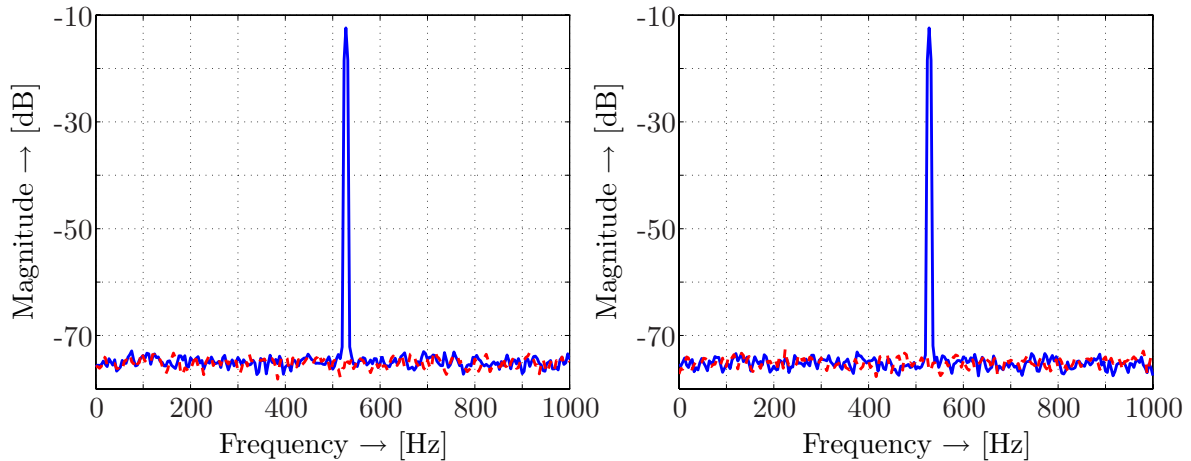


Figure 5.1: Controlling a single frequency disturbance of 528 Hz with the FxLMS algorithm (left) and the PCLMS algorithm (right) in simulation. The solid line [—] represents the uncontrolled output, the dashed line [- -] represents the controlled output.

However, in the case of the FxLMS algorithm it takes more time for the controller coefficients to converge, due to the eigenvalue-spread present in the secondary path \mathbf{S} . To demonstrate the slower convergence, in figure 5.2 the averaged adaptive controller coefficients are given, representing the two-norm difference between the true optimal controller coefficients and the adaptive controller coefficients that are averaged over one hundred experiments. Specifically, one hundred independent simulation runs are done with one hundred different realizations of the measurement noise $\zeta(k)$ and the resulting controller coefficients are averaged. If the adaptive controller coefficients converge perfectly, the solid line coincides with the dotted line. From figure 5.2 it can be concluded that within the 140 second time frame, in the case of the FxLMS algorithm, the adaptive controller coefficients converge after approximately 110 seconds³, at $t=120$ s. In the case of the PCLMS algorithm however, the adaptive controller coefficients are already converged after approximately five seconds, at $t=15$ s. That is due to the decoupling property of the adaptive controller. Note that in figure 5.2 the curve on the left and the one on the right do not have the same magnitude at $t = 0$ s. That has to do with the fact that the FxLMS algorithm adapts the adaptive controller $\mathbf{W}(k)$, while the PCLMS algorithm adapts the adaptive controller $\widetilde{\mathbf{W}}(k)$. These are different adaptive controllers and therefore have different magnitudes at $t = 0$ s.

After these simulation experiments, the adaptive controllers are implemented on the dSPACE 1005 system. The results that are obtained are given in figure 5.3. On the left the results are shown that are obtained with the FxLMS algorithm. The results shown on the right are obtained with the PCLMS algorithm. The figure shows the controlled and uncontrolled spectra that result from averaging all six error sensor outputs. Similar to the simulation results, an average reduction of approximately 61 dB is established for both algorithms. The convergence speed in real-time is equivalent to the convergence speed in the simulation experiment. Also, the simulation results and the real-time results correspond well, which confirms that accurate models are obtained from the SR-MIMO vibration isolation

³In the case of the PCLMS algorithm only 50 seconds are shown for the graph with the averaged adaptive controller coefficients, to make it more visible.

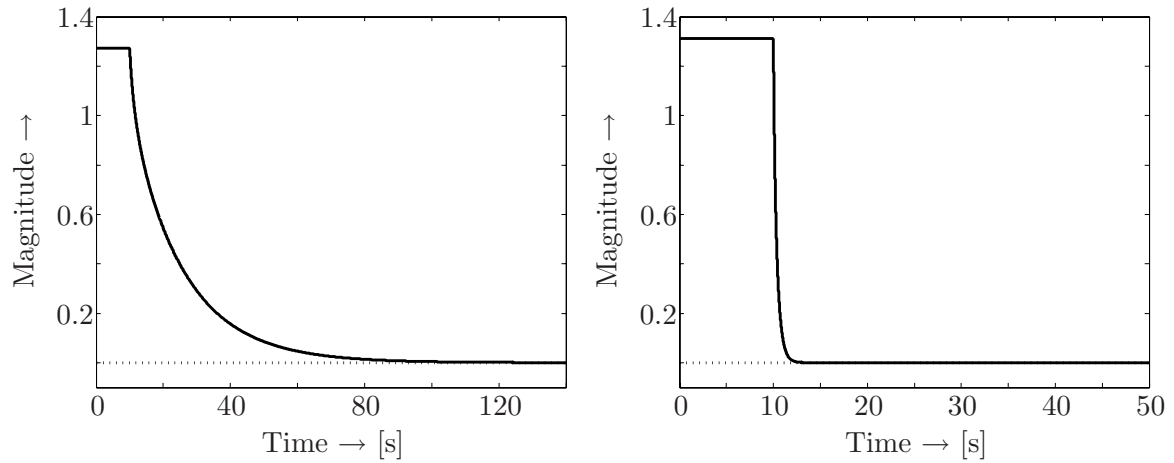


Figure 5.2: Averaged adaptive controller coefficients when eliminating the 528 Hz disturbance in a simulation with the FxLMS algorithm (left) and the PCLMS algorithm (right).

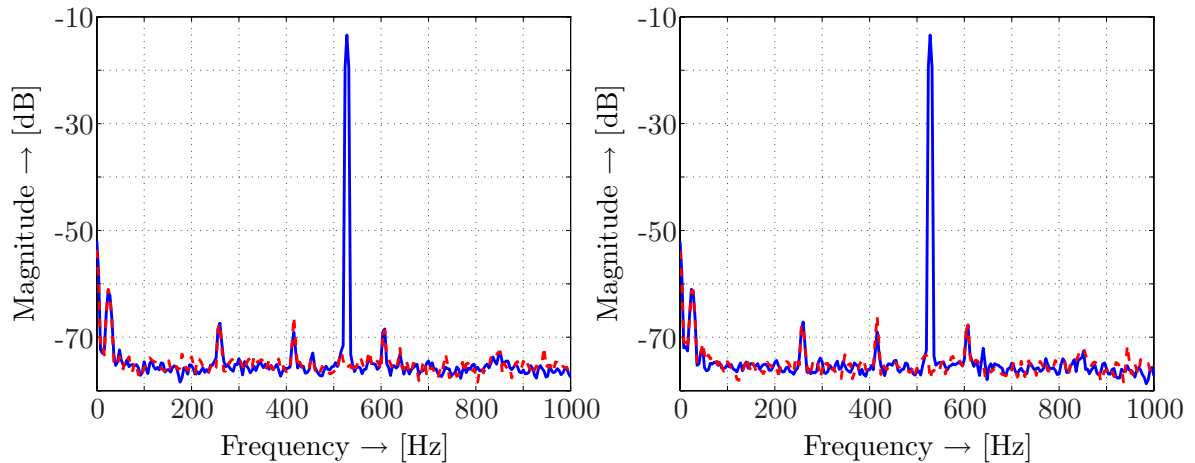


Figure 5.3: Controlling a single frequency disturbance of 528 Hz with the FxLMS algorithm (left) and PCLMS algorithm (right) in real-time. The solid line [—] represents the uncontrolled output, the dashed line [- -] the controlled output.

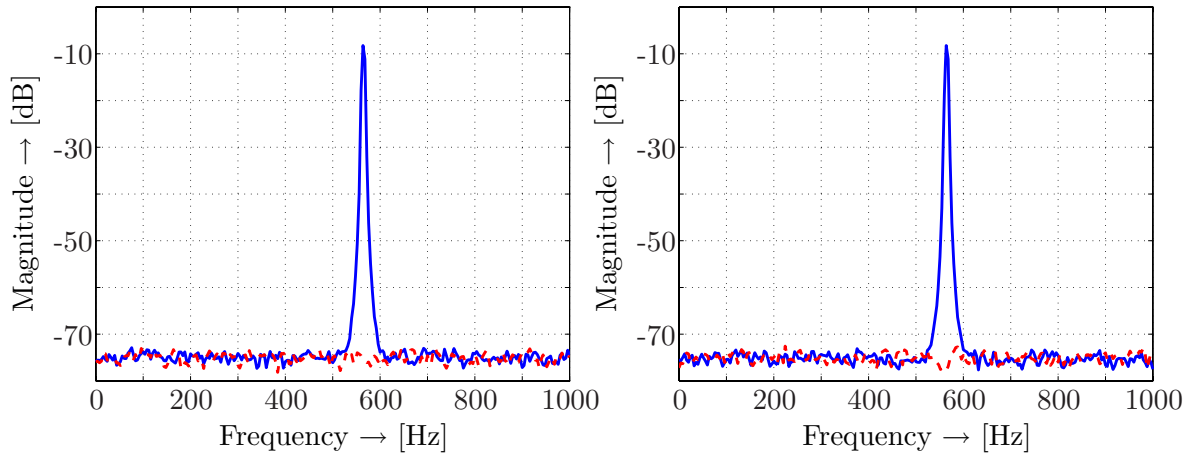


Figure 5.4: Controlling a single frequency disturbance of 565 Hz with the FxLMS algorithm (left) and the PCLMS algorithm (right) in simulation. The solid line [—] represents the uncontrolled output, the dashed line [- -] represents the controlled output.

set-up. Therefore, it can be concluded that the adaptive controllers act as desired. Note that several other small peaks are visible in the spectrum originating from the real-time data e.g. at 20 Hz, 260 Hz, 415 Hz and 605 Hz. These peaks are due to measurement noise present in the background and are not introduced by the controller.

It can be concluded from these first single frequency experiments that both algorithms are suitable to suppress the 528 Hz disturbance, but that the PCLMS algorithm exhibits a faster convergence speed.

5.2.3 Eliminating a disturbance output containing 565 Hz

In the second experiment it is attempted to eliminate a disturbance output that contains a single frequency of 565 Hz. The performance of both the FxLMS algorithm and the PCLMS algorithm in the simulation is shown in figure 5.4. A reduction of approximately 69 dB is obtained for both algorithms in the simulation, and the peak at 565 Hz is reduced to the noise level. It can again be concluded that a good performance is obtained and that the controllers act satisfactorily. In figure 5.5 the averaged adaptive controller coefficients are given for both algorithms. Within approximately 100 seconds (at $t=110$ s) the adaptive controller converges in the case of the FxLMS algorithm. The PCLMS algorithm still exhibits better convergence behavior than the FxLMS algorithm. In the case of the PCLMS algorithm, the controller converges rapidly, i.e. within approximately five seconds.

In figure 5.6 the real-time results are given for both algorithms. The peak at 565 Hz is reduced to the measurement noise level and a reduction of approximately 69 dB is achieved. This is in accordance with the one found in the simulation experiment. So far, it can be concluded that although the FxLMS algorithm exhibits slower convergence than the PCLMS algorithm, they are both able to eliminate the disturbances.

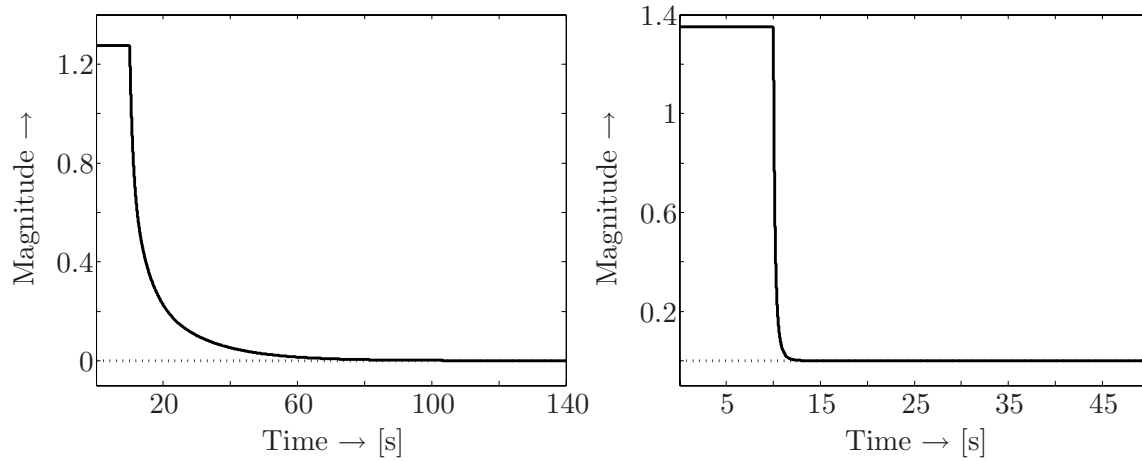


Figure 5.5: Averaged adaptive controller coefficients when eliminating the 565 Hz disturbance in simulation with the FxLMS algorithm (left) and the PCLMS algorithm (right)

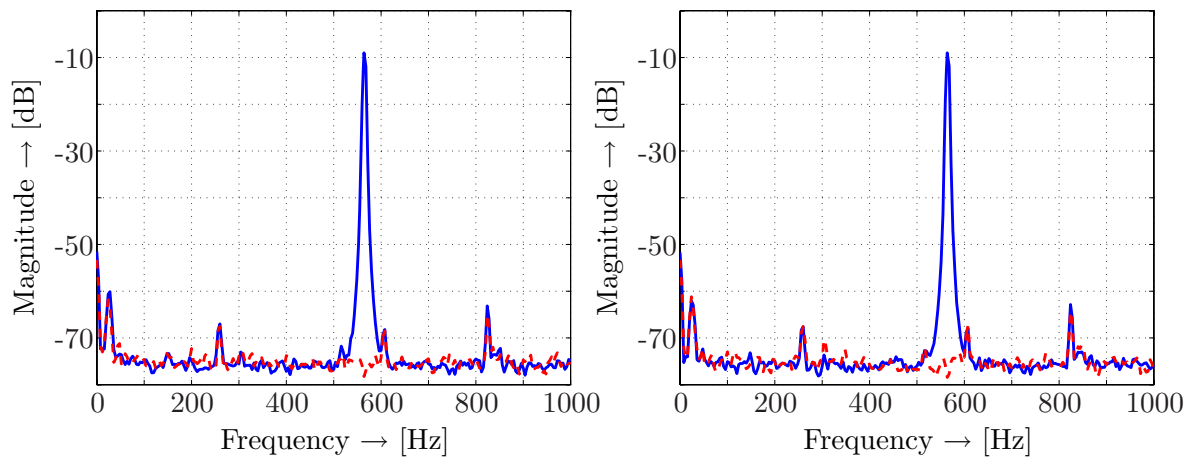


Figure 5.6: Controlling a single frequency disturbance of 565 Hz with the FxLMS algorithm (left) and PCLMS algorithm (right) in real-time. The solid line [—] represents the uncontrolled output, the dashed line [- -] the controlled output.

Table 5.2: Eigenvalue-spread and step-sizes in the experiment for both the FxLMS and the PCLMS algorithm.

	496 Hz	660 Hz
Eigenvalue-spread	1169	139
Step-size μ	0.48	1.27

5.3 Eliminating a disturbance output containing two frequencies

Section 5.3.2 discusses the experiment in which two frequencies need to be eliminated in the disturbance output. But first the preliminaries are discussed in section 5.3.1.

5.3.1 Preliminaries concerning the experiments

One experiment is performed in which two frequencies are eliminated. Specifically, the reference input $x(k)$ consists of the two frequencies 496 Hz and 660 Hz. Table 5.2 shows the eigenvalue-spread for the frequencies and also the step-sizes μ that are used.

In the simulation experiments, $\zeta(k)$ is maintained at approximately 25 dB SNR with respect to the disturbance output $\mathbf{d}(k)$. As in the previous experiments, in both the simulation experiments and real-time experiments the adaptive controller is switched on after $t = 10$ seconds, i.e. the adaptation of the controller parameters starts after $t = 10$ seconds. In all cases the adaptive controller parameters are initialized with zeros before the adaptation starts. Every experiment takes 200 seconds, which is 60 seconds longer than the experiments discussed previously. This is done since the convergence time is longer when two frequencies need to be eliminated.

As described in (Elliott, 2000b, Chapter 4, page 178), to eliminate two disturbance frequencies per experiment, two separate controllers are implemented, each eliminating a single frequency. Therefore, the reference input $x(k)$ is split up into two ‘sub’-reference inputs $x_1(k)$ and $x_2(k)$, with every sub-reference input containing a single frequency. This implementation approach is also proposed in (Johansson, 1998). In a true real-time application, the disturbance sources that each generate a different disturbance frequency must be uncorrelated to be able to distinguish between the different frequencies.

5.3.2 Performance of the algorithms

Both the FxLMS algorithm and the PCLMS algorithm are implemented in the simulation. For the FxLMS algorithm, FIRFs are extracted from the obtained secondary path model $\hat{\mathbf{S}}$ at 496 Hz and 660 Hz (as described earlier), that are implemented to represent the secondary path model $\hat{\mathbf{S}}$. For the PCLMS algorithm the transfers $\hat{\mathbf{U}}(e^{j2\pi 496T})$, $\hat{\mathbf{V}}(e^{j2\pi 496T})$, $\hat{\mathbf{U}}(e^{j2\pi 660T})$ and $\hat{\mathbf{V}}(e^{j2\pi 660T})$ are implemented. The models that are obtained in section 4.3 are used to represent the true primary path \mathbf{P} and secondary path \mathbf{S} in simulation.

Figure 5.7 shows the simulation result for the FxLMS algorithm on the left and for the PCLMS algorithm on the right. The two peaks at 496 Hz and 660 Hz are clearly visible in the disturbance output. The two peaks are suppressed by the adaptive controllers. In the case of

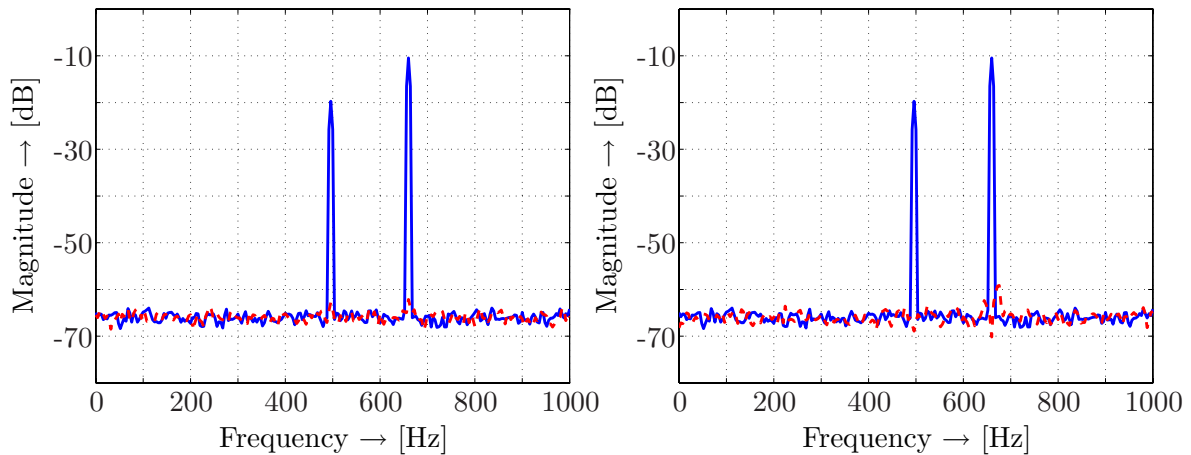


Figure 5.7: Controlling the disturbance output containing the two frequencies of 496 and 660 Hz with the FxLMS algorithm (left) and the PCLMS algorithm (right) in simulation. The solid line [—] represents the uncontrolled output, the dashed line [- -] represents the controlled output.

the FxLMS algorithm, the first peak is reduced by approximately 42 dB, while the second peak is reduced by approximately 51 dB. It can be seen though that - within the selected time frame - the FxLMS algorithm is not able to suppress the two frequencies completely and a small residual remains. This can be explained by the fact that the adaptive controller parameters do not converge completely in the 200s time frame. The PCLMS algorithm achieves a better performance within the 200 second time frame. The first peak is reduced by approximately 49 dB, while the second peak is reduced by 60 dB. The slower convergence in the case of the FxLMS algorithm can be explained by the large eigenvalue-spread that is present at 496 Hz in the secondary path. Slower convergence can be seen in figure 5.8 in which the normalized adaptive controller coefficients are shown with respect to time⁴. After 200 seconds the FxLMS algorithm still is not completely converging, whereas the PCLMS algorithm converges within approximately ten seconds after the controller is switched on. This shows the advantage of the PCLMS algorithm over the FxLMS algorithm, also when it concerns suppressing two frequencies. Still, both algorithms perform well.

Both of the adaptive controllers are also implemented in real-time. The results of the experiments are shown in figure 5.9. As in the simulation results, it can be concluded that the peaks are reduced to the noise level. The first peak is reduced by 55 dB and the second peak by 63 dB. As expected from the simulations, the PCLMS algorithm performs better than the FxLMS algorithm. The PCLMS algorithm reduces the first peak by 58 dB and the second peak by 67 dB. Obviously, the performance in the simulation is worse than the performance in real-time, since the SNR in the simulation is higher. The simulation results and the real-time results correspond well and the controllers act as desired. Furthermore, it can be concluded that the PCLMS algorithm outperforms the FxLMS algorithm with regards to convergence speed.

⁴In the case of the PCLMS algorithm a smaller time frame of 100 seconds is selected to make it more visible.

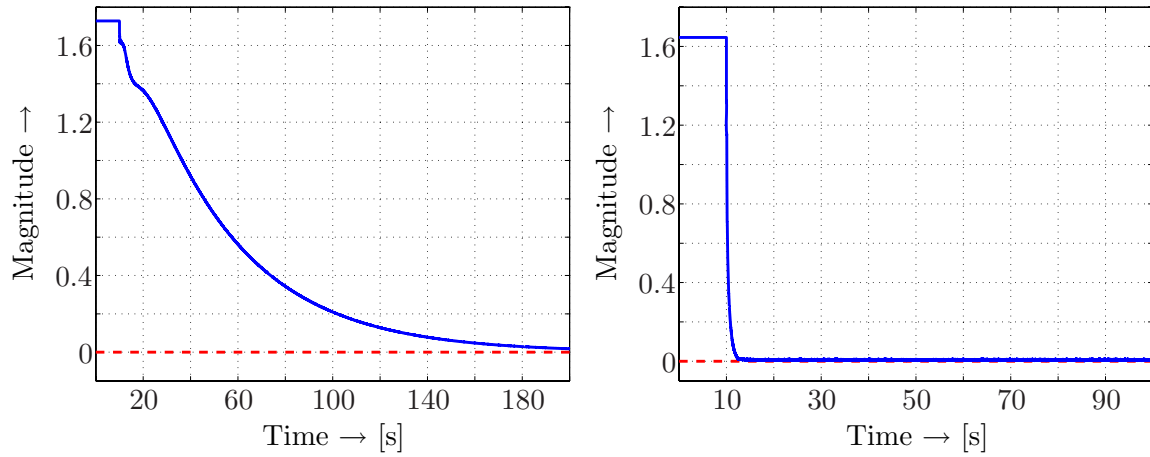


Figure 5.8: Averaged adaptive controller coefficients when eliminating the 496 Hz and 660 Hz disturbance in simulation with the FxLMS algorithm (left) and the PCLMS algorithm (right).

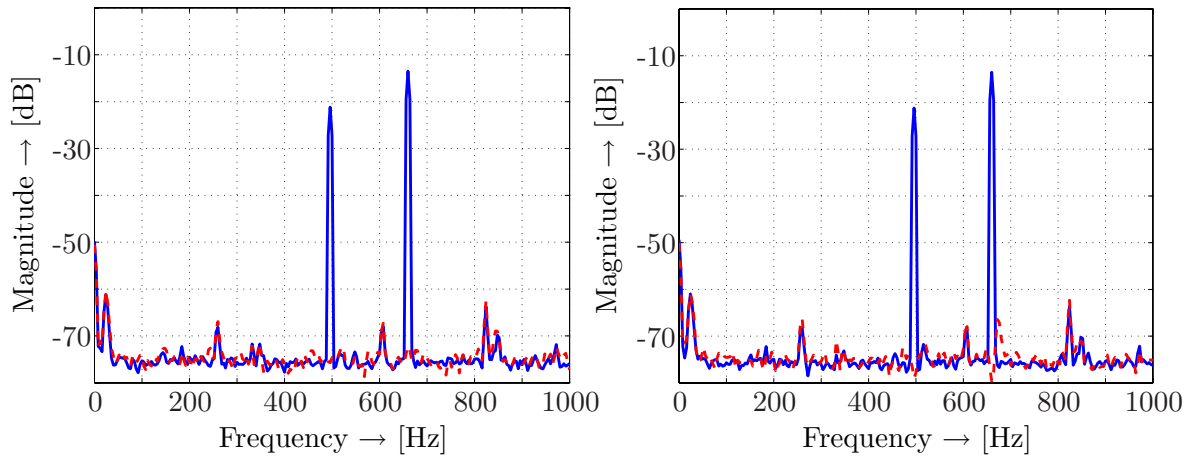


Figure 5.9: Controlling the disturbance output containing the two frequencies of 496 Hz and 660 Hz with the FxLMS algorithm (left) and the PCLMS algorithm (right) in real-time. The solid line [—] represents the uncontrolled output, the dashed line [- -] represents the controlled output.

Chapter 6

Control of broadband disturbances on the vibration isolation setups

6.1 Introduction

Usually the disturbance output has broadband characteristics instead of narrowband ones, and a different controller scheme is needed that can suppress a broadband disturbance output. Furthermore, in the previous chapter the use of a feedforward reference input was discussed. In real-life applications, however, a feedforward reference input is often not available or might be difficult to obtain. In such cases, a feedback controller needs to be used that operates without a feedforward reference input, but relies on the error sensor outputs. For an explanation of this, see chapter 2.

The main contribution of this chapter is the implementation and validation of the inner/outer based adjoint least mean square (IOALMS) algorithm (see section 3.4.2). Implementation is performed on both the SR-SISO vibration isolation setup and the SR-MIMO vibration isolation setup. See chapter 1 for a description of these setups. On the SR-SISO setup only the feedback control scenario is investigated. On the SR-MIMO setup both the feedforward control scenario and the feedback control scenario are investigated (see section 2.2). An additional contribution of this chapter is that the corresponding fixed gain controllers are implemented and validated, providing the ability to check the performance of the adaptive controllers. Throughout this chapter it is assumed that the reference input is a zero mean stochastic white noise sequence as discussed in chapter 2.

This chapter is organized as follows. Section 6.2 describes feedback control experiments that are performed on the SR-SISO vibration isolation setup. In section 6.3 feedforward and feedback control experiments are discussed that are performed on the SR-MIMO vibration isolation setup.

6.2 SR-SISO vibration isolation setup

This section is organized as follows. First, in section 6.2.1 the preliminaries are treated including, for instance, the choice of the number of coefficients in the controller. Section 6.2.2 explains how certain models in the control scheme are determined. Section 6.2.3 describes

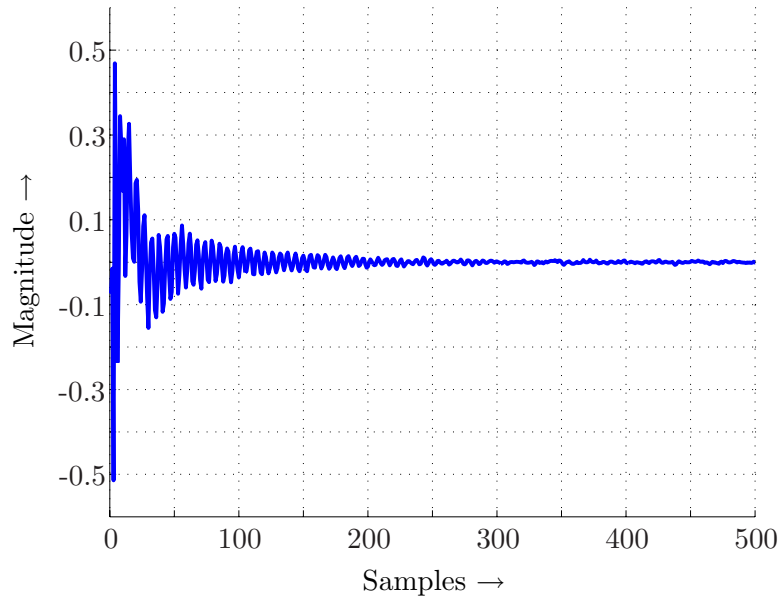


Figure 6.1: Impulse response of the fixed gain controller $\check{\check{W}}_o$.

the simulation experiments in Matlab and the real-time experiments on the dSPACE 1103 system as introduced in section E.2.1. Note that in this chapter models are often used, and thus, frequently the symbol $\widehat{(\cdot)}$ is used.

6.2.1 Preliminaries concerning the experiments

Since the SR-SISO vibration isolation setup has a single reference input $x(k)$, single actuator input $u(k)$ and a single error sensor output $e(k)$, $J = M = K = 1$ in figure 3.13. The IOALMS algorithm for the feedback control scenario is given in equation (3.72). The optimum value of the regularization parameter ρ in equation (3.72) is found to be $\rho = 1.36 \cdot 10^{-4}$. The regularization parameter ρ is set to compensate for model uncertainties so that a stable closed loop is obtained (see section 3.2.5). To speed up the convergence of the adaptive controller coefficients $\check{\check{w}}_o(k)$, the inverse outer factor model $\widehat{\mathbf{S}}_o^{-1}$ is incorporated in the control scheme, which is obtained from an inner/outer factorization of the secondary path model $\widehat{\mathbf{S}}$ as in equation (3.58). The inverse outer factor (with the $\check{(\cdot)}$ omitted and the $\widehat{(\cdot)}$ added) in figure 3.13 is not regularized i.e. $\varpi = 0$ in equation (3.65). The adaptive controller $\check{\check{W}}_o(k)$ in figure 3.13 has a FIRF structure, to create a problem that is linear in the coefficients. For an explanation of this, see the bottom part of page 39. To determine the adaptive controller length L in equation (3.6), i.e. the length of the vector $\check{\check{w}}_o(k)$ in equation (3.72), first the regularized fixed gain controller $\check{\check{W}}_o$ is computed in equation (3.14), of which the controller coefficients $\check{\check{w}}_o$ are determined using equation (3.33). The coefficients are derived using 10^5 samples of data for the auto-correlation matrix and the cross-correlation vector. See figure 6.1 for a depiction of the impulse response of the fixed gain controller $\check{\check{W}}_o$. Looking at the impulse response, the controller length is found to be $L = 200$. Starting approximately from there, the tail of the impulse response is almost zero. Furthermore, in equation (3.72), p and μ are tuning factors that are discussed later. In the simulation experiments approximately 25 dB SNR of measurement noise $\zeta(k)$ is added to the disturbance output $d(k)$. In both

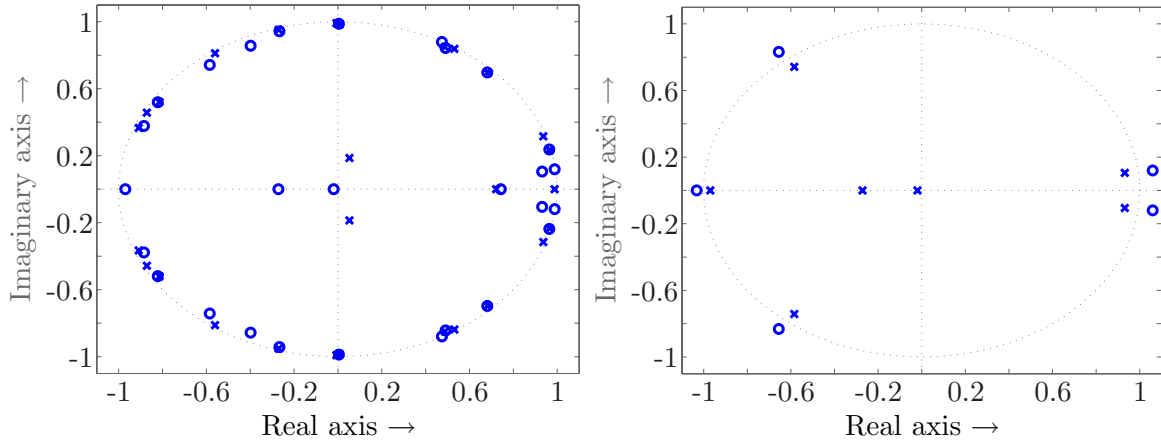


Figure 6.2: Pole/zero map of the outer factor model $\widehat{\mathbf{S}}_o$ (left) and the inner factor model $\widehat{\mathbf{S}}_i$ (right).

the simulation experiments and the real-time experiments the adaptive controller $\check{\mathbf{W}}_o(k)$ is switched on after one second. Furthermore, the controller parameters $\check{\mathbf{w}}_o(k)$ are initialized with zeros before the adaptation starts.

6.2.2 The inverse outer factor and the adjoint of the inner factor

The IOALMS algorithm requires the FIRF $\widehat{\mathbf{S}}_{i\text{FIRF}}^*$ of the adjoint inner factor model, $\widehat{\mathbf{S}}_i^*$, and the model of the inverse outer factor, $\widehat{\mathbf{S}}_o^{-1}$. These two models are obtained from the inner factor model $\widehat{\mathbf{S}}_i$ and the outer factor model $\widehat{\mathbf{S}}_o$, respectively. The latter two models can be found by an inner/outer factorization of the secondary path model $\widehat{\mathbf{S}}$ in equation (3.58). In figure 6.2 a pole/zero map is given of the outer factor model $\widehat{\mathbf{S}}_o$ and of the inner factor model $\widehat{\mathbf{S}}_i$. There are seven non-minimum phase zeros present in the secondary path model $\widehat{\mathbf{S}}$ (see section 4.2.2) which are captured in the inner factor model $\widehat{\mathbf{S}}_i$. Two zeros on the real axis are outside the axis of this plot: a zero at -3.69 and a zero at -51.88. The outer factor model $\widehat{\mathbf{S}}_o$ is stably invertible: all of its poles and zeros are strictly inside the unit circle. Compare these pole/zero maps with the pole/zero map of the secondary path model $\widehat{\mathbf{S}}$ depicted on the right in figure 4.4. That makes clear in what model particular poles and zeros are integrated. Bode diagrams of the 7th order inner factor model and the 28th order outer factor model are shown in figure 6.3. These figures show that the outer factor model has the same magnitude behavior as the secondary path model (see figure 4.6). Furthermore, as expected, the inner factor model has unit magnitude over the entire frequency range. It is explained why this is true on page 55, in particular by equation (3.59).

Given the inner factor model and the outer factor model, the adjoint of the inner factor model $\widehat{\mathbf{S}}_{i\text{FIRF}}^*$ and the inverse outer factor model $\widehat{\mathbf{S}}_o^{-1}$ are computed. To start with the latter: given the system matrices of the outer factor model $\widehat{\mathbf{S}}_o$, computing the inverse $\widehat{\mathbf{S}}_o^{-1}$ can easily be done in e.g. Matlab by using the command `inv`. Subsequently, the adjoint of the inner factor model $\widehat{\mathbf{S}}_{i\text{FIRF}}^*$ is determined. See the footnote on page 60 for a procedure. The left plot of figure 6.4 shows the impulse response of the inner factor model $\widehat{\mathbf{S}}_i$. The response dies down after approximately 200 samples. One could cut it off earlier, but here it has been chosen to

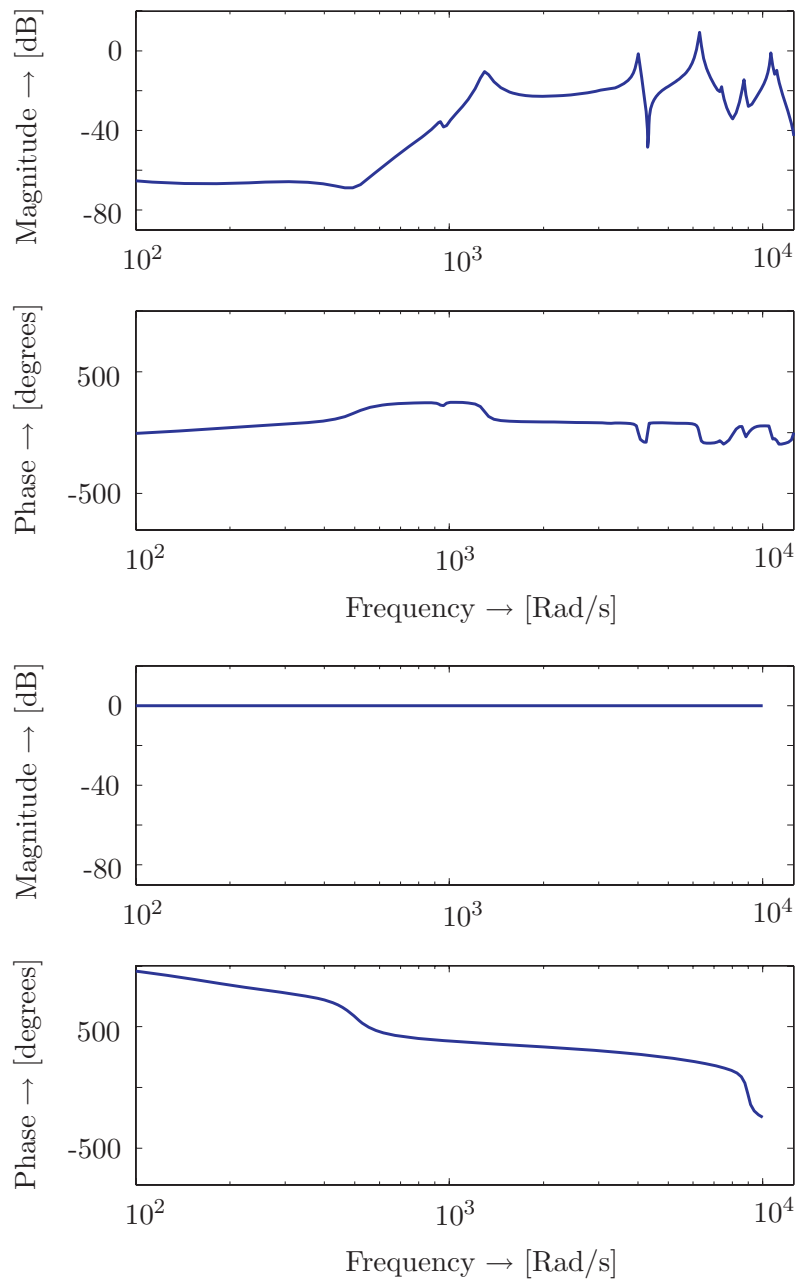


Figure 6.3: Bode diagram of the outer factor model $\hat{\mathbf{S}}_o$ (upper) and the inner factor model $\hat{\mathbf{S}}_i$ (lower).

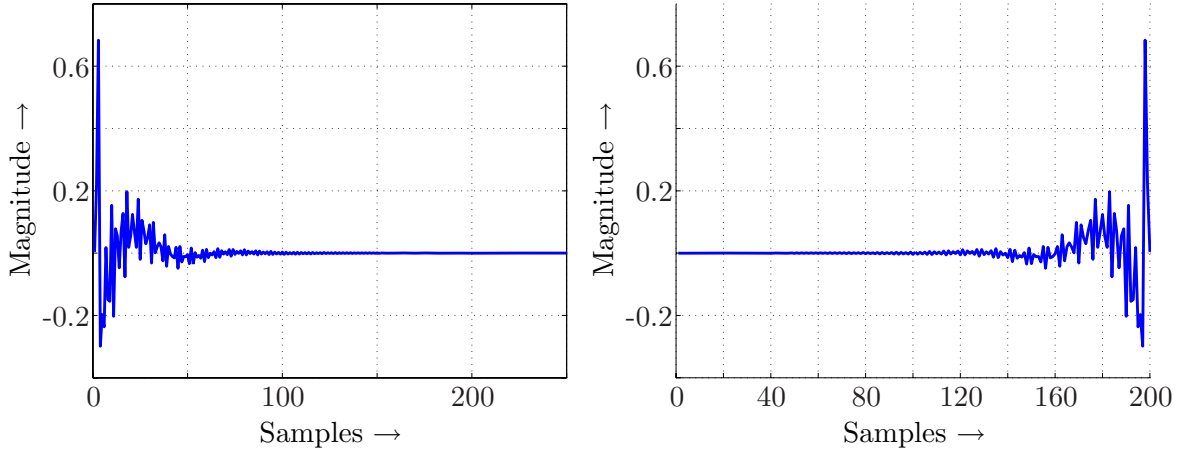


Figure 6.4: Impulse response of $\hat{\mathbf{S}}_i$ (left) and $q^{-p}\hat{\mathbf{S}}_i^*$ (right).

be on the safe side. Therefore, $\hat{\mathbf{S}}_{i\text{FIRF}}^*$ is estimated to have a length of 200 coefficients. This means that the parameter $p = 199$ in equation (3.72). On the right, figure 6.4 shows the 200 samples of the impulse response of $q^{-199}\hat{\mathbf{S}}_{i\text{FIRF}}^*$, which is a time-reversed and delayed copy of the impulse response of $\hat{\mathbf{S}}_i$ (see section 3.4.2 for details).

6.2.3 Fixed gain and adaptive feedback control

Due to the inverse outer factor model $\hat{\mathbf{S}}_o^{-1}$ in the control scheme and the regularization factor ρ , a low eigenvalue-spread of order 10 results in equation (3.33). For a definition of the eigenvalue-spread for the feedforward control scenario, see equation (3.49). Note that 10 is small considering the fact that the eigenvalue-spread should ideally be equal to unity for optimal convergence speed (see page 51). Furthermore, without the inverse outer factor model $\hat{\mathbf{S}}_o^{-1}$ in the control scheme and with $\rho = 0$, the eigenvalue-spread is of order 10^9 . Equation (3.48) learns that theoretical convergence would take months. With the inverse outer factor model $\hat{\mathbf{S}}_o^{-1}$ incorporated in the control scheme and with $\rho = 0$, the eigenvalue-spread is of order 10^4 . Although still not equal to unity, this is much lower. It demonstrates that post-conditioning considerably speeds up the convergence time of the adaptive controller coefficients $\check{\mathbf{w}}_o(k)$. After careful analysis the step-size μ in equation (3.72) is set to $\mu = 0.22$.

Figure 6.5 shows the learning curve in the error sensor output $e(k)$ resulting from the simulations. For information on what a learning curve is, see e.g. (Nascimento and Sayed, 1998). The results are averaged over 50 independent runs for the measurement noise $\zeta(k)$ and the disturbance input $x(k)$. In the figure, the dash-dot ‘horizontal’ line at approximately -5.2 dB represents the optimum performance that can be achieved in simulation by the fixed gain controller $\check{\mathbf{K}}$ in equation (3.74). The solid line represents the learning curve that is established by the regularized adaptive controller $\check{\mathbf{K}}(k)$ in equation (3.73). Within 25 seconds the adaptive controller $\check{\mathbf{K}}(k)$ has realized a reduction of 5.1 dB, which is very close to the optimum of 5.2 dB that is established by the regularized fixed gain controller $\check{\mathbf{K}}$. The reason that the performance of the regularized adaptive controller $\check{\mathbf{K}}(k)$ is not optimal (i.e. 5.2 dB) is due to a faulty adjustment of the error sensor output $e(k)$. This concerns the effect in the error sensor output due to the variation of the adaptive controller parameters around

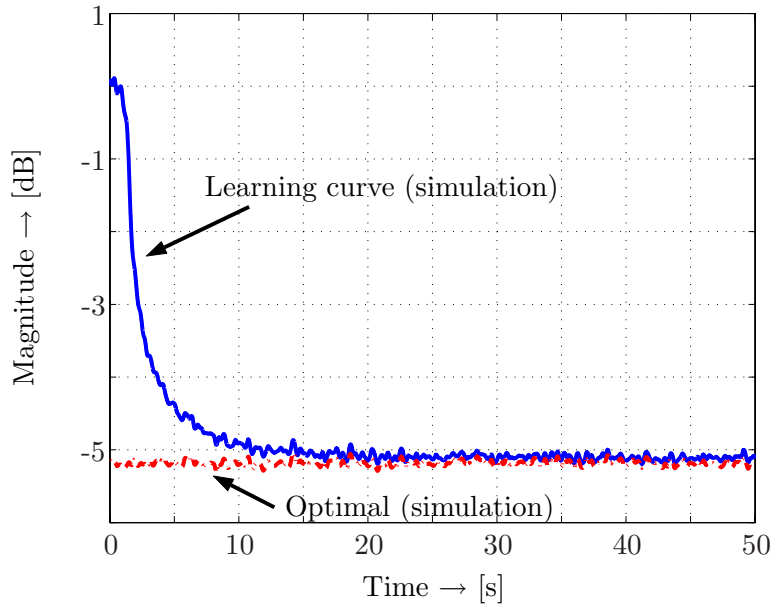


Figure 6.5: Learning curve in the error sensor output when controlling a broadband disturbance output with the IOALMS algorithm in simulation.

Table 6.1: The reduction of the four main resonance peaks of the SR-SISO vibration isolation setup in simulation (see figure 6.6).

Resonance peak	Peak 1	Peak 2	Peak 3	Peak 4
U: Uncontrolled magnitude (dB)	-27.3	-31.1	-24.0	-38.9
C: Controlled magnitude (dB)	-35.8	-36.5	-37.2	-41.5
U-C: Reduction (dB)	8.5	5.4	13.2	2.6

the optimum ones, see e.g. (Elliott, 2000b). To further demonstrate the reduction that is obtained in the error sensor output, see figure 6.6 in which spectra are given of the controlled output and the uncontrolled output. What is meant by this is the spectrum of the error sensor output, computed using the error sensor output data of the last second of a single experiment, with the converged controller $\hat{\mathbf{K}}(k)$ switched on and switched off, respectively. The four main resonance peaks of the setup are clearly suppressed. At some points in between the resonance peaks, there is amplification instead of suppression. This is known as the waterbed effect: diminishing the disturbance at some points will cause amplification at other points (Doyle et al., 1992). An overview of the suppression of the resonance peaks in simulation is given in table 6.1.

With these simulation results in mind, the adaptive feedback controller is implemented in real-time on the dSPACE 1103 system. In figure 6.7 the learning curve is presented that is established in the real-time experiments. Here the results are averaged over five independent runs for the disturbance input $x(k)$. The real-time results closely match the simulation results. From figure 6.7 it can be concluded that the convergence rate established in the real-time

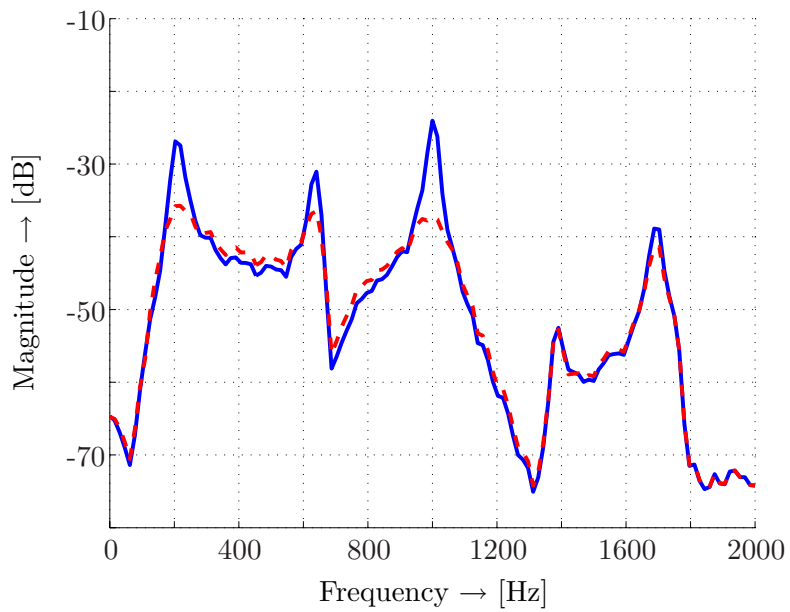


Figure 6.6: Spectrum of the error sensor output in simulation when the adaptive controller is switched off (solid, [—]) or on (dashed, [- -]).

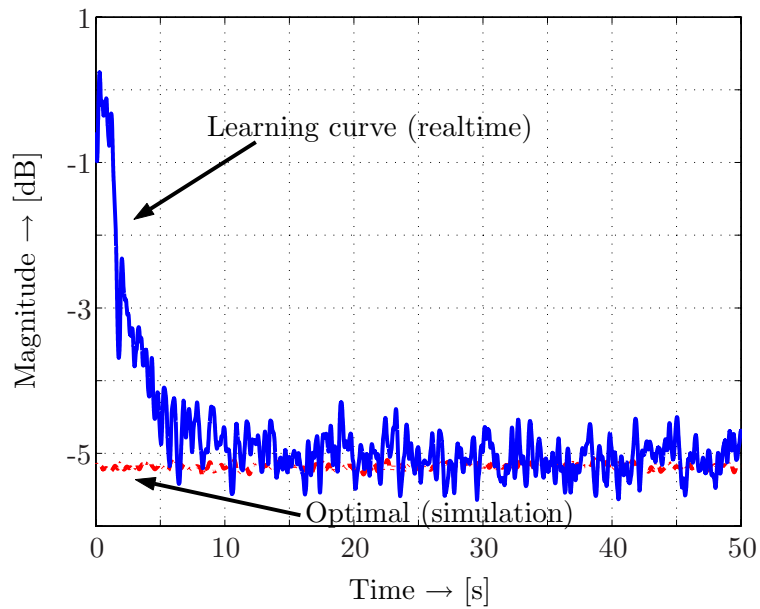


Figure 6.7: Learning curve in the error sensor output $e(k)$ when controlling a broadband disturbance output with the IOALMS algorithm in real-time.

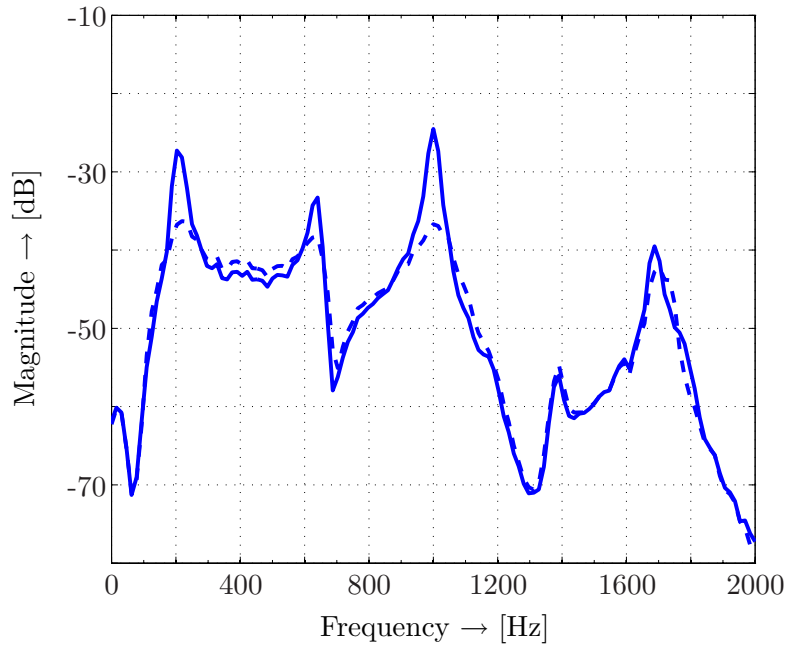


Figure 6.8: *Left:* Spectrum of the error sensor output in real-time when the adaptive controller is switched off (solid, [—]) or on (dashed, [- -]).

Table 6.2: *The reduction of the four main resonance peaks of the SR-SISO vibration isolation setup in real-time (see figure 6.8).*

Resonance peak	Peak 1	Peak 2	Peak 3	Peak 4
U: Uncontrolled magnitude (dB)	-27.3	-33.3	-24.5	-39.5
C: Controlled magnitude (dB)	-36.3	-38.3	-36.7	-42.2
U-C: Reduction (dB)	9.0	5.0	12.2	2.7

experiments is almost equivalent to the one found in the simulation experiments. A reduction of 5.0 dB is accomplished, which is only 0.1 dB less than the reduction obtained in simulation. The difference in reduction can be explained by the perfect plant knowledge assumption in simulation, which is not the case in real-time. After approximately 25 seconds, the regularized controller $\hat{\mathbf{K}}(k)$ has converged in real-time, just as it was observed in simulation. Since the learning curve established from the real-time experiments is based on an average of only five trials, the curve shows a larger variance than the learning curve shown in figure 6.5 that is established from the simulation experiments. Still, it is visible that the performance achieved in real-time is similar to the performance obtained in simulation. To further demonstrate the reduction obtained in real-time, figure 6.8 gives spectra of the controlled output and the uncontrolled output. An overview of the suppression of the resonance peaks in real-time is given in table 6.2. This concludes the discussion of the experiments on the SR-SISO vibration isolation setup. In the next section experiments on the SR-MIMO vibration isolation setup are described.

Table 6.3: *The reduction in dB within the six error sensor outputs that are obtained with the unregularized full order fixed gain feedforward controller in simulation.*

Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5	Sensor 6	Average
13.8	14.0	10.2	8.1	10.3	12.9	11.5

6.3 SR-MIMO vibration isolation setup

This section is organized as follows. In section 6.3.1 the results are described that are obtained with fixed gain feedforward controllers. Section 6.3.2 discusses the results that are achieved with the adaptive feedforward controller. The results obtained by the fixed gain feedback controller are presented in section 6.3.3. Finally, in section 6.3.4 the results are described that are achieved with the adaptive feedback controller. In each section, both simulation and real-time experiments are described. For control experiments, the dSPACE 1005 system is used (see section E.2.2).

6.3.1 Fixed gain feedforward control

In the previous section, that describes control on the SR-SISO vibration isolation setup, regularization is applied to stabilize the closed loop (i.e. $\rho > 0$ in e.g. equation (3.33)). In this section, the controller is also determined with regularization. However, as will become clear later, here regularization is needed to prevent actuator saturation instead of stabilizing the closed loop. Particularly, the unregularized controller cannot be implemented and executed in real-time due to oversteering of the actuators. Although the unregularized controller is not used, it is computed to be able to check the performance of its regularized counterpart.

Without regularization

The one input six output fixed gain controller \mathbf{W} is derived according to the model-based design technique discussed in appendix B. See also problem 3.2.1 on page 35 for the control objective and figure 3.1 for a block diagram of the control scheme. The controller is computed using the identified models of the primary path and secondary path, the identification of which is described in chapter 4. For the unregularized controller it holds true that $\beta = 0$ in equation (B.3). The causal/anti-causal split in equation (B.3) is performed in state space using the discrete-time algorithm that was developed in the framework of this thesis and that is described in (Leemhuis, 2004). To be in accordance with the real-time experiments, stochastic measurement noise is added to the error sensor outputs $\mathbf{e}(k)$ with a SNR of approximately 25 dB with respect to the disturbance outputs $\mathbf{d}(k)$. The identified models of the primary and secondary path are used to represent their true counterparts in simulation. After computing the full order controller, its performance is checked. The obtained reduction within the error sensor outputs is presented in table 6.3. An average reduction of 11.5 dB is achieved. This average is obtained by adding the results obtained in all six error sensor outputs and by dividing the result by six. The performance obtained in the first error sensor is shown in figure 6.9. The results for the other five error sensor outputs are not shown here for reasons of conciseness, but these displayed similar behavior. The solid line represents the error sensor output with the controller switched off, which means that only the disturbance

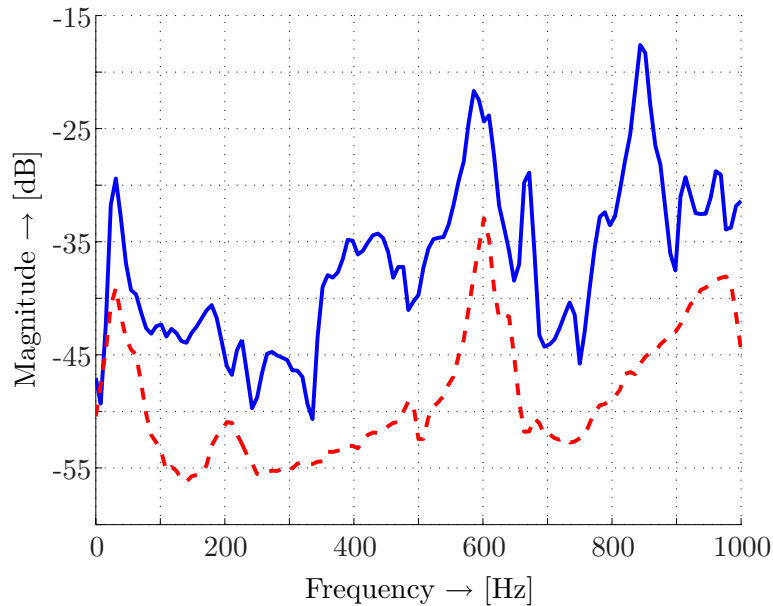


Figure 6.9: Performance of the unregularized fixed gain controller in simulation. Spectrum of the first error sensor output with (dashed, [- -]) and without (solid, [—]) control.

output contaminated with the measurement noise is present in the error sensor output. The dashed line shows the error sensor output if the controller is switched on. As these results make clear, over the entire frequency range from 0-1 kHz the disturbance output is suppressed. Below approximately 50 Hz and roughly between 560 Hz and 640 Hz the control performance is less than at other frequencies. This can be explained by the fact that in the fixed gain controller \mathbf{W} , the part $-\widehat{\mathbf{S}}_o^{-1}[\widehat{\mathbf{S}}_i^* \widehat{\mathbf{P}}]_-$ is not incorporated in equation (B.3) (with $[\cdot]_-$ the anti-causality operator), thereby compromising the performance. The anti-causal part originates from the inverted zeros of the secondary path model, which are outside the unit circle. In this control scheme, the anti-causal part can never be included in the controller, since it assumes knowledge of future reference inputs, which is not available¹. Thus, the anti-causal part needs to be neglected. It is concluded that the unregularized fixed gain controller \mathbf{W} performs as desired and that the results thus are promising. The controller, however cannot be implemented and executed in real-time on the dSPACE 1005 system because of two problems.

With regularization

For this, first see the block diagram depicted in figure 3.1. One problem is that the actuator inputs $\mathbf{u}(k)$ saturate at low frequencies due to a low-frequent high gain of the controller, i.e. the controller has a high magnitude in the low frequency area. Or, in other words: in the lower frequency region, the input $x(k)$ to the controller is strongly amplified at the outputs $\mathbf{u}(k)$. This can be seen in figure 6.10, which shows a Bode magnitude diagram of the controller from the first input to the first output. The diagrams from the first input to the other five outputs show similar high gain behavior and are omitted here to save space. From 0 Hz

¹In some applications, such as iterative learning control, this would be possible, see e.g. (Nijssse et al., 2001a; Phan and Frueh, 1998; Doh et al., 1999).

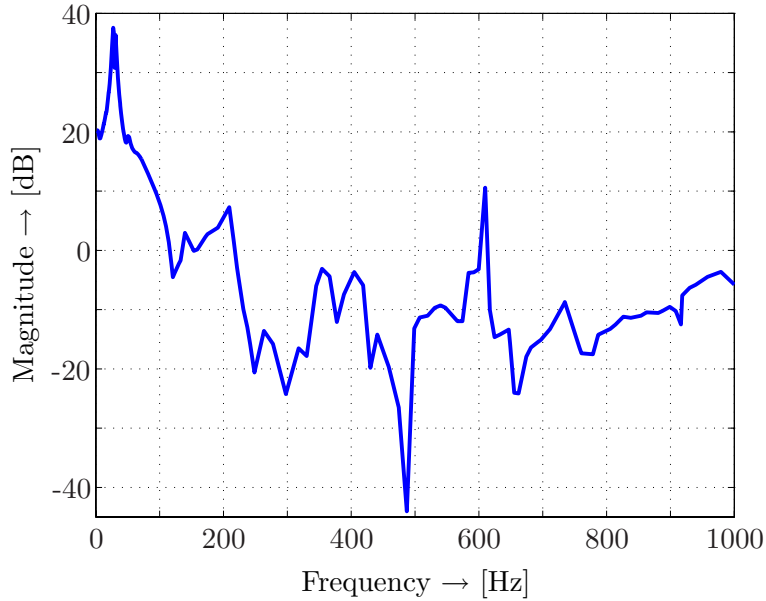


Figure 6.10: Bode magnitude diagram of the unregularized controller from its input to the first output.

up to approximately 200 Hz and close to approximately 600 Hz, the fixed gain controller has a significant gain. Real-time operation of the controller would result in actuator inputs $\mathbf{u}(k)$ that would be too high for the setup to handle, thus leading to saturation. That is not desirable, since saturation weakens the control performance. The low-frequent high gain can be explained by the low-frequent high gain of $\hat{\mathbf{S}}_o^{-1}$ in equation (B.3). In other words, $\hat{\mathbf{S}}_o$ has a low-frequent low gain. The latter can be explained by the fact that the secondary path model has a low-frequent low gain (see page 74). Therefore, to be able to implement the controller in real-time, the control effort is weighed by using regularization. The regularized controller can be designed using two approaches.

1. The regularized controller can be designed using $\beta > 0$ in equation (B.3) (see appendix B). This is the traditional, or one-step, regularization approach.
2. However, later on the inverse outer factor model \mathbf{S}_o^{-1} will be included in the adaptive control scheme. See figure 3.9 for a block diagram representation of this. Then an additional problem arises: the parameter ρ in equation (3.62) does not correspond to the parameter β in equation (B.3): ρ only affects $\check{\mathbf{W}}_o(k)$ and has no effect on the inverse outer factor \mathbf{S}_o^{-1} . Therefore, if the controller $\check{\mathbf{W}}$ is determined using the one-step approach, the performance of the controller $\check{\mathbf{W}}$ and that of the adaptive controller $\check{\mathbf{W}}(k)$ in equation (3.63) cannot be compared.

To be able to compare the performance of the fixed gain controller with that of the adaptive controller, regularization is performed in two steps as explained in section 3.4.1. First, a regularized outer factor is computed in equation (3.65). The regularization parameter is set to $\varpi = 0.06$. Second, the controller $\check{\mathbf{W}}_o$ is computed using the design technique as described in appendix B using $\beta = 3.0 \cdot 10^{-4}$. These values are determined in an iterative process so that

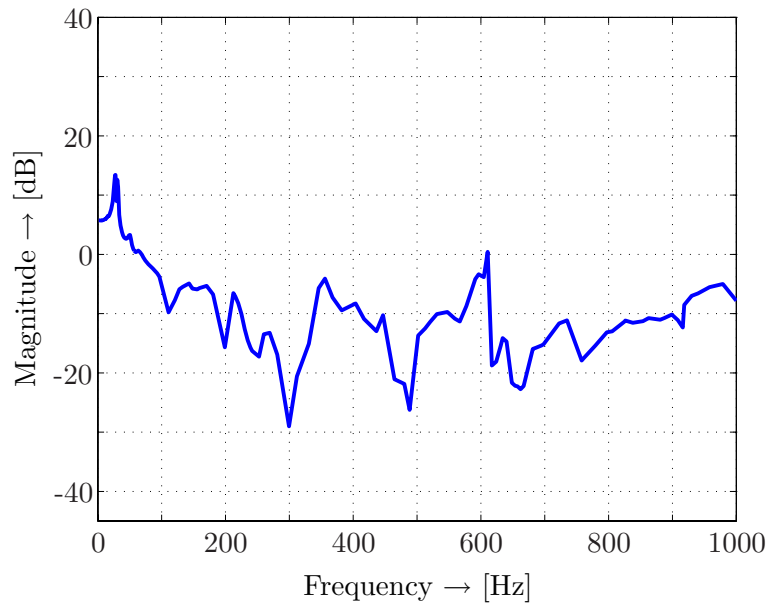


Figure 6.11: Bode magnitude diagram of the first channel of the regularized controller from its input to the first output.

the best possible performance in the error sensor outputs is established. The new regularized full order controller is given by equation (3.69). In figure 6.11 a Bode magnitude diagram of the controller is depicted. Comparison of this figure with figure 6.10 shows that the new controller has a much lower gain up to approximately 200 Hz and close to approximately 600 Hz. A minor side-effect is that at some frequencies gains are slightly higher (e.g. around 490 Hz). However, this effect is small and no further attention is paid to it.

Another problem is that the full order controller from equation (3.69) cannot be implemented and executed on the dSPACE system due to computational limitations. Therefore, the order of the controller is reduced by using controller order reduction. In particular, SMI is used to reduce the order of the controller by performing an identification experiment on the controller (see section 2.5.1). SMI has proven its use in reducing the model order in practice, see e.g. (Fraanje, 2004). The order reduction procedure is done in Matlab on the already computed regularized controller, according to the following procedure. The regularized controller is provided with a stochastic white noise input sequence of ten seconds and the output sequence of ten seconds is measured. Based on the obtained input/output data, a new reduced order controller is identified. The order of the controller is reduced to 135. The controller with order 135 can be executed without computational problems. A VAF of 99.9991% on the reduced order controller is obtained, which indicates that the lower order controller is almost as accurate as its full (higher) order counterpart. The fact that the order of the controller can be reduced so significantly without much performance loss can be explained by the redundancy present in the controller. There is a number of reasons for the redundancy (Fraanje, 2004, appendix B).

Simulating the reduced order regularized controller results in a performance as shown in table 6.4. For comparison, the table also includes the performance of the regularized controller when it is directly designed using the traditional regularization approach described

Table 6.4: The reduction in dB within the six error sensor outputs that are obtained with the regularized fixed gain feedforward controller in simulation.

	Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5	Sensor 6	Average
Two-step:	11.8	12.7	9.1	8.9	9.5	11.1	10.5
One-step:	11.7	12.7	8.9	8.7	9.4	11.2	10.4

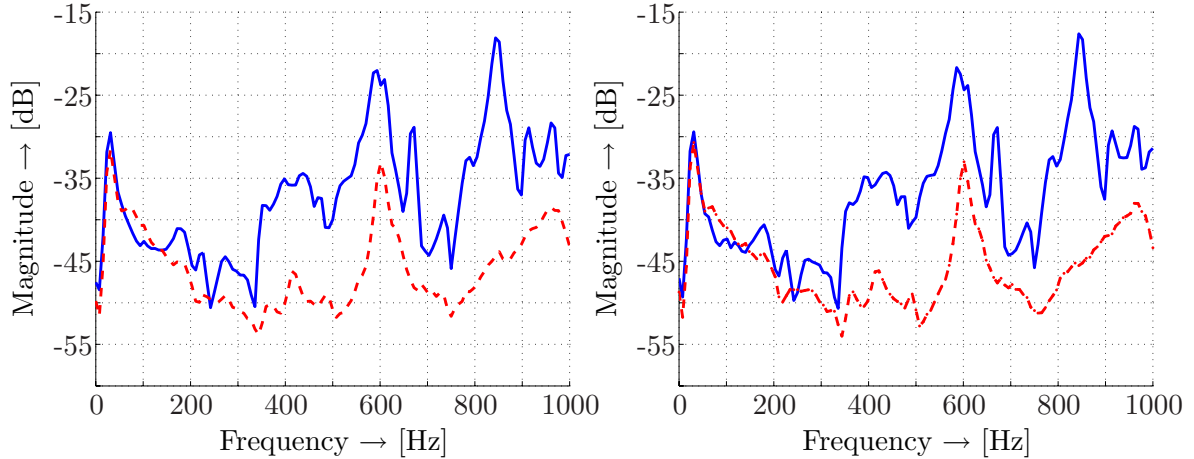


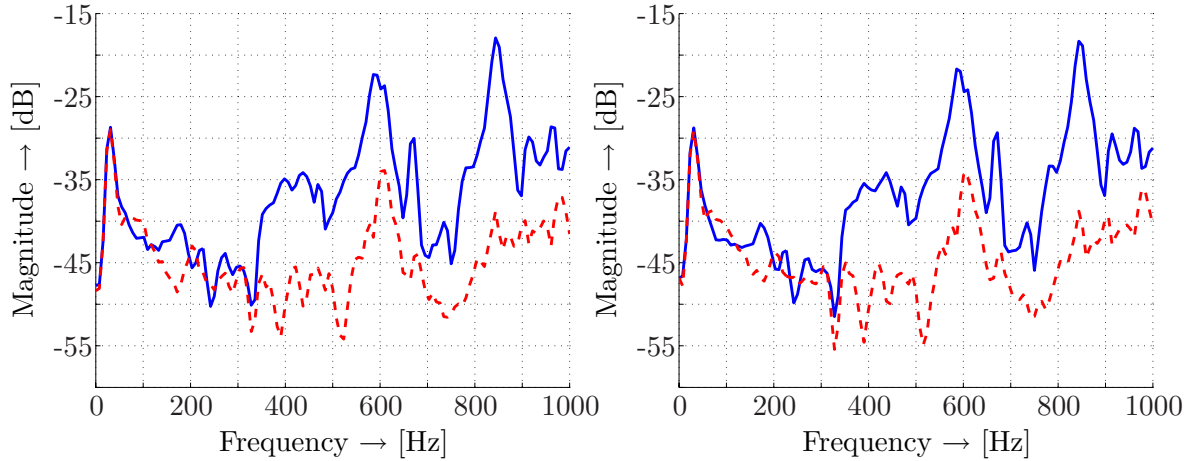
Figure 6.12: Performance of the regularized fixed gain controller in simulation with two-step regularization (left) and one-step regularization (right). Spectrum of the first error sensor output with (dashed, [- -]) and without (solid, [—]) control.

in appendix B. The regularization parameter is tuned to $\beta = 2.5 \cdot 10^{-5}$. Model order reduction is applied on the controller also and the order of the controller is reduced to 135. A VAF of 99.9990% on the reduced order regularized controller is obtained. An average performance of 10.5 dB is achieved in the error sensor outputs, which is 1.0 dB less than the performance of the unregularized controller (compare this with table 6.3). This performance loss is due to the regularization, obviously. The controller that is computed with the one-step approach yields a similar performance as the regularized controller that is designed with the two-step approach. Figure 6.12 shows the performance of the controller obtained in the first error sensor output (compare this with figure 6.9). The solid line represents the uncontrolled output, the dashed line represents the controlled output. The controller is not active in the low frequency area up to approximately 200 Hz. For frequencies higher than 200 Hz, the regularized controller yields a performance that is similar to that of the unregularized one.

It is concluded that the regularized fixed gain controller acts as desired. Therefore, the 135th order regularized controller is implemented in real-time. The results obtained in real-time are given in table 6.5 (compare this with table 6.4). Averaging 9.0 dB, the performance obtained in real-time is 1.5 dB less than the one obtained in simulation. This reduced performance can be explained by the fact that in simulation perfect plant knowledge is assumed, which is not the case in real-time. Again, the controller that is designed with the traditional regularization approach performs equal to the controller that is designed with the two-step approach. The performance of the controller on the first error sensor output is shown in figure

Table 6.5: The reduction in dB within the six error sensor outputs that are obtained with the reduced order regularized fixed gain feedforward controller in real-time.

	Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5	Sensor 6	Average
Two-step:	10.5	10.7	7.1	7.2	8.5	9.9	9.0
One-step:	10.7	11.0	7.3	7.3	8.6	10.2	9.1

**Figure 6.13:** Performance of the regularized fixed gain controller in real-time with two-step regularization (left) and one-step regularization (right). Spectrum of the first error sensor output with (dashed, [- -]) and without (solid, [—]) control.

6.13. It is concluded that the real-time results are in close agreement with the simulation results.

These results provide insight into the obtainable performance (since the fixed gain controller is the ‘optimum’ controller). Thus, the *adaptive* controller is implemented and validated, having the advantage with respect to the fixed gain controller of being able to adapt to small changes in the transfer paths and signals. For an explanation of this, see also chapter 2. Please note that from here on the one-step regularization approach is no further treated.

6.3.2 Adaptive feedforward control

Here the regularized adaptive controller is described, since the unregularized version cannot be implemented on the dSPACE system, i.e. the unregularized adaptive controller would oversteer the actuators. This is already known from the fixed gain control experiments.

A block diagram of the adaptive control scheme is shown in figure 3.12. As with the fixed gain control experiments on the SR-MIMO vibration isolation setup, the identified models of the primary and secondary paths are used to represent their true counterparts in simulation. The same regularization parameters are used as for the fixed gain controller. The regularized outer factor model from equation (3.65) with $\varpi = 0.06$, is inverted and incorporated in the adaptive control scheme to speed up the convergence of the adaptive controller coefficients $\check{\mathbf{w}}_o(k)$. The adaptive controller $\check{\mathbf{W}}_o(k)$ is updated using the IOALMS algorithm in equation

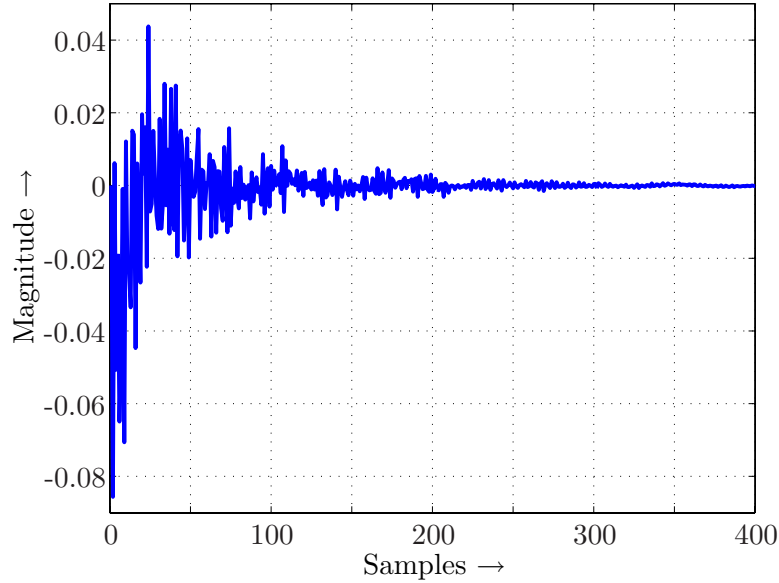


Figure 6.14: Impulse response of $\check{\mathbf{W}}_o$ from the first input to the first output.

(3.71). The adaptive controller $\check{\mathbf{W}}_o(k)$ is regularized with a factor $\rho = 3.0 \cdot 10^{-4}$ as given in equation (3.71). Note that it is necessary to regularize the (inverse) outer factor model. Not doing this and only regularizing $\check{\mathbf{W}}_o(k)$ would require such a high ρ that the performance of the adaptive controller would be considerably decreased. A reduction of more than 2 dB on average cannot be obtained in the error sensor outputs. Performing control experiments without the inverse outer factor model included in the control scheme (figure 3.7) is possible but would result in extremely slow convergence. Convergence would literally takes days.

Instead of using a controller that is a SSF, the adaptive controller $\check{\mathbf{W}}_o(k)$ is a FIRF to create a problem that is linear in the coefficients. For an explanation of this, see page 39. The number of adaptive controller coefficients in each of the six channels of the adaptive controller $\check{\mathbf{W}}_o(k)$ is obtained by examining the impulse response of the controller $\check{\mathbf{W}}_o$ (see figure 3.10). $\check{\mathbf{W}}_o$ has already been computed for the fixed gain control experiments. In figure 6.14 the impulse response of $\check{\mathbf{W}}_o$ is shown from the first input to the first output. The other five impulse responses are omitted for reasons of conciseness. An adaptive controller length of $L = 200$ is chosen here. More coefficients could not be implemented on the dSPACE system due to computational limitations. Choosing $L = 200$ means that a total $6 \times 200 = 1200$ coefficients have to be adapted at every sampling instant.

Due to the regularization, a low eigenvalue-spread of order 10^2 is accomplished (equation (3.49)). Low, that is, considering that the eigenvalue-spread is of order 10^5 if regularizing with ρ only and of order 10^3 if regularizing with ϖ only. The regularized adjoint inner factor model $\check{\mathbf{S}}_{i\text{FIRF}}^*$ (see figure 3.12) is used to filter the error sensor outputs and is described by 6×6 FIRFs. The number of coefficients p in equation (3.71) is derived by examining the impulse response of the regularized adjoint inner factor model $\check{\mathbf{S}}_i^*$. Figure 6.15 shows one of the 36 impulse responses of $\check{\mathbf{S}}_i^*$. Based on the impulse response, a value of 100 is chosen for the parameter p , so that $\check{\mathbf{S}}_{i\text{FIRF}}^*$ consists of $6 \times 6 \times 100$ coefficients. To save space the other 35 impulse responses are not shown. The step-size μ in the IOALMS algorithm is found

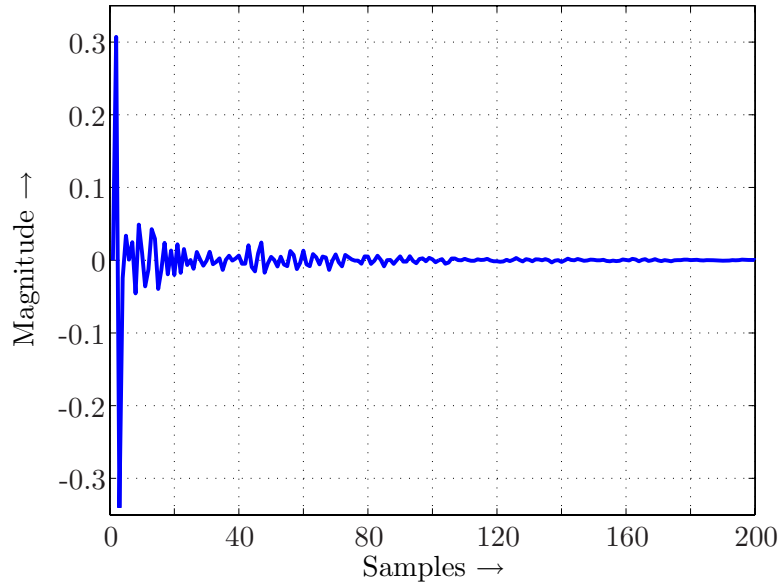


Figure 6.15: Impulse response of $\hat{\mathbf{S}}_i^*$ from the first input to the first output.

Table 6.6: The reduction in dB within the six error sensor outputs that are obtained with the regularized adaptive feedforward controller in simulation.

Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5	Sensor 6	Average
11.7	12.6	8.9	8.4	9.1	11.1	10.3

to be 0.01. The reduction in the error sensor outputs after 300 seconds of convergence is shown in table 6.6. It appears that the performance is in accordance with the performance of the fixed gain controller (compare the results with those in table 6.4): 10.3 dB for the adaptive controller versus 10.5 dB for the fixed gain controller. On the left, figure 6.16 shows the performance of the controller in the first error sensor output. It can be noted that the adaptive controller is not active in the low-frequency region, similar to the fixed gain controller (see the left depiction in figure 6.12). On the right, figure 6.16 shows a learning curve that indicates how fast the adaptive controller coefficients converge. Within approximately 40 seconds, the adaptive controller coefficients converge close to their optimum value.

With these simulation results in mind, the adaptive controller is implemented on the real-time setup. The results are shown in table 6.7. The performance of the adaptive controller is 0.4 dB better than the performance of the fixed gain controller in real-time, which is understandable. The fixed gain controller is dependent on models of both the primary and secondary paths. These two models both contain model uncertainty. The adaptive controller is only dependent on a model of the regularized inverse outer factor model and the regularized inner factor model. Both of these models stem from the secondary path model only, which means less uncertainty and thus slightly better results. Figure 6.17 presents the performance of the controller obtained in the first error sensor output. The results are in close agreement with the simulation results (see figure 6.16 for comparison) and the fixed gain control results in

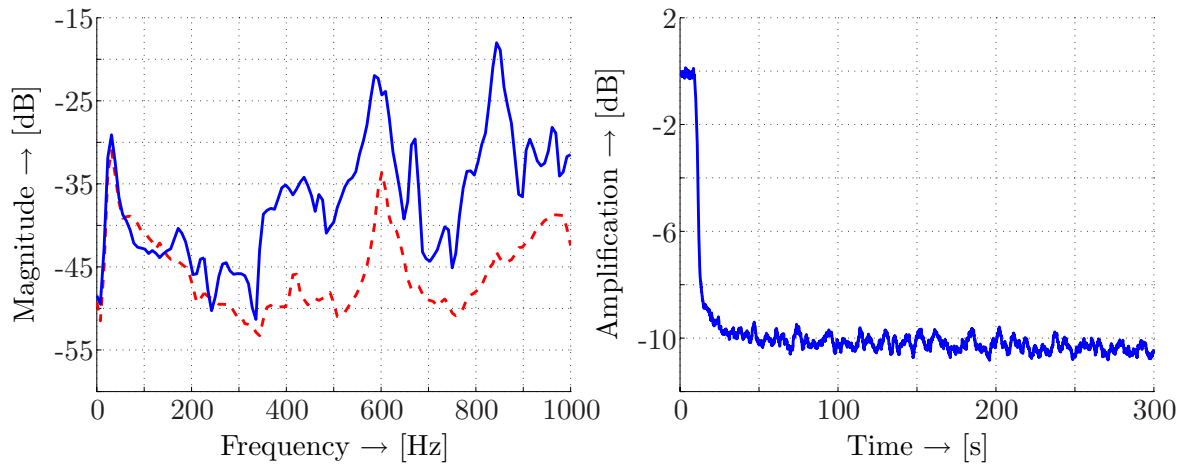


Figure 6.16: Performance of the regularized adaptive feedforward controller in simulation. Left: Spectrum of the first error sensor output with (dashed, [- -]) and without (solid, [—]) control. Right: Learning curve in the error sensor outputs.

Table 6.7: The reduction in dB within the six error sensor outputs that are obtained with the regularized adaptive feedforward controller in real-time.

Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5	Sensor 6	Average
11.3	11.4	7.3	7.1	8.6	10.5	9.4

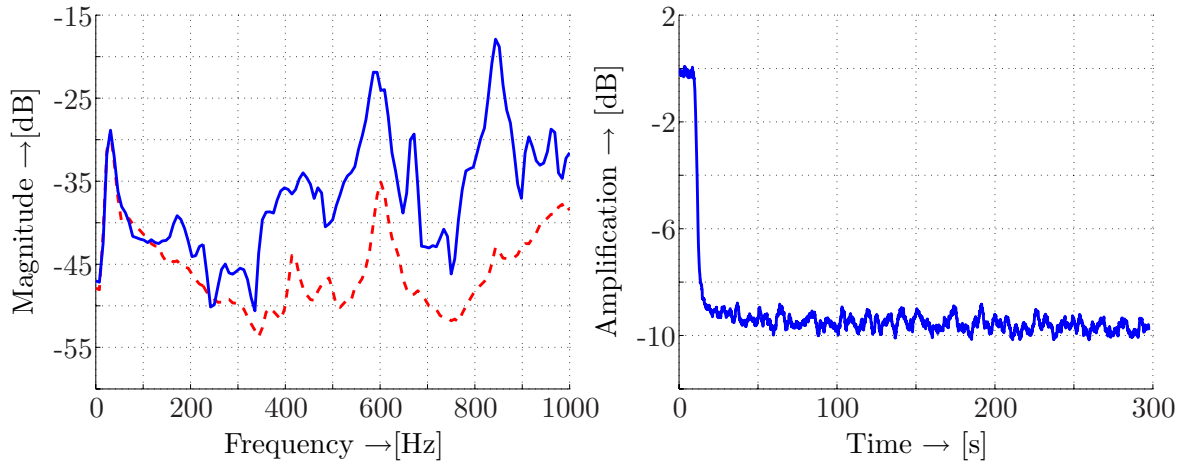


Figure 6.17: Performance of the regularized adaptive feedforward controller in real-time. Left: Spectrum of the first error sensor output with (dashed, [- -]) and without (solid, [—]) control. Right: Learning curve in the error sensor outputs.

Table 6.8: VAF values and SNR values stemming from the identification of the secondary path \mathbf{S} of the SR-MIMO vibration isolation setup (model order 58) .

Output	1	2	3	4	5	6	Average
VAF (%)	98.8	98.6	98.7	99.2	98.9	99.0	98.9
SNR (dB)	1912	18.6	18.9	20.9	19.8	20.1	19.6

both simulation and real-time. The learning curve matches the learning curve obtained from the adaptive control experiments in simulation as shown in figure 6.16 on the right. Within approximately 40 seconds the adaptive controller converges close to its optimum value.

6.3.3 Fixed gain feedback control

For the feedback control experiments a new secondary path model $\hat{\mathbf{S}}$ of order 58 is estimated, since the higher order counterpart cannot be implemented due to computational limitations on the dSPACE system. In the feedback control scenario, the controller is more complex with *six* inputs instead of one, thus increasing the computational load on the dSPACE system. The same order is used for the primary path model, since that model is not explicitly needed in the real-time experiments. In table 6.8 the VAF and SNR values are given for the 58th order model (compare with e.g. table 4.2), showing an average VAF of 98.9 % (19.6 dB SNR). The new lower order model is less accurate, especially in the lower frequency region. However, the controller is not active in the lower frequency region (due to regularization), so the consequences of this are limited.

The feedback controller is designed with regularization for the following two reasons:

1. The models for the primary and secondary path deviate from the true plants for obvious reasons. Especially in the feedback control scenario, model uncertainty is unwanted, since that could lead to instability of the feedback controller. For an explanation of

Table 6.9: *The reduction in dB within the six error sensor outputs that are obtained with the regularized fixed gain feedback controller in simulation.*

Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5	Sensor 6	Average
3.6	5.3	2.4	2.9	2.8	3.8	3.5

this, see page 44. To overcome that problem, the controller is regularized so that model uncertainties are compensated for, i.e. the controller is stabilized, similarly to the regularization applied in the feedback control experiments on the SR-SISO vibration isolation setup. Note that implementing the controller without regularization on the real-time setup leads to instability.

2. Regularization is necessary to avoid oversteering the actuators, which is a problem that is encountered in the feedforward control scenario also.

The regularization is again performed in two steps. First, a regularized outer factor is computed using equation (3.65). This yields a value of 0.01 for the regularization factor ϖ . The outer factor is inverted to obtain the inverse outer factor which is included in the control scheme. Subsequently, a regularized controller $\check{\mathbf{W}}_o$ is computed. The controller can be designed equivalently to the model-based design technique discussed in appendix B. This is not described in the appendix, though. For this, see e.g. (Elliott, 2000b). In addition to an inner/outer factorization of the secondary path \mathbf{S} , the design requires the computation of an inner/outer factorization of the primary path \mathbf{P} . However, this leads to computational problems and therefore the controller is not obtained in a model-based fashion. Instead, the fixed-gain controller is designed using the auto-correlation matrix and cross-correlation vector of the disturbance and filtered disturbance signals using equation (3.31). Specifically, it is assumed that the fixed gain controller has a FIRF structure. The controller $\check{\mathbf{W}}_o$ is regularized with a factor $\rho = 3 \cdot 10^{-3}$. Each channel in the controller is approximated by a FIRF of 60 coefficients. This means that the controller has a total of $6 \times 6 \times 60$ coefficients. More coefficients cannot be implemented on the dSPACE system. The complete fixed gain controller is given by equation (3.74). In the simulations approximately 25 dB measurement noise $\zeta(k)$ is added to the disturbance outputs $\mathbf{d}(k)$.

Table 6.9 shows the reduction obtained in simulation. An average reduction of 3.5 dB is achieved. See figure 6.18 for a depiction of the controller performance in the first error sensor output. Particularly the high frequency resonance peaks are suppressed. The feedback controller has no effect on the resonance peaks in the lower frequency region, which can be explained by the regularization that is applied. This was also observed in the feedforward control experiments. However, from the knowledge obtained from the feedback control experiments on the SR-SISO vibration isolation, it is concluded that the results are satisfactory and that the suppression of the resonance peaks is as desired.

The regularized fixed gain controller is also implemented on the experimental setup. Table 6.10 shows the obtained reduction. On average, the controller performs exactly the same as the controller in simulation. The performance of the controller on the first error sensor output is shown in figure 6.19. The real-time results closely match the simulation results and good performance is obtained.

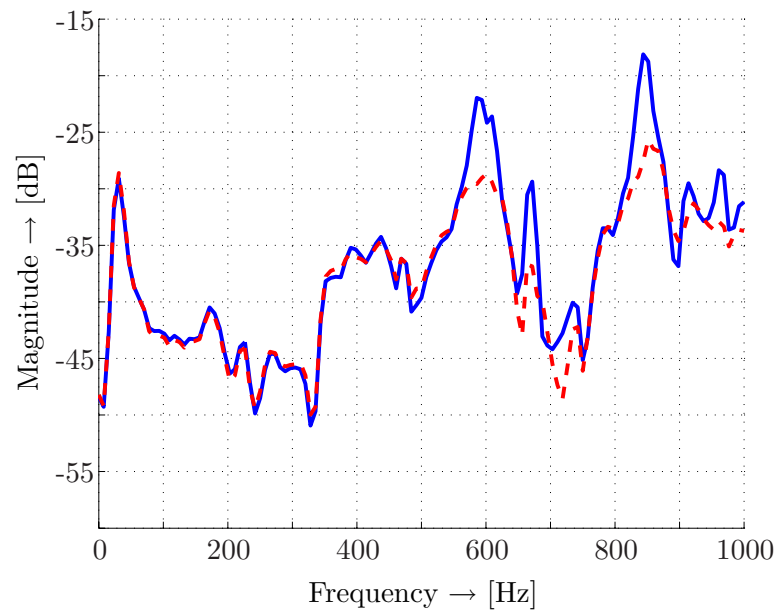


Figure 6.18: Performance of the regularized fixed gain feedback controller in simulation. Spectrum of the first error sensor output with (dashed, [- -]) and without (solid, [—]) control.

Table 6.10: The reduction in dB within the six error sensor outputs that are obtained with the regularized fixed gain feedback controller in real-time.

Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5	Sensor 6	Average
3.8	5.2	2.4	2.9	2.7	4.0	3.5

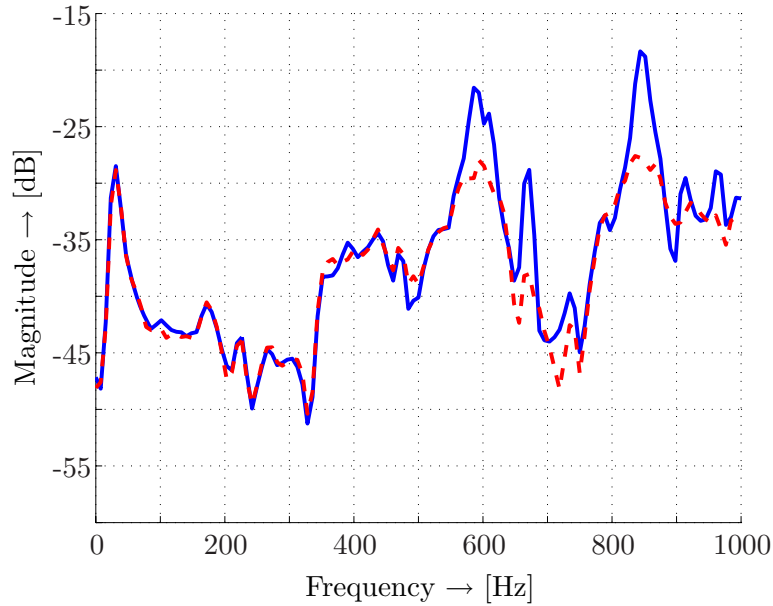


Figure 6.19: Performance of the regularized fixed gain feedback controller in real-time. Spectrum of the first error sensor output with (dashed, [- -]) and without (solid, [—]) control.

Table 6.11: The reduction in dB within the six error sensor outputs that are obtained with the regularized adaptive feedback controller in simulation.

Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5	Sensor 6	Average
3.7	5.3	2.5	2.9	2.9	3.9	3.5

6.3.4 Adaptive feedback control

The IOALMS control scheme depicted in figure 3.13 is implemented. The filter-update is given in equation (3.72). The regularization parameters are the same as those used in the fixed gain control experiments: the outer factor is regularized with $\varpi = 0.01$, while the adaptive controller is regularized with $\rho = 3 \cdot 10^{-3}$. The number of coefficients of each channel in the adaptive controller $\check{\mathbf{W}}_o(k)$ is set to $L = 60$. Thus, $6 \times 6 \times 60 = 2160$ controller coefficients are adapted at every sampling instant. The regularized adjoint inner factor model is determined to filter the error sensor outputs. Here a value of 39 is chosen for p in equation (3.70). The adaptive feedback controller is then simulated for 300 seconds. The step-size is tuned to $\mu = 0.01$. The obtained reduction is shown in table 6.11. An average reduction of 3.5 dB is established, which is identical to the reduction that is obtained with the fixed gain controller in simulation and real-time. The performance obtained with the adaptive controller on the first error sensor output is shown in figure 6.21 on the left. The resonance peaks are clearly suppressed by the adaptive controller. It can be concluded so far that the adaptive controller performs as desired. To gain more insight into the convergence, a learning curve is shown on the right in figure 6.21 which demonstrates the learning behavior of the adaptive controller. The controller is converged within approximately 30 seconds after it is switched on.

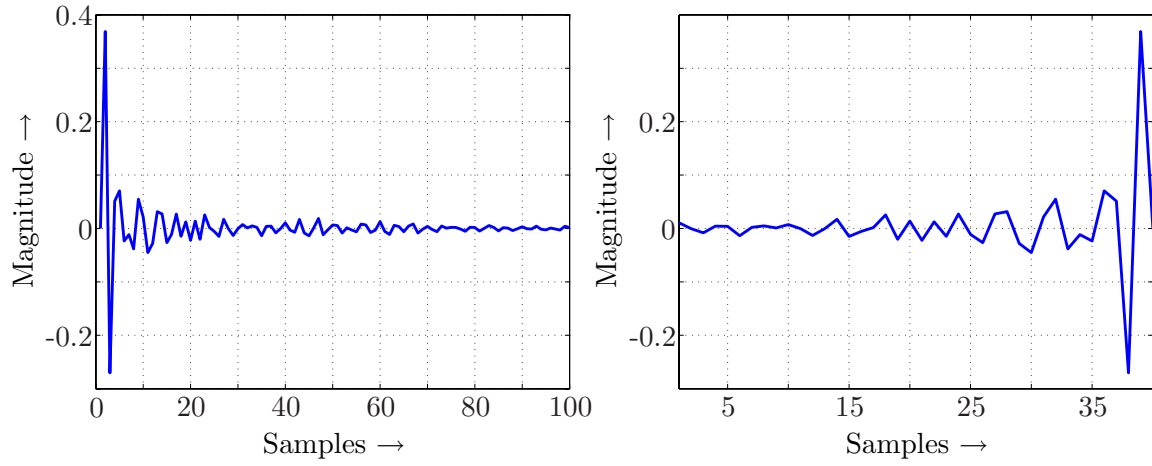


Figure 6.20: Impulse response of $\hat{\mathbf{S}}_i$ (left) and impulse response of $\hat{\mathbf{S}}_{i_{\text{FIRF}}}^*$ (right) from the first input to the first output.

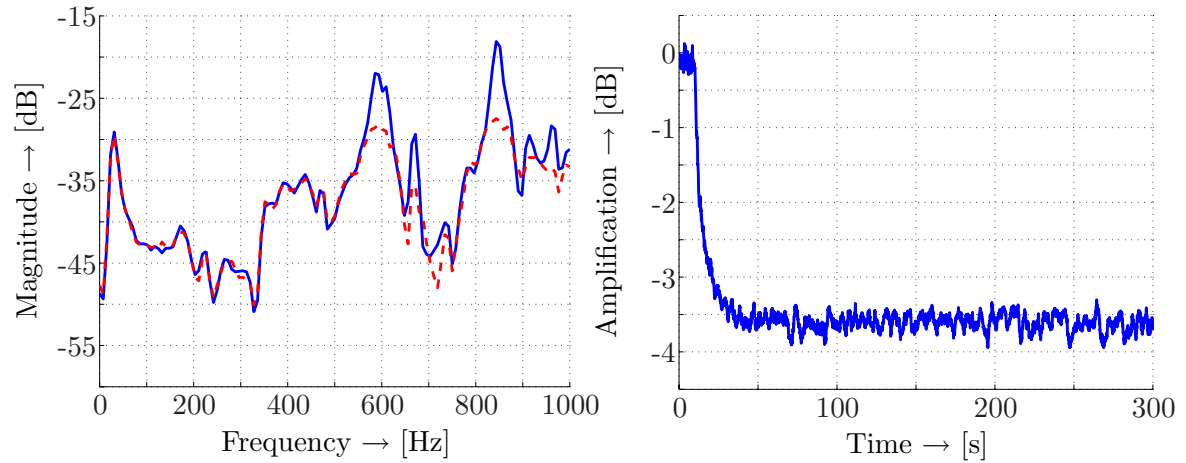


Figure 6.21: Performance of the regularized adaptive feedback controller in simulation. Left: Spectrum of the first error sensor output with (dashed, [- -]) and without (solid, [—]) control. Right: Learning curve in the error sensor outputs.

Table 6.12: *The reduction in dB within the six error sensor outputs that are obtained with the unregularized adaptive feedback controller in simulation.*

Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5	Sensor 6	Average
7.0	8.6	3.9	3.8	6.4	5.1	5.8

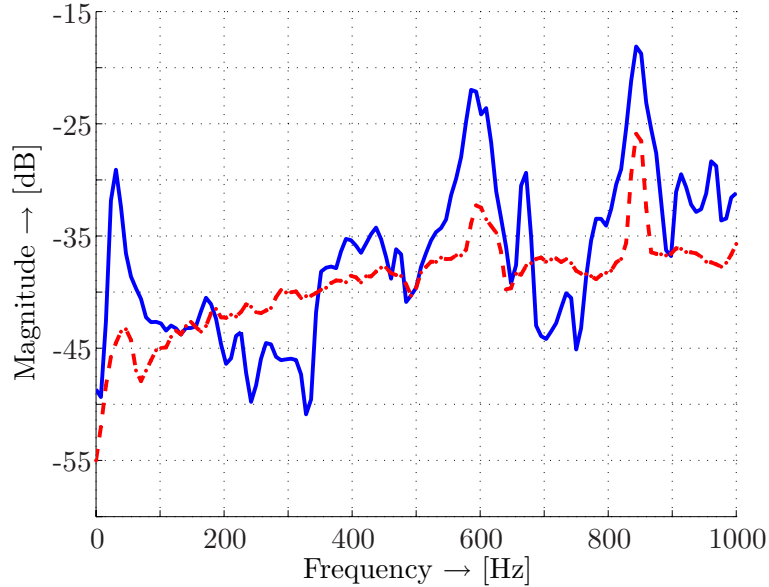


Figure 6.22: *Performance of the unregularized adaptive feedback controller in simulation. Spectrum of the first error sensor output with (dashed, [- -]) and without (solid, [—]) control.*

These results show the performance of the regularized controller, which is of limited length and based on a lower order model of the secondary path. For comparison, the performance of the unregularized controller of full length is checked also, which is based on the full order model for the secondary path. Thus, a good conclusion can be drawn concerning the performance of the ‘truncated adaptive feedback controller’. The reduction obtained with an adaptive controller of $6 \times 6 \times 200$ coefficients, a full order model for the secondary path \mathbf{S} and no regularization is shown in table 6.12. On average the reduction is 5.8 dB, which is only 2.3 dB more than for the truncated adaptive controller. The performance of the controller on the first error sensor output is shown in figure 6.22. The performance is good, particularly on the resonance peaks. In contrast to the good reduction obtained on the resonance peaks, at some points in between the resonance peaks there is amplification instead of suppression. This is a consequence of feedback control (Doyle et al., 1992). The controller yields better performance especially in the low frequency area. That is logical since there is no regularization and the models are perfect. This controller cannot be implemented in real-time, however, which is why the attention is turned again to the regularized controller which is based on a lower order model.

The adaptive controller is implemented on the real-time setup. The performance after 300 seconds is shown in table 6.13. The real-time performance is depicted in figure 6.23 on the

Table 6.13: The reduction in dB within the six error sensor outputs that are obtained with the regularized adaptive feedback controller in real-time.

Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5	Sensor 6	Average
3.7	5.3	2.4	2.9	2.8	4.0	3.5

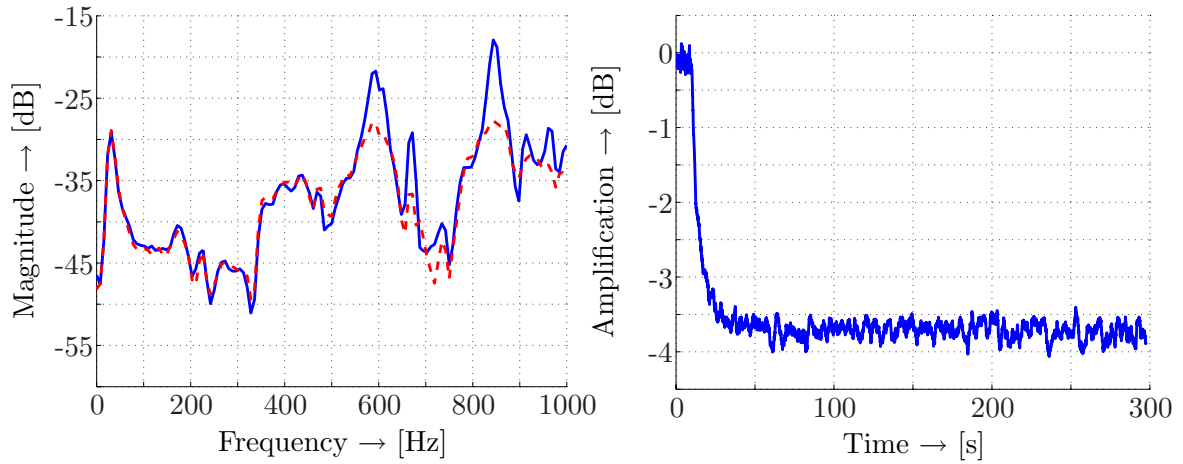


Figure 6.23: Performance of the regularized adaptive feedback controller in real-time. Left: Spectrum of the first error sensor output with (dashed, [- -]) and without (solid, [—]) control. Right: Learning curve in the error sensor outputs.

left. This figure shows the performance of the adaptive feedback controller on the first error sensor output, which is in close correspondence with the performance obtained during the real-time experiments. On the right of the figure, a learning curve is presented. The results achieved by the fixed gain controller are also similar to the results obtained by the adaptive controller.

Chapter 7

Conclusions and recommendations

7.1 Conclusions

In this thesis the design, implementation and validation of control algorithms is described for two statically determinate vibration isolation setups: a single reference input single actuator input and single error sensor output (SR-SISO) vibration isolation setup and a single reference input multiple actuator input multiple error sensor output (SR-MIMO) vibration isolation setup. The objective of these setups is to determine if the (flexible) receiver can be isolated from the (rigid) source by the statically determinate hybrid isolation mounts, so that disturbances stemming from the source are reduced at the receiver. Here, subspace model identification (SMI) is used for the identification of the transfer paths (i.e. the primary and secondary path). In particular, the past-output multi-variable output error state space SMI routine is used.

The following conclusion can be drawn from the identification experiments:

- SMI is able to identify accurate linear models of the primary and secondary paths of both setups. From the recorded input/output data contaminated with noise, only the trends and a non-zero mean value are removed. Subsequently, the number of (block) rows in the Hankel matrix are chosen and the data are presented to the SMI routine. After selecting the proper model order, linear models can be estimated in a single multivariable experiment. Based on validation data and model measuring techniques, it is demonstrated that the models for both setups are almost perfect. For all models, variance-accounted-for (VAF) values of over 99.7% are obtained. This demonstrates the potential of SMI for the identification of models for these kinds of mechatronic systems.

The following conclusions can be drawn from the control experiments:

- For the SR-SISO vibration isolation setup a feedback controller is designed, implemented and validated which is able to reject a broadband disturbance (0-2 kHz) in the error sensor output. For control design, the internal model control principle is used, which transforms the feedback control problem into an equivalent feedforward one. An adaptive controller is used and the parameters of the controller are updated with an adjoint

least mean square type of algorithm. To speed up the convergence of the controller parameters the inverse outer factor model is included in the control scheme. To be able to implement the controller and to obtain a stable closed loop, a small regularization factor is included in the adaptive controller, preventing instability. The feedback controller is able to damp the four main resonance peaks of the setup. An average reduction in the error sensor output of 5.1 dB is achieved in simulation and of 5.0 dB in real-time. Simulation and real-time results concur. This indicates that the identified models can be used for vibration isolation control.

- On the SR-MIMO vibration isolation setup, two types of feedforward adaptive controllers are designed, implemented and validated for suppression of narrowband disturbances. Specifically, the filtered-x least mean squares algorithm and the principle component least mean square algorithm are used to drive the coefficients of the adaptive controller. Both algorithms are able to reject a disturbance output that contains one frequency (528 Hz or 565 Hz) and two frequencies (496 Hz and 660 Hz). In simulation and real-time, the disturbances with the particular frequencies are suppressed equally well and the real-time results are in close correlation with the simulation results. Given the same step-size in both algorithms, the filtered-x least mean square algorithm requires a longer convergence time. This is due to a larger eigenvalue-spread in the secondary path at the frequency under consideration.
- Based on the identified models of the primary and secondary paths of the SR-MIMO vibration isolation setup a feedforward fixed gain and an adaptive controller are designed, implemented and validated for suppression of broadband disturbances between 0 and 1 kHz. The fixed gain controller is designed in state space using the feedforward model based Wiener controller. To enable implementation of the controller, controller order reduction is performed using SMI (by performing an identification experiment on the controller). It is demonstrated that the lower order controller is nearly as accurate as its full order counterpart and that it can be implemented successfully, and practically without loss of performance. Both the fixed gain controller and the adaptive controller show a similar performance in simulation and real-time. Both controllers are successfully able to reject broadband disturbances in the error sensor output. The adaptive controller is updated with an adjoint least mean square type of algorithm. To prevent over-actuation, regularization is applied, thus robustifying the controller. In simulation, the fixed gain controller achieves an average reduction of 10.5 dB in the error sensor outputs. In real-time, this is 9.0 dB. The regularized adaptive controller yields a reduction of 10.3 dB in simulation and 9.4 dB in real-time. Thus, the adaptive controller performs better in real-time than the fixed gain controller. This can be explained by the fact that the adaptive controller does not require information about the primary path and is able to adapt to small changes. To increase the adaptation, the state space based inverse outer factor model is included in the control scheme. This leads to convergence within a few seconds.
- Finally, a fixed gain and adaptive feedback controller are designed, implemented and validated for suppression of broadband disturbances (0-1 kHz) using the internal model control principle. The fixed gain controller and the adaptive controller perform similarly in simulation and real-time. Instead of designing the fixed gain controller in state space, it is designed as a finite impulse response filter. Both controllers are able to reject

broadband disturbances in the error sensor outputs. The main resonance peaks of the setup are suppressed and an average reduction of 3.5 dB is achieved in the error sensor outputs in simulation and real-time, both with the fixed gain and the adaptive controller. To speed up the convergence of the adaptive controller, the state space inverse outer factor model is included in the control scheme. The adaptive controller converges within seconds. To prevent over-actuation and to stabilize the closed loop, regularization is applied.

7.2 Recommendations

The following recommendations are made:

- The SR-SISO and the SR-MIMO vibration isolation setup are equipped with piezo-electric actuators. Although identification and control were successfully applied, this success was for a large part due to the fact that the voltages to the piezo-electric elements were limited, so that a large hysteresis cycle could be avoided. In the case of nonlinear behavior, SMI can no longer be applied. To be able to increase the voltages one might consider combating the nonlinear behavior of the piezo-electric elements by using e.g. a Preisach model of the hysteresis cycle.
- In this thesis the systems are assumed to be linear time invariant. In some particular cases the system may slowly change over time. It would be interesting to implement the SMI routine online, so that models can be estimated online.
- Regularization is used to robustify the controller and to prevent actuator saturation. It is interesting to develop a procedure for automatic tuning of the regularization parameters via optimization, since this is an important aspect of controller design.

Appendix A

Filter structures

A.1 Introduction

In this appendix the FIR filter (FIRF) structure, the IIR filter (IIRF) structure and the SS filter (SSF) structure are explained. Especially the FIRF structure and the SSF structure are important to understand, since these are used for control and identification purposes in this thesis. At the same time when introducing the filter structures, there is elaborated on the notation which is used in this appendix in particular and this thesis in general.

This appendix is organized as follows. In section (A.2) the difference is explained between the z -transform variable and the unit delay operator q^{-1} . In section A.3 FIRF structures are explained, while in section A.4 IIRF structures are introduced. SSF structures are explained in section A.5. If a more detailed analysis on the topic of filters and filter structures is required, the user is referred to the textbooks (Elliott, 2000*b*; Ljung, 1999; Ljung and Glad, 1994; Roberts and Mullis, 1987; Enden, 1989; Oppenheim and Shafer, 1975).

A.2 The z -transform variable and the unit delay operator q^{-1}

In the signal processing literature the unit delay operator is very often denoted by z^{-1} (Elliott, 2000*b*). This convention is widely adopted, even though z is also used as a complex variable of the z -transform (Kwakernaak and Sivan, 1991) and thus the use of z^{-1} as the unit delay operator is not correct. In the control and identification literature on the other hand, the unit delay operator is often denoted by the symbol q^{-1} (Ljung and Glad, 1994; Åström and Wittenmark, 1997), or sometimes by the symbol \mathcal{D}^{-1} . For consistency, in this thesis the unit delay operator will also be denoted by q^{-1} . Using q^{-1} as the unit delay operator does not mean that the variable z is not used. On the contrary, also the variable z is used, but this indicates a z -transformed quantity in the frequency domain. As such, z does not have any relation with unit delay operator. So, q^{-1} indicates an operation in the time domain, while z indicates an operation in the frequency domain. Let us now clarify the unit delay operator q^{-1} . If the unit delay operator operates on the sequence $\{u(k)\}_{k=1}^N$, with N the number of samples of the sequence, it transforms the sequence a sample delayed (Fuller et al., 1996).

$$q^{-1} \{u(k)\}_{k=2}^N = \{u(k)\}_{k=1}^{N-1},$$

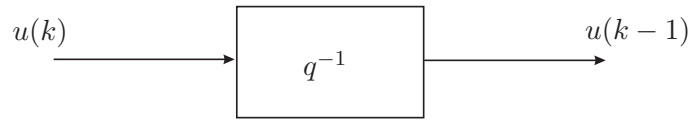


Figure A.1: The use of q^{-1} as a unit delay operator.



Figure A.2: A SISO system \mathbf{R}_{FIR} with input $u(k)$ and output $y(k)$.

where ‘ k ’ is the sample number, normalized¹ with the sample time T . This is graphically depicted in figure A.1.

A.3 Finite impulse response filter structure

Take a look figure A.2 in which a single input single output (SISO) filter \mathbf{R}_{FIR} is depicted. $u(k)$ is the scalar input and $y(k)$ is the scalar output. Suppose that the output $y(k)$ is a linear combination of the weighted current value and weighted past values of the input $u(k)$ as follows:

$$y(k) = r_0 u(k) + r_1 u(k-1) + r_2 u(k-2) + \cdots + r_{L-1} u(k-L+1), \quad (\text{A.1})$$

where $r_l \in \mathbb{R}$, ($l = 0, \dots, L-1$), are scalar coefficients (or weights, or taps as they are also often called). By using the unit delay operator q^{-1} , $y(k)$ can be written in shorthand notation as:

$$y(k) = \mathbf{R}_{\text{FIR}}(q^{-1})u(k), \quad (\text{A.2})$$

where $\mathbf{R}_{\text{FIR}}(q^{-1}) = \left(\sum_{l=1}^L r_{l-1} q^{-l+1} \right)$. As such, $\mathbf{R}_{\text{FIR}}(q^{-1})$ is an operator which operates on the input $u(k)$ as in equation (A.1) and which results in the output $y(k)$. The input/output relationship in equation (A.2) is graphically depicted in figure A.3. Since the output $y(k)$ is a linear combination of the weighted current value and the weighted previous values of the input $u(k)$ only, the system \mathbf{R}_{FIR} has a finite impulse response structure and thus is a FIRF. The name FIR comes from the fact that every *finite* input sequence which is input to the system \mathbf{R}_{FIR} leads to a *finite* output sequence at the output of the filter. FIRFs are also denoted as *moving average* filters, *transversal* filters, or *non-recursive* filters. The following is a particular property of a FIRF: if the input $u(k)$ is a Kronecker delta function $\delta(k)$ (Elliott, 2000b) (see also figure A.4):

$$\delta(k) = \begin{cases} 1, & k = 0 \\ 0, & k \neq 0 \end{cases},$$

then the output $\{y(k)\}_{k=0}^{L-1}$ is identical to the coefficients r_l , ($l = 0, \dots, L-1$). That is

¹Normalization means that the notation kT has to be used, instead of only k , but that kT is abbreviated to simply k . As such k is the normalized sample number.

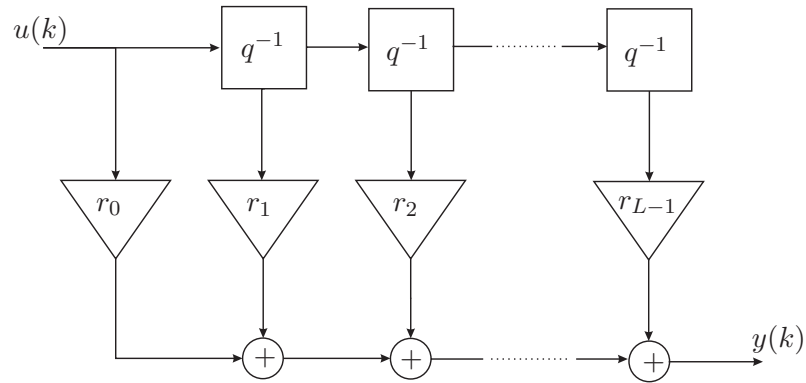


Figure A.3: The output $y(k)$ is a linear combination of the weighted current value and weighted previous values of the input $u(k)$.

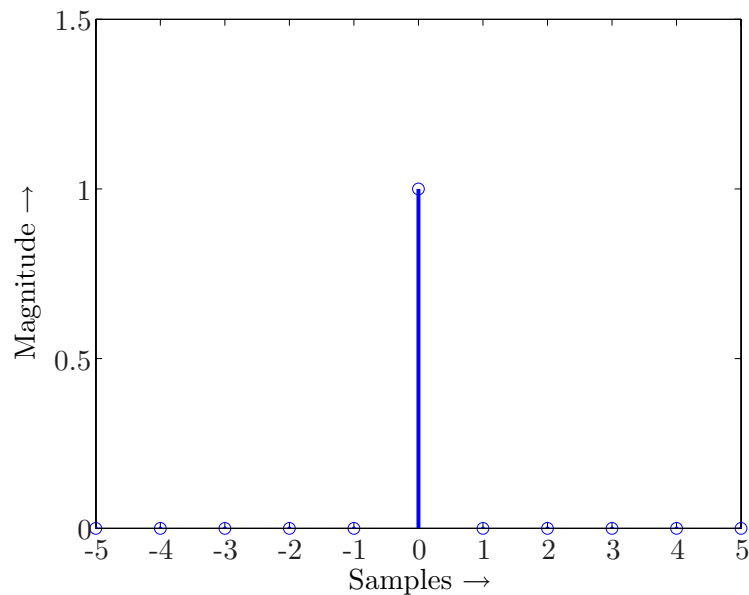


Figure A.4: Kronecker delta function $\delta(k)$ displayed here on a limited x-axis between $k = -5$ and $k = 5$. Recall that the Kronecker delta function $\delta(k)$ is zero $\forall k \neq 0$.

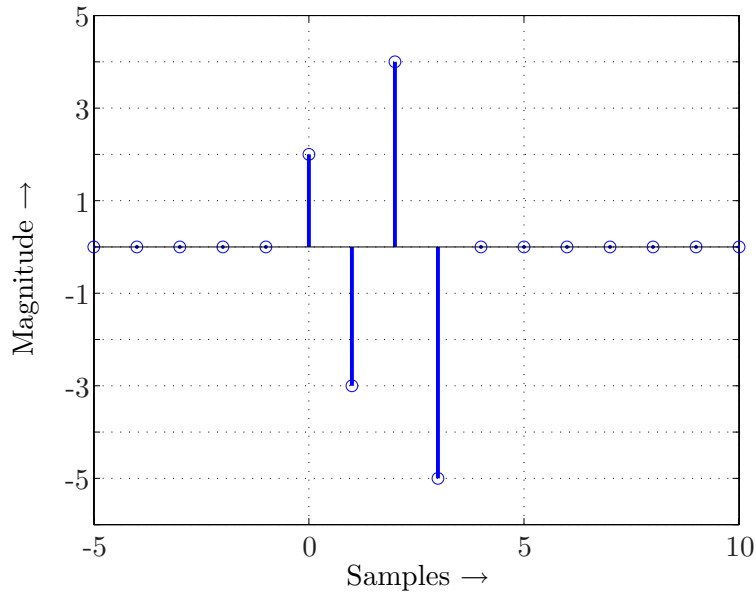


Figure A.5: Impulse response of the system in equation (A.3). For clarity, also the impulse response for samples ($k = -5, \dots, -1$) is included, since the Matlab function `impulse` only gives back the impulse response from $k = 0$.

illustrated by an example.

Example A.3.1 (Finite impulse response filter). Assume the following FIRF:

$$\mathbf{R}_{FIR}(q^{-1}) = 2 - 3q^{-1} + 4q^{-2} - 5q^{-3} \quad (\text{A.3})$$

The coefficients of the FIRF are $r_0 = 2$, $r_1 = -3$, $r_2 = 4$, $r_3 = -5$. This FIRF can easily be realized in Matlab with the command `R = tf([2 -3 4 -5], [1 0 0 0], 1)`, where it is assumed that the system is sampled at a sampling frequency of 1 Hz. A ten second impulse response for this system can be constructed with the command `impulse(R, 10)`. The impulse response is the output of the system \mathbf{R}_{FIR} with the input as given in figure A.4. The impulse response is given in figure A.5. From the figure it is clear that the values of the impulse response are identical to the coefficients of the FIRF (i.e. $r_0 = 2$, $r_1 = -3$, $r_2 = 4$, $r_3 = -5$). For $0 > k \geq 5$ the impulse response equals zero.

Up to now, the explanation is based on the time domain. It is interesting to move to the frequency domain to explain the z -transform of the FIRF. In the frequency domain, the output $\mathbf{Y}(z)$ can be written as:

$$\mathbf{Y}(z) = \mathbf{R}_{FIR}(z)\mathbf{U}(z),$$

where $\mathbf{Y}(z)$ is the z -transform of the output $y(k)$, $\mathbf{U}(z)$ is the z -transform of the input $u(k)$ and $\mathbf{R}_{FIR}(z)$ is the z -transform of the FIRF. $\mathbf{R}_{FIR}(z)$ is also called the transfer function of the system \mathbf{R}_{FIR} , since it relates the z -transform of the input sequence to the z -transform of the output sequence. $\mathbf{R}_{FIR}(z)$ is given by:

$$\mathbf{R}_{FIR}(z) = \sum_{l=1}^L r_{l-1}z^{-l+1} = r_0 + r_1z^{-1} + r_2z^{-2} + \dots + r_{L-1}z^{-L+1}. \quad (\text{A.4})$$

Note that equation (A.2) and equation (A.4) are similar with the exception of the symbols q and z and the power of the symbols on the left hand side of both equations. Indeed, as far as notation is concerned, in the time domain the operator $\mathbf{R}_{\text{FIR}}(q^{-1})$ can easily be derived from the frequency domain transfer function $\mathbf{R}_{\text{FIR}}(z)$, by substituting q for z on the right hand side of both equations and q^{-1} for z on the left hand side of both equations. However, there is an important difference which lies in the interpretation between equation (A.2) and equation (A.4). That is that equation (A.4) is defined in the frequency domain and as such, $\mathbf{R}_{\text{FIR}}(z)$ is not operator which can act on a time domain input $u(k)$. Equation A.4 can also be denoted by a polynomial in z . Specifically, equation (A.4) can be rewritten as:

$$\begin{aligned}
\mathbf{R}_{\text{FIR}}(z) &= \frac{h_0 + h_1 z^{-1} + h_2 z^{-2} + \cdots + h_{L-1} z^{-L+1}}{1} \\
&= \frac{h_0 z^{L-1} + h_1 z^{L-2} + h_2 z^{L-3} + \cdots + h_{L-1}}{z^{L-1}} \\
&= h_0 \frac{z^{L-1} + \frac{h_1}{h_0} z^{L-2} + \frac{h_2}{h_0} z^{L-3} + \cdots + \frac{h_{L-1}}{h_0} z^{L-1}}{z^{L-1}} \\
&= h_0 \frac{(z - \kappa_1)(z - \kappa_2) \cdots (z - \kappa_{L-1})}{(z)(z) \cdots (z)} \\
&= h_0 \frac{\prod_{i=1}^{L-1} (z - \kappa_i)}{\prod_{i=1}^{L-1} (z)},
\end{aligned} \tag{A.5}$$

where $\prod_{i=1}^{L-1}$ is the product operator. Which now is a transfer function with a numerator and a denominator. $\mathbf{R}_{\text{FIR}}(z)$ characterizes itself in that it has $L - 1$ zeros κ_i which can lie at a certain location in the complex plan (inside, on, or outside the unit circle). Furthermore, $\mathbf{R}_{\text{FIR}}(z)$ has $L - 1$ repeated poles in the origin. Although a FIRF is commonly termed as an all-zero filter as it seems from equation (A.2), this is somewhat misleading, since \mathbf{R}_{FIR} has also $L - 1$ repeated poles in the origin which is particularly clear from the third expression in equation (A.5). $\mathbf{R}_{\text{FIR}}(z)$ is termed minimum phase if all zeros κ_i are strictly smaller than unity ($|\kappa_i| < 1$). Otherwise the system is called non-minimum phase. Up to this point, the parameters r_l of the FIRF are fixed i.e. the parameters did not depend on the sampling instant k . Later on in this thesis there will be made use of a FIRF as an adaptive controller. Then the parameters of the FIRF depend on the sampling instant k . For example, a SISO FIRF which is dependent on the sampling instant k is denoted as:

$$\mathbf{R}_{\text{FIR}}(k, q^{-1}) = \sum_{l=1}^L r_{l-1}(k) q^{-l+1} = r_0(k) + r_1(k) q^{-1} + r_2(k) q^{-2} + \cdots + r_{L-1}(k) q^{-L+1}, \tag{A.6}$$

with $r_l(k)$, ($l = 0, \dots, L - 1$), the coefficients of the FIRF, which are now dependent on the normalized sampling instant k and thus can vary in time.



Figure A.6: A SISO system \mathbf{R}_{IIR} with input $u(k)$ and output $y(k)$.

A.4 Infinite impulse response filter structure

Take a look figure A.6. Suppose the output $y(k)$ of the system \mathbf{R}_{IIR} is a linear combination of weighted current and weighted past values of the output $u(k)$ and also of weighted previous values of the output $y(k)$:

$$y(k) = a_1 y(k-1) + \dots + a_N y(k-N) + h_0 u(k) + h_1 u(k-1) + \dots + h_{L-1} u(k-L+1),$$

where $a_n \in \mathbb{R}$, ($l = 0, \dots, L-1$) and $h_l \in \mathbb{R}$, ($n = 1, \dots, N$), are scalar coefficients. N is denoted as the order of the system. By using the unit delay operator, it is possible to write $y(k)$ in shorthand notation as:

$$y(k) = \sum_{n=1}^N a_n y(k-n) + \sum_{l=0}^{L-1} h_{l-1} u(k).$$

This again can be written in operator notation as:

$$y(k) = \mathbf{R}_{\text{IIR}}(q^{-1})u(k), \quad (\text{A.7})$$

where $\mathbf{R}_{\text{IIR}}(q^{-1}) = \left(\sum_{l=0}^{L-1} h_{l-1} q^{-l+1} \right) / \left(1 + \sum_{n=1}^N a_n q^{-n} \right)$. The input/output relationship as in equation (A.7) is graphically depicted in figure A.7. This figure can be compared directly with figure A.3. From figure A.7 the recursions are clear in the output $y(k)$. On the contrary to the FIRF, the filter \mathbf{R}_{IIR} now has a so called *infinite* impulse response structure and is called an IIRF. The reason for this is clear: in case of an IIRF, every finite input sequence results in an infinite output sequence. These IIRFs are also denoted as *pole/zero* filters, *recursive* filters, or *auto-regressive moving average* filters (Elliott, 2000b). In the frequency domain, the output $\mathbf{Y}(z)$ can be written as:

$$\mathbf{Y}(z) = \frac{\mathbf{H}(z)}{\mathbf{A}(z)} \mathbf{U}(z) = \mathbf{R}_{\text{IIR}}(z) \mathbf{U}(z),$$

where $\mathbf{Y}(z)$ is again the z -transform of the output $y(k)$, $\mathbf{U}(z)$ the z -transform of the input $u(k)$ and $\mathbf{R}_{\text{IIR}}(z)$ is the z -transform of the IIRF, which is given by:

$$\mathbf{R}_{\text{IIR}}(z) = \frac{\sum_{l=1}^L h_{l-1} z^{-l+1}}{1 + \sum_{n=1}^N a_n z^{-n}}. \quad (\text{A.8})$$

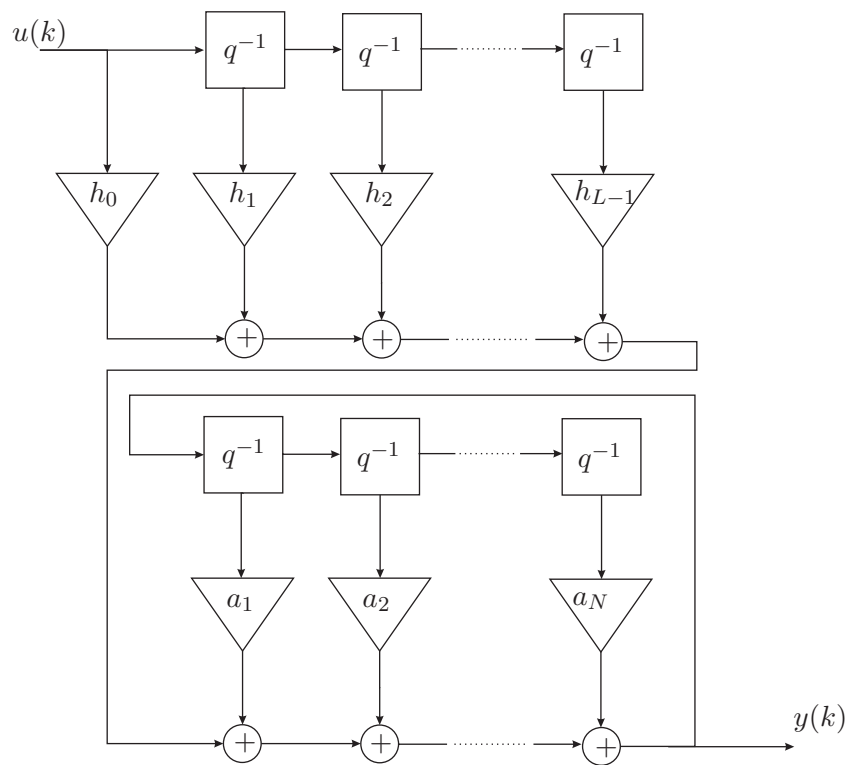


Figure A.7: The output $y(k)$ is a linear combination of the weighted current value and weighted previous values of the input $u(k)$ and also of weighted previous values of the output $y(k)$.

Equation (A.8) can also be denoted by a polynomial in z with a numerator and a denominator. Specifically, equation (A.8) can be rewritten as follows:

$$\begin{aligned}
 \mathbf{R}_{\text{IIR}}(z) &= \frac{h_0 z^{L-1} + h_1 z^{L-2} + h_2 z^{L-3} + \cdots + h_{L-1} z^{N-L+1}}{z^N + a_1 z^{N-1} + a_2 z^{N-2} + \cdots + a_N} \\
 &= h_0 \frac{z^{L-1} + \frac{h_1}{h_0} z^{L-2} + \frac{h_2}{h_0} z^{L-3} + \cdots + \frac{h_{L-1}}{h_0}}{z^N + a_1 z^{N-1} + a_2 z^{N-2} + \cdots + a_N} z^{N-L+1} \\
 &= h_0 \frac{(z - \kappa_1)(z - \kappa_2) \cdots (z - \kappa_{L-1})}{(z - \lambda_1)(z - \lambda_2) \cdots (z - \lambda_N)} z^{N-L+1} \quad . \quad (\text{A.9}) \\
 &= h_0 \frac{\prod_{i=1}^{L-1} (z - \kappa_i)}{\prod_{i=1}^N (z - \lambda_i)} z^{N-L+1}
 \end{aligned}$$

However, $\mathbf{R}_{\text{IIR}}(z)$ now characterizes itself by the fact that it has $L - 1$ zeros κ_i (just like the FIRF) but also N poles λ_i which are not bounded to the origin, but can be everywhere in the complex z -plane. Two important observations can be made related to equation (A.9):

- For stability of the system \mathbf{R}_{IIR} , the absolute values of the poles λ_i need not be larger than unity: $|\lambda_i| \leq 1$. If all poles are strictly smaller than unity, i.e. $|\lambda_i| < 1$, the system is said to be strictly stable, or asymptotically stable. If some poles are identical to unity, i.e. $|\lambda_i| = 1$, the system \mathbf{R}_{IIR} is said to be marginally stable.
- If the number of poles N is equal or larger than the number of zeros $L - 1$: $N \geq L - 1$ (including poles and zeros in the origin), the filter \mathbf{R}_{IIR} is said to be causal, which means that a current value of the output does not depend on future values of the input. That is illustrated by an example later. Note that the filter is called proper when $N \geq L - 1$ and strictly proper when $N > L - 1$. If the number of poles N is smaller than the number of zeros $L - 1$: $N < L - 1$ (including poles and zeros in the origin), the filter \mathbf{R}_{IIR} is said to be anti-causal. As such, the current system output then depends on future values of the input.

Example A.4.1 (Causal/anti-causal system). *Assume the following IIRF with a single zero and a single pole:*

$$\mathbf{R}_{\text{IIR}}(q^{-1}) = \frac{2 - 3q^{-1}}{1 - .5q^{-1}}. \quad (\text{A.10})$$

By using equation (A.10) it can be written that:

$$y(k) = 0.5y(k-1) + 2u(k) - 3u(k-1).$$

The output $y(k)$ only depends on the weighted previous values of the output $y(k)$ and the weighted current and weighted previous value of the input $u(k)$. Therefore, the system is said to be causal: the output $y(k)$ is not dependent on future inputs. Now suppose the system is given by:

$$\mathbf{R}_{\text{IIR}}(q^{-1}) = \frac{2 - 3q^{-1}}{q^{-1}}, \quad (\text{A.11})$$

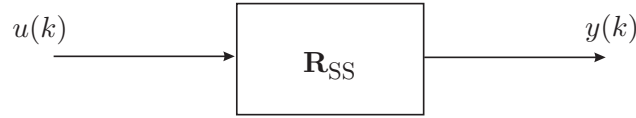


Figure A.8: A SISO system \mathbf{R}_{SS} with input $u(k)$ and output $y(k)$.

which has a single zero, but no poles. The fact that the system has no poles can be seen if both the numerator and the denominator are multiplied by q . This results in $R_{IIR}(q^{-1}) = \{2q - 3\}$ and thus the system has only a numerator. By using (A.11) it can also be written that:

$$y(k - 1) = 2u(k) - 3u(k - 1).$$

Shifting both the output on left hand side and the input on the right hand side by one sample (this does not change the input/output relationship), gives:

$$y(k) = 2u(k + 1) - 3u(k).$$

From the latter equation it can be seen that the output $y(k)$ depends on the weighted current input $u(k)$ and the weighted future input $u(k + 1)$. The system is now said to be anti-causal. Such a system can generally not be implemented in real-time (Nijssse et al., 2001a).

A.5 State space filter structures

Take a look figure A.8. In a filter with a state space structure the relationship between the input $u(k)$ and output $y(k)$ is written as a system of first order difference equations using a so called state vector $\mathbf{z}(k)$ (Kwakernaak and Sivan, 1991; Ljung, 1999; Phillips and Nagle, 1995; Rugh, 1996)). The state vector reflects the state of the system (Ljung, 1999):

$$\begin{aligned} \mathbf{z}(k + 1) &= \mathbf{A}\mathbf{z}(k) + \mathbf{b}u(k) \\ y(k) &= \mathbf{c}\mathbf{z}(k) + \mathbf{d}u(k) \end{aligned} \quad (\text{A.12})$$

where $\mathbf{A} \in \mathbb{R}^{N \times N}$, column vector $\mathbf{b} \in \mathbb{R}^N$, row vector $\mathbf{c} \in \mathbb{R}^N$ and scalar \mathbf{d} . $\mathbf{z}(k) \in \mathbb{R}^N$ is the state of the system. Note that N is called the state dimension, but very often it is also called the order. Later on it will be observed that N here is directly related to the N in case of the IIR filter. Furthermore note that in the SISO case, the SSF is defined in the time domain using a matrix \mathbf{A} , two vectors \mathbf{b} , \mathbf{c} and a single scalar \mathbf{d} . In the MIMO case, the vectors and the scalar will also become matrices. Although the SSF is defined in the time-domain, it can not be expressed immediately in operator notation using the unit delay operator q^{-1} . Before the SSF is written in operator notation, first the relationship between has to be made clear between the SSF and the transfer function of the SSF. If the transfer function of the SSF is known, the operator in the time domain can immediately be derived from it. So, let us now derive the transfer function of the SSF. In the frequency domain, it is true that:

$$\begin{aligned} (z\mathbf{I} - \mathbf{A})\mathbf{Z}(z) &= \mathbf{b}\mathbf{U}(z) \\ \mathbf{Y}(z) &= \mathbf{c}\mathbf{Z}(z) + \mathbf{d}\mathbf{U}(z) \end{aligned} \quad (\text{A.13})$$

By substituting the upper expression for the z transformed state $\mathbf{z}(k)$ i.e. $\mathbf{Z}(z)$, which is:

$$\mathbf{P}(z) = (z\mathbf{I} - \mathbf{A})^{-1} \mathbf{b}\mathbf{U}(z), \quad (\text{A.14})$$

in the lower expression for $\mathbf{Y}(z)$ in equation (A.13), the transfer function can be derived. So, first the expression in equation (A.14) has to be substituted in the lower expression in equation (A.13):

$$\mathbf{Y}(z) = \mathbf{c}(z\mathbf{I} - \mathbf{A})^{-1} \mathbf{b}\mathbf{U}(z) + \mathbf{d}\mathbf{U}(z).$$

Next, both the left hand side and the right hand side of the latter equation are divided by $\mathbf{U}(z)$, which gives the transfer function:

$$\mathbf{R}_{\text{SS}}(z) = \mathbf{c}(z\mathbf{I} - \mathbf{A})^{-1} \mathbf{b} + \mathbf{d}. \quad (\text{A.15})$$

Equation (A.15) is the transfer function of the SSF which can directly be compared with the transfer function of the IIRF in equation A.8. This makes clear the relationship between the SSF and the IIRF. This does not mean that a SSF and IIRF are identical. It is only shown that it is possible to derive the transfer function from a SSF. The poles λ_i of the SSF are equal to eigenvalues of the matrix \mathbf{A} . The same stability conditions apply concerning the the eigenvalues as with the IIRF. Let us now move on to the time domain. The expression for $\mathbf{R}_{\text{SS}}(z)$ in equation (A.15) can also be written as:

$$\mathbf{H}(z) = \mathbf{c}z^{-1} (\mathbf{I} - z^{-1}\mathbf{A})^{-1} \mathbf{b} + \mathbf{d}, \quad (\text{A.16})$$

where the z -transform variable is manipulated on the right hand side. The expression in equation (A.15) and the equation in (A.16) are identical. The time domain operator $\mathbf{R}(q^{-1})$ can be derived by substituting q^{-1} for z on the right hand side and substituting q for z on the left hand side of the expression in equation (A.16).

$$\mathbf{R}(q^{-1}) = \mathbf{c}q^{-1} (\mathbf{I} - q^{-1}\mathbf{A})^{-1} \mathbf{b} + \mathbf{d}.$$

and now it can be written:

$$y(k) = \left\{ \mathbf{c}q^{-1} (\mathbf{I} - q^{-1}\mathbf{A})^{-1} \mathbf{b} + \mathbf{d} \right\} u(k) \quad (\text{A.17})$$

The input/output relationship as given in equation (A.17) is graphically depicted in figure A.9. Let us clarify the discussion by a simple example.

Example A.5.1 (Relationship SSF and IIRF). *Assume the following first order SSF:*

$$\begin{aligned} p(k+1) &= 0.5p(k) + u(k) \\ y(k) &= -2p(k) + 2u(k) \end{aligned}$$

From equation (A.15) it can be deduced that the transfer function of this system is given by:

$$\mathbf{R}(z) = \frac{-2}{z - .5} + 2$$

This transfer function is equal to:

$$\begin{aligned} \mathbf{R}(z) &= \frac{-2}{z - .5} + 2 \\ &= \frac{-2}{z - .5} + 2 \frac{z - .5}{z - .5} \\ &= \frac{2z - 3}{z - .5} \end{aligned} \quad (\text{A.18})$$

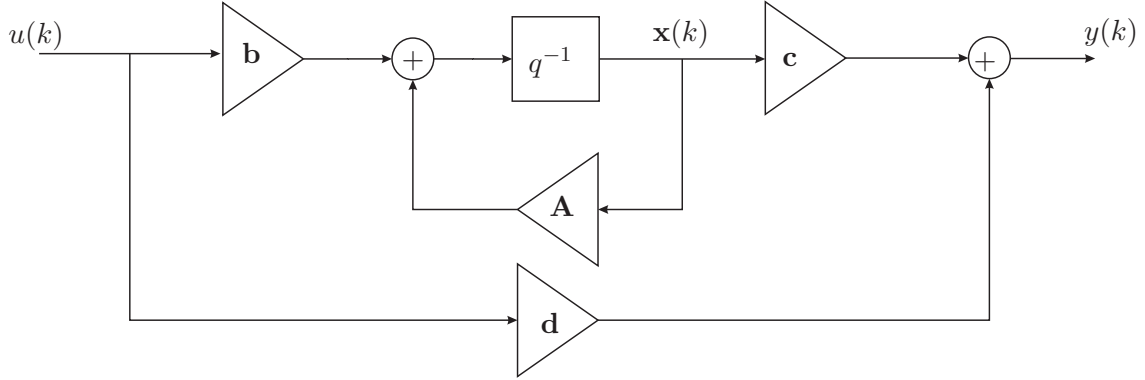


Figure A.9: The relationship between the input $u(k)$ and the output $y(k)$ in case of a SSF.

By dividing both numerator and the denominator of the last expression in equation (A.18) on the right hand side by z this can be rewritten to:

$$\mathbf{R}(z) = \frac{2 - 3z^{-1}}{1 - .5z^{-1}},$$

and this is exactly what is stated in equation (A.16). In the time-domain it can be written written:

$$\mathbf{R}(q^{-1}) = \frac{2 - 3q^{-1}}{1 - .5q^{-1}},$$

and this is the operator as defined in equation (A.11). So from the state space filter the IIR filter is derived.

That can be taken one step further to establish the relationship between the SSF and the FIRF. From the transfer function in equation (A.16) a mathematical series expansion can be made of the part $(\mathbf{I} - z^{-1}\mathbf{A})^{-1}$. The series expansion can be derived to be (Oppenheim and Shafer, 1975; Kwakernaak and Sivan, 1991):

$$\begin{aligned} (\mathbf{I} - z^{-1}\mathbf{A})^{-1} &= \frac{1}{(\mathbf{I} - z^{-1}\mathbf{A})} \\ &= 1 + \mathbf{A}z^{-1} + \mathbf{A}^2z^{-2} + \mathbf{A}^3z^{-3} + \dots \end{aligned}$$

The third expression can now be substituted for $(\mathbf{I} - z^{-1}\mathbf{A})^{-1}$ in equation A.16. This gives:

$$\begin{aligned} \mathbf{H}(z) &= \mathbf{c}z^{-1} (1 + \mathbf{A}z^{-1} + \mathbf{A}^2z^{-2} + \mathbf{A}^3z^{-3} + \dots) \mathbf{b} + \mathbf{d} \\ &= \mathbf{c}z^{-1} (\mathbf{b} + \mathbf{A}\mathbf{b}z^{-1} + \mathbf{A}^2\mathbf{b}z^{-2} + \mathbf{A}^3\mathbf{b}z^{-3} + \dots) + \mathbf{d} \\ &= \mathbf{d} + (\mathbf{c}\mathbf{b}z^{-1} + \mathbf{c}z^{-1}\mathbf{A}\mathbf{b}z^{-1} + \mathbf{c}z^{-1}\mathbf{A}^2\mathbf{b}z^{-2} + \mathbf{c}z^{-1}\mathbf{A}^3\mathbf{b}z^{-3} + \dots) \\ &= \mathbf{d} + \mathbf{c}\mathbf{b}z^{-1} + \mathbf{c}\mathbf{A}\mathbf{b}z^{-2} + \mathbf{c}\mathbf{A}^2\mathbf{b}z^{-3} + \mathbf{c}\mathbf{A}^3\mathbf{b}z^{-4} + \dots \end{aligned}$$

Where the impulse response of the filter is derived. From the impulse response a FIRF can be made. However, on the contrary to a FIRF, the impulse response is not finite. This is

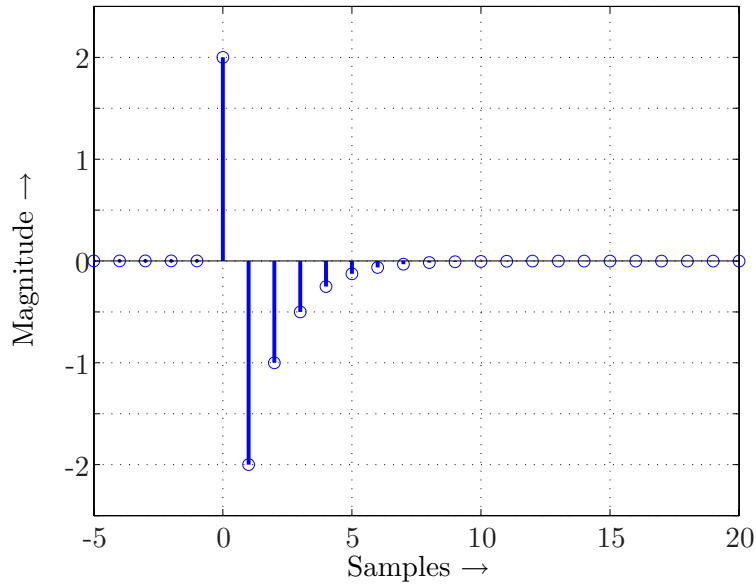


Figure A.10: Impulse response of the filter in equation (A.16).

the same as in an IIRF. The impulse response is infinite. On the other hand, although the impulse response is infinite, the impulse response can decay to zero very fast. Let us clarify that by an example.

Example A.5.2 (Relationship SSF, IIRF and FIRF). Assume the following first order SS filter as seen earlier:

$$\begin{aligned} z(k+1) &= 0.5z(k) + u(k) \\ y(k) &= -2z(k) + 2u(k) \end{aligned} .$$

The impulse response for this filter can easily be derived as:

$$\begin{aligned} \mathbf{R}(z) &= d + cbz^{-1} + cabz^{-2} + ca^2bz^{-3} + ca^3z^{-4} + ca^4bz^{-5} + ca^5bz^{-6} + ca^6bz^{-7} \dots \\ &= 2 - 2z^{-1} - 1z^{-2} - 0.5z^{-3} - 0.25z^{-4} - 0.125z^{-5} - 0.0625z^{-6} - 0.03125z^{-7} + \dots \end{aligned}$$

The impulse response is depicted in figure A.10.

From the picture it can be deduced the the impulse response decays to zero fast. From this a nice observation can be made. Suppose the SS filter is a real system and the impulse response in figure A.10 is measured. From the impulse response, the FIRF can easily be constructed.

Appendix B

The feedforward model based Wiener controller

This appendix states the feedforward model based Wiener controller in terms of models of the primary path \mathbf{P} and secondary path \mathbf{S} . For the development of the model based Wiener controller, references are available. Specifically, (Vidyasagar, 1985, section 6.2) gives an explanation on the derivation of model based Wiener controller for the feedforward control scenario. (Fraanje, 2004, Chapters 2, 3) also considers the development of the model based Wiener controller in detail for both the feedforward control scenario and the feedback control scenario. In (Fraanje, 2004, Chapter 4) a robust version of the model based Wiener controller is considered using a stochastic model uncertainty for the secondary path \mathbf{S} . Finally, (Elliott, 2000b, Chapter 5) also explains how the model based Wiener controller can be derived for the feedforward control scenario. (Elliott, 2000b, Chapter 6) considers robust feedback controllers to deal with uncertainties in the secondary path \mathbf{S} (see also (Elliott, 1999; Hu et al., 1998; Rafaely and Elliott, 1999; Elliott, 2000a)). Due to the presentations in the mentioned references, in this appendix only the main results are stated.

In figure B.1 a block-diagram of a MIMO feedforward SD-AVIC system is given which is a copy of figure 3.1. Ideally, the controller \mathbf{W} should be such that the transfer path $\mathbf{S}\mathbf{W}$ equals that of the transfer path \mathbf{P} . This implies that the controller \mathbf{W} has to be equal to:

$$\mathbf{W} = -\mathbf{S}^{-1}\mathbf{P}, \quad (\text{B.1})$$

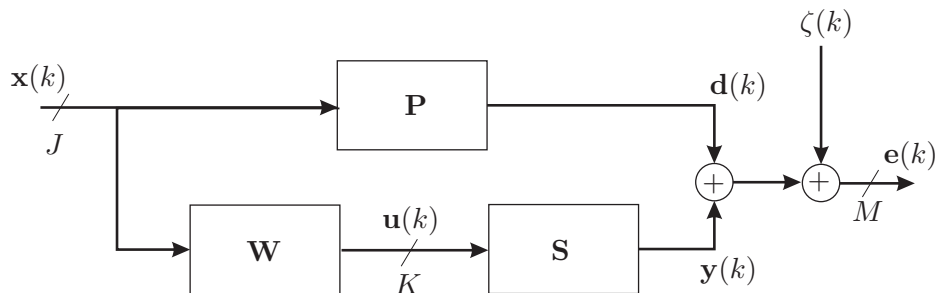


Figure B.1: Block-diagram of a MIMO feedforward SD-AVIC system with a fixed gain controller \mathbf{W} and asymptotically stable primary path \mathbf{P} and secondary path \mathbf{S} .

assuming that \mathbf{S} can be inverted. In most control systems, the controller \mathbf{W} can not be computed like in equation (B.1), since the secondary path \mathbf{S} has zeros which are outside the unit circle (the inverted system would be unstable). The main question is then how to compute the controller \mathbf{W} . The answer is that the controller \mathbf{W} is not computed directly as in equation (B.1), but that there is searched for the best suitable controller \mathbf{W} which satisfies a certain criterion. Specifically, the computation of the controller \mathbf{W} is done by minimizing a cost-function. Good performance is generally defined in terms of minimizing a mean square, or H_2 , type of cost-function posed on the error sensor outputs (Elliott, 2000b). This can be understand, since that has the physical interpretation of minimizing the power of the error sensor outputs. This results in the following controller design problem (and immediately the regularized version is treated).

Problem B.0.1 (Model based Wiener controller). *Given the asymptotically stable primary path \mathbf{P} and the asymptotically stable secondary path \mathbf{S} , find an asymptotically stable controller $\check{\mathbf{W}}$, such that:*

$$\check{\mathbf{W}} = \arg \min_{\mathbf{W}} J,$$

where the cost-function is defined as:

$$J = E[\mathbf{e}^T(k)\mathbf{e}(k)] + E[\beta\mathbf{u}^T(k)\mathbf{u}(k)], \quad (\text{B.2})$$

with E the statistical expectation operator (Leon-Garcia, 1994) and where term $E[\beta\mathbf{u}^T(k)\mathbf{u}(k)]$ is added to robustify the controller.

The solution to this problem is given by the model based Wiener controller as explained in section 3.2.2. However, in that particular section the Wiener controller is derived assuming a FIRF structure for the controller of which the parameters are derived in terms of correlation and cross-correlation matrices of the filtered reference inputs and disturbance outputs (see equation (3.26)). Here the controller is defined in terms of models of the primary path \mathbf{P} and secondary path \mathbf{S} . Without going in further details, the main result are stated. Given the primary path \mathbf{P} and the secondary path \mathbf{S} with no zeros on the unit circle. Then the model based Wiener controller is given by:

$$\check{\mathbf{W}} = -\mathbf{S}_{auo}^{-1}[\mathbf{S}_{aui}^*\mathbf{P}_{au}]_+, \quad (\text{B.3})$$

with $[\cdot]_+$ the causality operator (Fraanje, 2004). The proof can be found in (Vidyasagar, 1985), (Fraanje, 2004) (see equation (2.20)) and (Elliott, 2000b) (see equation (5.3.31)). \mathbf{P}_{au} represents the primary path which is augmented due to the regularization factor β :

$$\mathbf{P}_{au} = \begin{bmatrix} \mathbf{P} \\ \mathbf{0}_{M \times J} \end{bmatrix},$$

with $\mathbf{0}_{M \times J}$ a $M \times J$ matrix with zeros. Furthermore

$$\mathbf{S}_{au} = \mathbf{S}_{aui}\mathbf{S}_{auo},$$

represents an inner/outer factorization of the augmented secondary path \mathbf{S}_{au} (Vidyasagar, 1985; Dewilde and van der Veen, 1998) (see also equation (3.58)):

$$\mathbf{S}_{au} = \begin{bmatrix} \mathbf{S} \\ \sqrt{\beta}\mathbf{I}_{M \times K} \end{bmatrix},$$

with $\sqrt{[\cdot]}$ the square root operation and with $\mathbf{I}_{M \times K}$ a $M \times K$ identity matrix.

Appendix C

Simulation example: subspace model identification of the primary path of the one-degree-of-freedom mass/damper/spring system

C.1 Introduction

In this appendix a practical example is given to demonstrate the potential of subspace model identification (SMI). Here the primary path $\mathbf{P}(s)$ is identified as defined in equation (2.8) with the sizes of the constants as defined in table 2.1:

$$\mathbf{P}(s) = \frac{f_r(s)}{f_p(s)} = \frac{c_d s + k_f}{m s^2 + c_d s + k_f} = \frac{0.01s + 10}{0.1s^2 + 0.01s + 10}.$$

In Matlab, the 2^{nd} order continuous time system is discretized at a sampling frequency¹ of 20 Hz, by using a zero-order-hold function (Phillips and Nagle, 1995). Discretization of the continuous time system results in a 2^{nd} order discrete time system of which the system ‘matrices’ are given by:

$$\left[\begin{array}{c|c} \mathbf{A} & \mathbf{b} \\ \hline \mathbf{c} & d \end{array} \right] = \left[\begin{array}{cc|c} 1.7508 & -0.4975 & 0.5000 \\ 2.0000 & 0 & 0 \\ \hline 0.2540 & 0.1172 & 0 \end{array} \right]. \quad (\text{C.1})$$

In accordance with equation (2.15), these matrices can be considered the system matrices of the following system:

$$\mathbf{P} \sim \begin{cases} \mathbf{z}(k+1) & = \mathbf{A}\mathbf{z}(k) + \mathbf{b}x(k) + \psi(k) \\ e(k) & = \mathbf{c}\mathbf{z}(k) + dx(k) + \iota(k) \end{cases} \quad (\text{C.2})$$

The primary path \mathbf{P} has a complex conjugate pole at $0.8754 \pm 0.4782j$.

¹20 Hz is more than sufficient to cover the (single) resonance peak which is at exactly 10 rad/s=1.59 Hz i.e. 20 Hz is beyond the frequency of the frequency of the resonance peak.

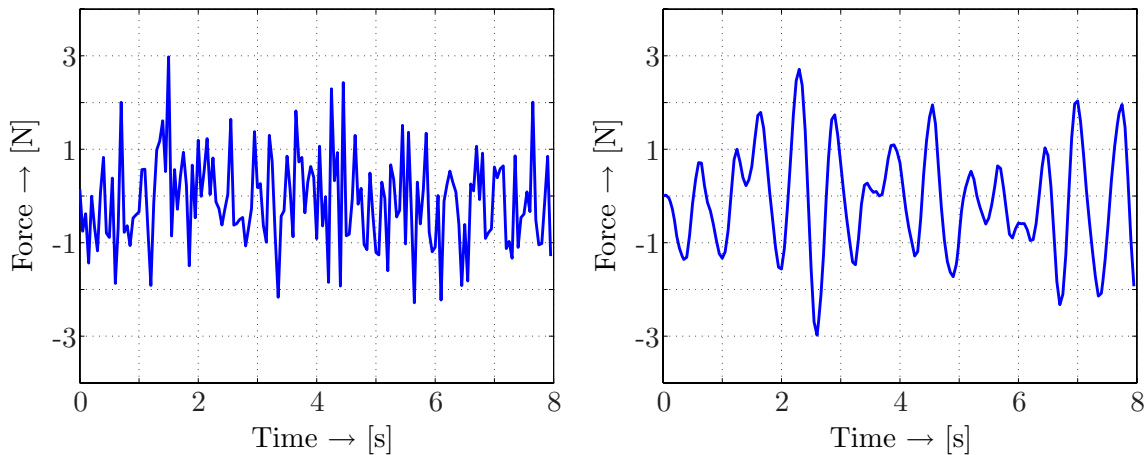


Figure C.1: The generated input sequence (left) and the captured output sequence (right).

Problem C.1.1 (Estimating the system matrices). Assume that the system matrices in equation (C.2) are unknown (i.e. the quantities in equation (C.1) are unknown), but that an input/output data-set of length \mathcal{N} is available:

$$[x(k), e(k)]_{k=1}^{\mathcal{N}}.$$

The primary goal of the SMI procedure is to find estimates $\hat{\mathbf{A}}$, $\hat{\mathbf{b}}$, $\hat{\mathbf{c}}$ and \hat{d} of the system matrices \mathbf{A} , \mathbf{b} , \mathbf{c} and d based on the input/output data-set.

Two simulation experiments are performed: in the first experiment it is assumed that the process noise $\psi(k)$ and the measurement noise $\iota(k)$ are zero in equation (C.2). In the second experiment 50 dB process noise $\psi(k)$ and 50 dB measurement noise $\iota(k)$ are added, with respect to the noiseless state and the noiseless output, respectively.

This appendix is organized as follows. In section C.2 identification is described in case there is no process and measurement noise. Finally, in section C.3 identification is described in case there is process and measurement noise.

C.2 Without process and measurement noise.

A zero mean, unit variance, white Gaussian noise sequence is generated which is chosen to be $\mathcal{N}=160$ samples (eight seconds of data) and which is fed to the system \mathbf{P} in equation (C.2) i.e. this is the persistently exciting input sequence (see figure C.1). The output is recorded. The input/output data-set is split in four seconds of data for training the model and four seconds of data for validating the model. In real-life, the input/output data-set is obviously obtained from the real-life system and is not withdrawn from a simulation environment. The number of rows in the Hankel matrix is set to $s = 10$. Loosely speaking: the parameter s can be seen as the maximum model order (minus one) which can be identified. As such, in this case up to model order nine can be identified. The number of rows s is the first of two parameters which need to be set in the subspace identification procedure. Although a proper choice is relevant, it is not critical. Defining the system, setting the constants and generating the signals in Matlab is done using the following m-code.


```

% Initialize constants
m = 0.1; % Mass
kf = 10; % Spring constant
c = 0.01; % Damper constant
Fs = 20; % Sample frequency
LL = 160; % Number of samples
s = 10; % Rows in the Hankel matrix

% Discretize the primary path
P = tf([0 c kf],[m c kf]); % Continuous time transfer function
Pd = c2d(P,1/Fs,'zoh'); % Discrete time transfer function
[A,B,C,D] = ssdata(Pd); % Extract system matrices

% Generate main signals
x = randn(LL,1); % Zero mean unit variance white noise
e = dlsim(A,B,C,D,x); % Output sequence

% Generate training and validation data
xt = x(1:end/2); % Training input data
et = e(1:end/2); % Training output data
xv = x(end/2+1:end); % Validation input data
ev = e(end/2+1:end); % Validation output data

```

Given these initial settings the identification problem can be solved in three steps: system order estimation, system matrix estimation and model validation.

- **Step 1: System order estimation.** The first Matlab routine which need to be called in the SMI procedure is the routine which builds the data-matrix, which is commonly indicated by R . From the data-matrix R the so called singular values are computed from which the order of the model can be estimated². If the order is chosen, the matrices $\hat{\mathbf{A}}$ and $\hat{\mathbf{c}}$ can directly be constructed from the data-matrix R and $\hat{\mathbf{b}}$ and \hat{d} can be computed by solving a linear least square problem using the input/output data-set and $\hat{\mathbf{A}}$ and $\hat{\mathbf{c}}$. In the SLICOT toolbox the routine to construct the data-matrix R is the routine `order`.

```

% Compute data matrix
[R,Ns,sval] = order(1,3,1,4,2,s,et,xt);

```

The routine requires various arguments, which are not described in detail. In Matlab the user can type `help order` for a detailed overview of arguments. The first argument is worth mentioning though. Setting this first scalar argument to the scalar one '1', selects the PO-MOESP routine, although other routines are also available. Note that the routine `order` gives back the data-matrix R , an estimate of the order N_s and the singular values `sval` (N_s is derived from `sval`). The singular values `sval` for this particular problem are plotted in figure C.2. From the singular values it is clear that the model order should be equal to 2, since the singular values display a clear gap

²To be precise: the singular values are the singular values of the approximated extended observability matrix of the system (2.15) (Verhaegen, 1994).

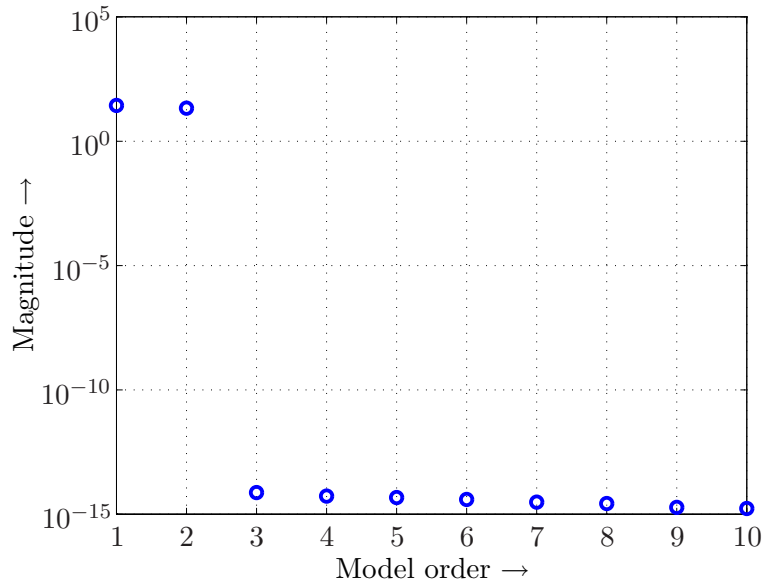


Figure C.2: Singular vales of the system (2.15) with system matrices as in equation (C.1).

between the second and the third singular value. To be exact, from the third singular values onwards, the singular values drop to a size which equals the machine precision. This is expected since the true system order is 2. As such, it can be seen that SMI, by means of the singular values, gives back an excellent estimate of the appropriate model order N_s . Therefor, order 2 is chosen. The choice of the model order is the second parameter which need to be chosen in the SMI procedure. Based on the chosen model order and the input/output data-set, the system matrices can be estimated.

- **Step 2: System matrix estimation.** The estimation of the matrices \mathbf{A} , \mathbf{b} , \mathbf{c} and d is done using the routine `sident` as follows:

```
% Compute system matrices
[Ae,Ce,Be,De] = sident(1,1,s,2,1,R);
```

The function `sident` also uses several arguments. The first scalar (one - '1'), selects the PO-MOESP routine, the fourth scalar (two - '2'), selects the model order and the last argument is the computed data-matrix R which is computed by the routine `order`. The routine `sident` immediately gives back estimates of \mathbf{A} , \mathbf{b} , \mathbf{c} and d .

- **Step 3: Model validation.** After the computation of the system matrices, the model is validated.

```
% Model validation
SYSe      = ss(Ae,Be,Ce,De,1/Fs);      % Construct system in Matlab
x0v       = inistate(SYSe,ev,xv);      % Compute initial state
eev       = dlsim(Ae,Be,Ce,De,xv,x0v); % Estimated validation sequence
Result_VAF = vaf(ev,eev);              % Compute VAF
```

The final command gives back the result in terms of the VAF. The VAF appears to be exactly 100% which indicates that the estimated model is perfectly equal to the true system. The estimated matrices are given by:

$$\hat{\mathbf{P}} \sim \left[\begin{array}{c|c} \hat{\mathbf{A}} & \hat{\mathbf{b}} \\ \hline \hat{\mathbf{c}} & \hat{d} \end{array} \right] = \left[\begin{array}{cc|c} 0.8870 & -0.5851 & 0.9946 \\ 0.3911 & 0.8638 & -0.5238 \\ \hline 0.3035 & 0.3339 & -1.5328 \cdot 10^{-16} \end{array} \right] \quad (\text{C.3})$$

These matrices are however not identical to the system matrices as in equation (C.1). That would have been a coincidence. They are only similar up to a similarity transformation. However, the model $\hat{\mathbf{P}}$ with the matrices as in equation (C.3) has the same input/output behavior as the system \mathbf{P} with the matrices in equation (C.1). Inspection of the eigenvalues of the estimated matrix $\hat{\mathbf{A}}$ learns that they are identical to the eigenvalues of the true matrix \mathbf{A} i.e. a complex conjugate pole at $0.8754 \pm 0.4782j$.

It is concluded that SMI is able to find an accurate linear model. Next experiments are described in which process noise $\psi(k)$ and measurement noise $\iota(k)$ are added.

C.3 With 50 dB process noise and 50 dB measurement noise.

The process noise $\psi(k)$ and the measurement noise $\iota(k)$ are set to a signal-to-noise-ratio (SNR) of 50 dB with respect to the noiseless state and the noiseless output in equation (C.2), respectively. That is done by replacing the section `% Generate main signals` as described earlier by the following m-code.

```
% Generate main signals
SNR          = 50; % signal to noise ratio
x            = randn(LL,1);
[e,z]        = dlsim(A,B,C,D,x);
[xn,z,w,VarNoiseW] = snr(SNR,z);
[en,e,v,VarNoiseV] = snr(SNR,e);
e            = dlsim(A,[B [1 0;0 1] [0;0]],C,[D 0 0 1],[u w v]);
```

which gives us a noisy output signal. The singular values are generated in the same way using the command `order` and are given in figure C.3 on the left. On purpose the y-axis of the figure is kept equivalent to the y-axis in figure C.2 to see the effect of the noise on the singular values. On the right of figure C.3 a zoomed version is given. It can be concluded that a clear distinction exist between the noiseless case in figure C.2 and the noisy case in figure C.3. In the noiseless case, the gap between the second and the third singular value is much larger. But also in the noisy case, inspection of the singular values reveals clearly that the model order should be 2, since still a distinct gap can be seen between singular values 2 and 3. As such, the system matrices are estimated using the following m-code:

```
% Compute system matrices and noise covariances
[Ae,Ce,Be,De] = sident(1,1,s,2,1,R);
```

The VAF value for this particular experiment is now 99.9% which is worse than 100%. The estimated system matrices for the particular realizations of the input sequence $x(k)$, the

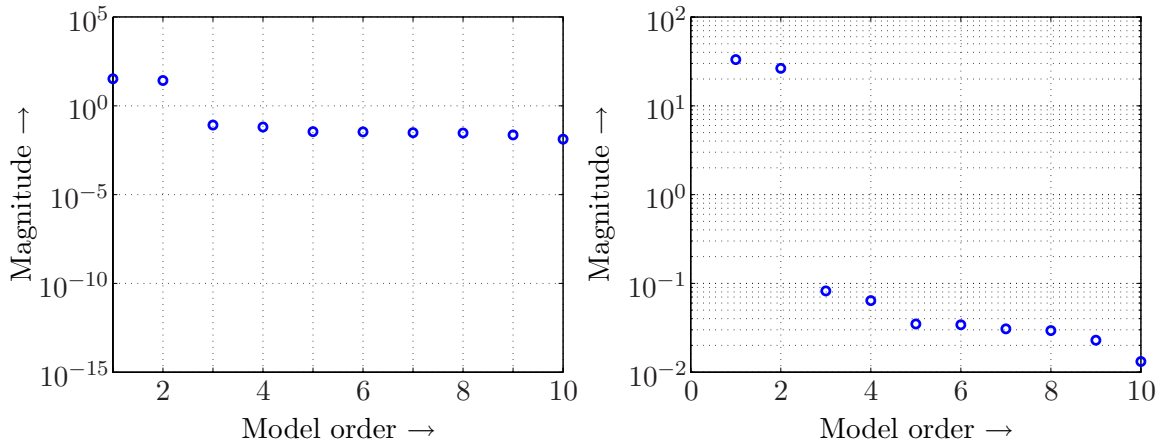


Figure C.3: Singular vales of the system (2.15) with system matrices as in equation (C.1) based on noisy input/output data.

process noise $\psi(k)$ and the measurement noise $\iota(k)$ are given by:

$$\left[\begin{array}{c|c} \widehat{\mathbf{A}} & \widehat{\mathbf{b}} \\ \hline \widehat{\mathbf{c}} & \widehat{d} \end{array} \right] = \left[\begin{array}{cc|c} 0.8512 & 0.5822 & 0.9019 \\ -0.3930 & 0.9007 & 0.7045 \\ \hline 0.3603 & -0.2699 & 8.1752 \cdot 10^{-4} \end{array} \right] \quad (\text{C.4})$$

Again, these system matrices are not identical to the matrices as defined in equation (C.1), for similar reasons as explained earlier. Inspection of the eigenvalues of $\widehat{\mathbf{A}}$ learns that they are almost identical to the eigenvalues of \mathbf{A} i.e. a complex conjugate pole at $0.8760 \pm 0.4777j$. They are not exactly identical due to the noise which is present in the system. Still, our conclusion is that SMI is able to identify an accurate linear model.

To get more insight in SMI, 250 different identification experiments are performed. For every experiment a different realization is used for the input $x(k)$, the process noise $\mathbf{w}(k)$ (50 dB) and the measurement noise $v(k)$ (50 dB). After every experiment the eigenvalues are computed of the estimated matrix $\widehat{\mathbf{A}}$. As seen earlier, the eigenvalues are a complex conjugate pole-pair of the form $\alpha \pm j\beta$. In figure C.4 on the left the particular pole $\alpha + j\beta$ is plotted. It can be seen that the identified poles lie distributed around the optimal pole $0.8754 \pm 0.4782j$. The optimal pole lies at the crossing of the two solid lines. If the average pole over the 250 experiments is computed, which is $0.8753 \pm 0.4782j$, it can be seen that the average pole is approximately equal to the true pole. As such, SMI gives back a consistent estimate. On the right of figure C.4 the VAF is plotted on the validation data for all 250 experiments. It is obvious that the VAF is not constant due to the different realizations of input $x(k)$, the process noise $\mathbf{w}(k)$ and the measurement noise $v(k)$. Most of the time though, the VAF is above 99.5%. Note that the VAF can never be 100%. Even if the system matrices are 100% accurate, the VAF would not be 100%.

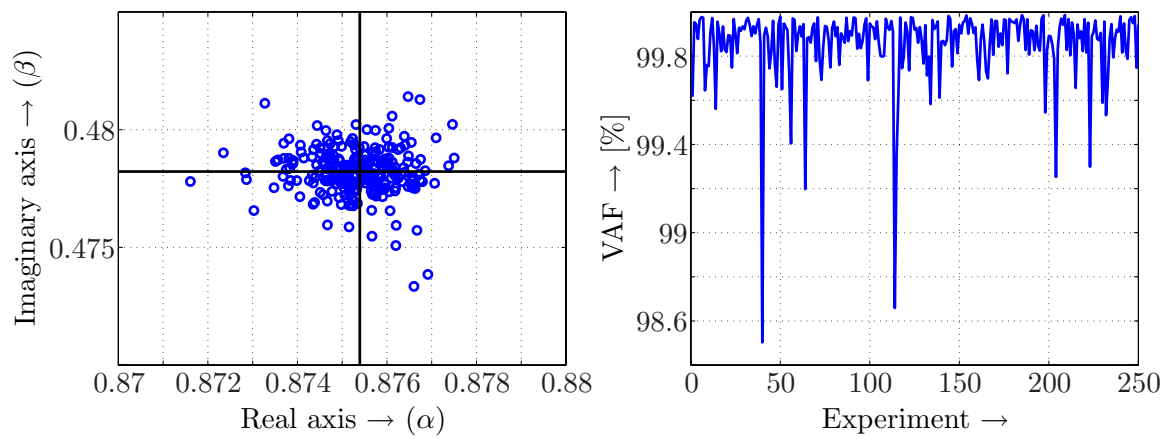


Figure C.4: Left: The pole $\alpha + j\beta$ for 250 different experiments. Right: The VAF on the validation output data for 250 different experiments. The big plus in the figure denotes the true pole. The average VAF over the 250 experiments is 99.8570% and the variance of the VAF is 0.0288.

Appendix D

Extracting a finite impulse response model for purposes of tonal control

In chapter 4 a model $\hat{\chi}$ is identified for the system χ , which is a broadband model valid over the entire frequency band. However, for controlling a single tone, a broadband model is not required. More specifically, a FIRF of two coefficients is sufficient to model the input/output behavior of a SISO plant for an input signal consisting of a single tone. For that reason, from the estimated model $\hat{\mathbf{S}}$ of the secondary path \mathbf{S} a simple FIRF is extracted which models the transfer of the secondary path \mathbf{S} at the disturbance frequency.

The transfer of the secondary path \mathbf{S} for frequency $\omega_1 T$ is given by the matrix of complex plant responses:

$$\mathbf{S}(e^{j\omega_1 T}) \in \mathbb{C}^{M \times K}, \quad (\text{D.1})$$

with $j^2 = -1$. Every element of $\mathbf{S}(e^{j\omega_1 T})$ is a complex coefficient. The element (m, l) , $(m = 1, \dots, M)$, $(l = 1, \dots, K)$, of equation (D.1) is given by:

$$\mathbf{S}_{(m,l)}(e^{j\omega_1 T}) = \alpha + j\beta, \quad (\text{D.2})$$

with $\alpha, \beta \in \mathbb{R}$. Equation (D.2) can be transformed to a finite impulse response model $\mathbf{s}_{(m,l)}(z^{-1})$ of two coefficients:

$$\begin{aligned} \mathbf{s}_{(m,l)}(z^{-1}) &= s_{(m,l)}^{(1)} + s_{(m,l)}^{(2)} z^{-1} \\ &= s_{(m,l)}^{(1)} + s_{(m,l)}^{(2)} e^{-j\omega_0 T} \\ &= s_{(m,l)}^{(1)} + s_{(m,l)}^{(2)} \cos(\omega_0 T) - j s_{(m,l)}^{(2)} \sin(\omega_0 T), \end{aligned} \quad (\text{D.3})$$

with z^{-1} the z-domain operator. In order to obtain the unknown coefficients $s_{(m,l)}^{(1)}$ and $s_{(m,l)}^{(2)}$, on the RHS of equation (D.2) and the right hand side of the third (lowest) equation in (D.3) must be equalized:

$$\alpha + j\beta = s_{(m,l)}^{(1)} + s_{(m,l)}^{(2)} \cos(\omega_0 T) - j s_{(m,l)}^{(2)} \sin(\omega_0 T) \quad (\text{D.4})$$

Equation (D.4) gives rise to two equations with two unknowns which can be solved for $s_{(m,l)}^{(1)}$ and $s_{(m,l)}^{(2)}$:

$$\begin{aligned} s_{(m,l)}^{(1)} + s_{(m,l)}^{(2)} \cos(\omega_0 T) &= \alpha \\ -s_{(m,l)}^{(2)} \sin(\omega_0 T) &= \beta \end{aligned} \quad (\text{D.5})$$

The solution for $s_{(m,l)}^{(1)}$ and $s_{(m,l)}^{(2)}$ which follows from (D.5) is given by:

$$\begin{bmatrix} s_{(m,l)}^{(1)} \\ s_{(m,l)}^{(2)} \end{bmatrix} = \begin{bmatrix} 1 & \cos(\omega_0 T) \\ 0 & -\sin(\omega_0 T) \end{bmatrix}^{-1} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \quad (\text{D.6})$$

where $[\cdot]^{-1}$ denotes the inverse of the matrix $[\cdot]$. When in equation (D.6) the approximates $\hat{\alpha}$ and $\hat{\beta}$ (which are known from the model $\hat{\mathbf{S}}$) are substituted for α and β , together with the normalized frequency $\omega_0 T$, the equation (D.6) can be solved and a simple FIRF can be found consisting of two taps $\hat{s}_{(m,l)}^{(1)}$ and $\hat{s}_{(m,l)}^{(2)}$.

Appendix E

Hardware used for the identification and control tasks

E.1 The components used for data acquisition

The piezo-electric actuators are Pst150/5/40 VS10 actuators from Piezomechanik with a length of 46 mm, a stroke of 40 μm and a stiffness of 12 $\text{N}/\mu\text{m}$ (see figure E.1 on the left). The control signals which are sent to the piezo-electric actuators, are amplified by three channel Piezomechanik SVR 150/3 piezo-amplifiers with a voltage range between -30 V and 150 V (see figure E.1 on the right). Also, the amplifiers can provide a manual offset also between -30 V and 150 V. The acceleration sensors for the SR-MIMO vibration isolation



Figure E.1: Left: A Piezomechanik Pst150/5/40 VS10 piezo-electric actuator. Right: A Piezomechanik SVR 150/3 piezo-amplifier.

set-up are from Brüel and Kjær and are miniature 4393 sensors fitted with a subminiature connector. The sensors have a charge sensitivity of $0.316 \text{ pCm}^{-1}\text{s}^{-2}$ and can detect signals in a frequency range between 0.1 Hz - 16.5 kHz (see figure E.2 on the left). The sensor for the 1DOF vibration isolation set-up is from Brüel and Kjær and is a 8001 impedance head with a build in acceleration sensor (see figure E.2 in the middle). The sensor can detect signals between 1 Hz - 10 kHz and has a charge sensitivity of $3 \text{ pCm}^{-1}\text{s}^{-2}$. The signals from the acceleration sensors are amplified by four channel Brüel and Kjær 2692 conditioning Nexus amplifiers with build in low-pass and high-pass filters (see figure E.2 on the right) (Enden,

1989). The low-pass filter can be used for anti-aliasing purposes and is a 2^{nd} order low-pass filter which rolls-off with 40 dB per decade. The cut-off frequency can be set to 0.1 kHz, 1 kHz, 3 kHz, 10 kHz, 22.4 kHz, 30 kHz and 100 kHz. Reconstruction filters from Frequency devices Inc. are used to condition the signals to the piezo-electric actuators. The filters are 4^{th} order 424L4L Bessel filters with a cut-off frequency at 1 kHz and 80 dB roll-off per decade. The shaker is a Brüel and Kjær 4809 vibration exciter which can generate vibrations between 10 Hz and 20 kHz and which has a first axial resonance of 20 kHz (see figure E.3 on the left). The signals which are sent to the shaker are amplified by a Brüel and Kjær 2706 amplifier.

E.2 dSPACE systems

In this section the dSPACE systems are described which are used for the experiments. First in section E.2.1 the DS1103 controller is explained, a single controller board which is plugged directly into an expansion slot of a general purpose computer. In section E.2.2 the fast modular dSPACE system is described, which is placed in a separate expansion box which is linked to a general purpose computer.

E.2.1 The dSPACE DS1103 controller board

For control a single board dSPACE DS1103 PPC ISA controller board was used with 128 megabyte internal main memory (see figure E.3 on the right). The DS1103 board contains a Motorola PowerPC 604e master general purpose processor running at 400 MHz and a Texas Instruments TMS320F240 slave digital signal processor running at 20 MHz. The controller board is mounted on an ISA slot of a general purpose computer with an Intel Pentium II 250 megahertz processor. The master system of the controller board provides four 16 bit multiplexed analog to digital converter (ADC) inputs of four channels each and four 12 bit ADC input channels. Furthermore, the master system of the board provides eight 14 bit digital to analog converter (DAC) output channels. The voltage ranges of the input and output channels are between -10 V and 10 V. The latter explains why amplifiers for the input signals to the six piezo-electric actuators are used i.e. to make use of their full voltage range. The slave system of the controller board provides two 10 bits ADC units of eight inputs each, giving rise to sixteen extra ADC input channels in addition to the master system.



Figure E.2: Left: A Brüel and Kjær 4393 acceleration sensor. Middle: A Brüel and Kjær 8001 impedance head. Right: A Brüel and Kjær 2692 Nexus amplifier.

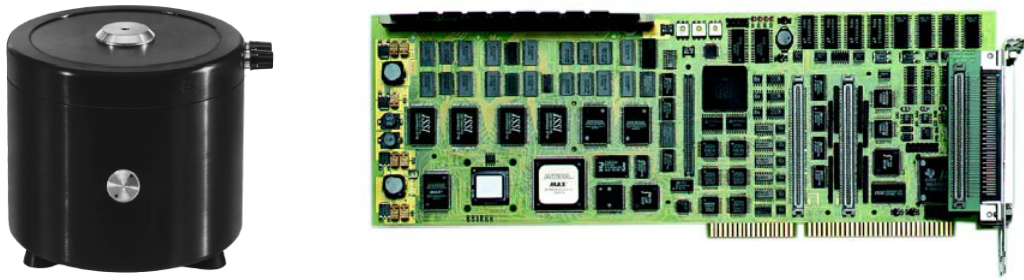


Figure E.3: *Left: The Brüel and Kjær 4809 vibration exciter. Right: The ISA single board DS1103 dSPACE controller.*

Programming the dSPACE controller board is performed via the real-time interface in Matlab Simulink (Redfern and Campbell, 1998; Thoeni, 1994).

E.2.2 The modular dSPACE system with DS1005 controller board

Besides the DS1103 controller board with integrated ADCs and DACs, also experiments are performed with a modular system from dSPACE. Specifically, with the DS1005 processor board, two DS2001 high-resolution ADC boards and two DS2102 fast DAC boards. The five boards are integrated in an dSPACE expansion box, which provides space for nine dSPACE boards. As such, with the modular system there is the possibility to use several controller boards (e.g. several DS1005 controller boards) for parallelization. The connection between the dSPACE system and the PC is made via a fast optical link which provides a data transfer rate of 1.25 gigabyte per second. The DS1005 controller board is equipped with a 800 megahertz IBM 750 FX PowerPC and 128 megabyte of main memory. The DS2001 has five high-resolution ADCs, so in total the two DS2001 boards provide ten high-resolution ADCs. The DS2101 has six fast DACs, so in total the two DS2102 DAC boards provides twelve fast DACs.

Appendix F

The two-step regularization approach

On page 57 it is mentioned that it is possible to regularize the inverse outer factor model for greater regularization in the controller. In this appendix is elaborated on the solution.

Take a look at the feedforward control scheme depicted in figure F.1, with J reference inputs, K actuator inputs and M error sensor outputs. Assume that the controller $\check{\check{W}}$ is given as a series connection of a regularized inverse outer factor $\check{\check{S}}_o^{-1}$ and a certain regularized controller \check{W} (see page 25 for the other symbols). The regularized outer factor satisfies equation (3.65):

$$\check{\check{S}}_o^* \check{\check{S}}_o = \mathbf{S}_o^* \mathbf{S}_o + \varpi \mathbf{I}_K.$$

Here a first regularization step is taken if $\varpi > 0$. Assuming a regularized H_2 type of solution for the controller (as in this thesis), the controller \check{W} in figure F.1 is given by:

$$\check{W} = \arg \min_{\check{W}} E [e^T(k)e(k)] + E [\rho \bar{u}^T(k)\bar{u}(k)],$$

in which there is a control penalty on the $\bar{u}(k)$. Basically, by including the regularized inverse outer factor $\check{\check{S}}_o^{-1}$, a new control problem is created in the sense of finding a controller \check{W}

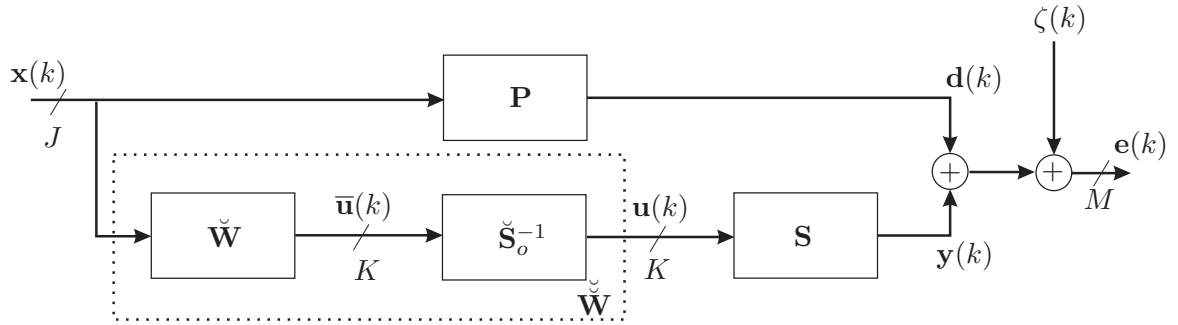


Figure F.1: Block-diagram of a MIMO feedforward SD-AVIC system with a fixed gain controller $\check{\check{W}}$ which is a series connection of a regularized inverse outer factor $\check{\check{S}}_o^{-1}$ and a regularized controller \check{W} .

which is acting on a new secondary path \mathbf{S}^n given by:

$$\mathbf{S}^n = \mathbf{S}\check{\mathbf{S}}_o^{-1}.$$

It is easy to understand that the solution for $\check{\mathbf{W}}$ can be derived from equation (B.3):

$$\check{\mathbf{W}} = -\left(\check{\mathbf{S}}_o^n\right)^{-1} \left[\left(\check{\mathbf{S}}_i^{n*}\right)_{K \times M} \mathbf{P} \right]_+,$$

where

$$\mathbf{S}^n = \check{\mathbf{S}}_i^n \check{\mathbf{S}}_o^n \tag{F.1}$$

represents the regularized inner/outer factorization of the augmented new secondary path $\check{\mathbf{S}}^n$:

$$\check{\mathbf{S}}^n = \begin{bmatrix} \mathbf{S}^n \\ \sqrt{\rho} \mathbf{I}_{M \times K} \end{bmatrix},$$

such that it holds that:

$$\left(\check{\mathbf{S}}_o^n\right)^* \left(\check{\mathbf{S}}_o^n\right) = \left(\mathbf{S}^n\right)^* \left(\mathbf{S}^n\right) + \rho \mathbf{I}_K. \tag{F.2}$$

The full controller $\check{\check{\mathbf{W}}}$ is given by:

$$\begin{aligned} \check{\check{\mathbf{W}}} &= \check{\mathbf{S}}_o^{-1} \check{\mathbf{W}} \\ &= -\check{\mathbf{S}}_o^{-1} \left(\check{\mathbf{S}}_o^n\right)^{-1} \left[\left(\check{\mathbf{S}}_i^{n*}\right)_{K \times M} \mathbf{P} \right]_+. \end{aligned} \tag{F.3}$$

Summary

Vibration isolation endeavors to reduce the transmission of vibration energy from one structure (the source) to another (the receiver), to prevent undesirable phenomena such as sound radiation. A well-known method for achieving this is passive vibration isolation (PVI). In the case of PVI, mounts are used - consisting of springs and dampers - to connect the vibrating source to the receiver. The stiffness of the mount determines the fundamental resonance frequency of the mounted system and vibrations with a frequency higher than the fundamental resonance frequency are attenuated. Unfortunately, however, other design requirements (such as static stability) often impose a minimum allowable stiffness, thus limiting the achievable vibration isolation by passive means.

A more promising method for vibration isolation is hybrid vibration isolation control. This entails that, in addition to PVI, an active vibration isolation control (AVIC) system is used with sensors, actuators and a control system that compensates for vibrations in the lower frequency range. Here, the use of a special form of AVIC using statically determinate stiff mounts is proposed. The mounts establish a statically determinate system of high stiffness connections in the actuated directions and of low stiffness connections in the unactuated directions. The latter ensures PVI in the unactuated directions. This approach is called statically determinate AVIC (SD-AVIC). The aim of the control system is to produce anti-disturbance forces that counteract the disturbance forces stemming from the source. Using this approach, the vibration energy transfer from the source to the receiver is blocked in the mount due to the anti-forces.

This thesis deals with the design of controllers generating the anti-forces by applying techniques that are commonly used in the field of signal processing. The control approaches - that are model-based - are both adaptive and fixed gain and feedforward and feedback oriented. The control approaches are validated using two experimental vibration isolation setups: a single reference single actuator single error sensor (SR-SISO) setup and a single reference input multiple actuator input multiple error sensor output (SR-MIMO) setup.

Finding a plant model can be a problem. This is solved by using a black-box modelling strategy. The plants are identified using subspace model identification. It is shown that accurate linear models can be found in a straightforward manner by using small batches of recorded (sampled) time-domain data only. Based on the identified models, controllers are designed, implemented and validated.

Due to resonance in mechanical structures, adaptive SD-AVIC systems are often hampered by slow convergence of the controller coefficients. In general, it is desirable that the SD-AVIC system yields fast optimum performance after it is switched on. To achieve this result and speed up the convergence of the adaptive controller coefficients, the so-called inverse outer factor model is included in the adaptive control scheme. The inner/outer factorization, that

has to be performed to obtain the inverse outer factor model, is completely determined in state space to enable a numerically robust computation. The inverse outer factor model is also incorporated in the control scheme as a state space model. It is found that fast adaptation of the controller coefficients is possible.

Controllers are designed, implemented and validated to suppress both narrowband and broadband disturbances. Scalar regularization is used to prevent actuator saturation and an unstable closed loop. In order to reduce the computational load of the controllers, several steps are taken including controller order reduction and implementation of lower order models. It is found that in all experiments the simulation and real-time results correspond closely for both the fixed gain and adaptive control situation. On the SR-SISO setup, reductions up to 5.0 dB are established in real-time for suppressing a broadband disturbance output (0-2 kHz) using feedback-control. On the SR-MIMO vibration isolation setup, using feedforward-control reductions of broadband disturbances (0-1 kHz) of 9.4 dB are established in real-time. Using feedback-control, reductions are established up to 3.5 dB in real-time (0-1 kHz). In case of the SR-MIMO setup, the values for the reduction are obtained by averaging the reductions obtained in all sensor outputs.

The results pave the way for the next generation of algorithms for SD-AVIC.

Samenvatting

Trillingsisolatie in mechanische systemen richt zich op het verminderen van de transmissie van trillingsenergie van de ene structuur (de bron) naar de andere structuur (de ontvanger). Een bekende methode van trillingsisolatie is passieve trillingsisolatie (PTI). In geval van PTI zijn de noodzakelijke verbindingen tussen bron en ontvanger ten bate van ondersteuning uitgevoerd in de vorm van veren en dempers. De stijfheden van deze veren beïnvloeden en bepalen daardoor de fundamentele resonantiefrequentie van het systeem. Trillingen van de bron met een frequentie hoger dan deze fundamentele resonantiefrequentie worden gereduceerd overgedragen naar de ontvanger. Trillingen met een frequentie lager dan deze fundamentele resonantiefrequentie worden ongereduceerd overgedragen. Aldus zijn deze fundamentele resonantiefrequenties ontwerpcriteria ten bate van PTI. Echter, andere ontwerpcriteria (zoals statische stabiliteit) vereisen een bepaalde ondergrens voor de stijfheid. Hierdoor worden de toepassingsmogelijkheden van passieve trillingsisolatie beperkt.

Een veelbelovende methode voor trillingsisolatie is hybride trillingsisolatie. In geval van hybride trillingsisolatie wordt PTI gecombineerd met een actief trillingsisolatiesysteem (AVIC-systeem) dat bestaat uit sensoren, actuatoren en een regelsysteem. Het AVIC-systeem compenseert voor trillingen met een frequentie lager dan de fundamentele resonantiefrequentie. Hier wordt verder een speciale vorm van AVIC voorgesteld dat gebruik maakt van statisch bepaalde ondersteuning. De statisch bepaalde ondersteuning bestaan uit een systeem dat verbindingen met een hoge stijfheid realiseert in de geactueerde richtingen en lage stijfheid in de niet-geactueerde richtingen. Dat laatste zorgt voor PTI in de niet-geactueerde richtingen. De aanpak wordt aangeduid met de naam statisch bepaalde actieve trillingsisolatie (SD-AVIC). Het doel van het regelsysteem hierbij is om antikrachten op te wekken in de geactueerde richtingen van de ondersteuning die de verstoringen afkomstig van de bron teniet doen. Door gebruik te maken van deze methode wordt de trillingsenergie van de bron naar de ontvanger geblokkeerd in de ondersteuning.

Dit proefschrift heeft als onderwerp het ontwerp van regelsystemen die deze antikrachten opwekken door gebruik te maken van technieken uit de signaalverwerking. De - modelgebaseerde - regelstrategieën zijn gebaseerd op het feedforward regelaarprincipe en het feedback regelaarprincipe en zijn wel of niet adaptief. De regelsystemen zijn experimenteel gevalideerd op twee trillingsisolatie-opstellingen: een opstelling met een enkele referentie-ingang, enkele actuatoringang en een enkele foutsensoruitgang (SR-SISO) en een opstelling met een enkele referentie-ingang, meerdere actuatoringen en meerdere foutsensoruitgangen (SR-MIMO).

Modellen van de systemen, die nodig zijn in deze aanpak, worden opgesteld door gebruik te maken van een black-box-identificatiemethode. De systemen worden geïdentificeerd met behulp van subspace-modelidentificatie. Aangetoond is dat hiermee op een eenvoudige manier nauwkeurige lineaire modellen kunnen worden verkregen door alleen gebruik te maken

van kleine hoeveelheden (gesampelde) tijdsdomeindata. Op basis van deze modellen worden regelaars ontworpen en geïmplementeerd.

Door de aanwezigheid van resonanties in mechanische structuren is de convergentiesnelheid van adaptieve SD-AVIC-systemen laag. In het algemeen is het echter gewenst dat een SD-AVIC-systeem snel de optimale prestatie levert nadat het is ingeschakeld. Om snellere convergentie te verkrijgen, is het zogeheten inverse outer factor-model in de adaptieve regelschema's opgenomen. Om reden van numerieke robuustheid is de inner/outer-factorisatie, die nodig is om het inverse outer factor-model te bepalen, uitgevoerd met gebruikmaking van state space modellen. Evenzo is het model in state space-vorm opgenomen in het regelschema.

De regelaars zijn ontworpen, geïmplementeerd en gevalideerd met als doel zowel smalbandige als breedbandige verstoringen te onderdrukken. Scalaire regularisatie is toegepast om verzadiging van de actuatoren te voorkomen en een instabiele gesloten lus. Om de processorbelasting door regelaars in de digitale implementatie te beperken, zijn verschillende stappen ondernomen, zoals het reduceren van de orde van de regelaar en het implementeren van lagere orde modellen.

Het is aangetoond dat de resultaten van de simulaties grote overeenkomst vertonen met resultaten verkregen uit experimenten. Dit geldt voor zowel de toegepaste adaptieve, als de niet adaptieve regelaars. Op de SR-SISO-opstelling worden in geval van een feedback-gebaseerde regeling reducties van 5.0 dB behaald van breedbandige storingen (0-2kHz). Op de SR-MIMO-opstelling worden in geval van een feedforward-gebaseerde regeling reducties van 9.4 dB behaald van breedbandige storingen (0-1kHz). In geval er een feedback-gebaseerde regeling wordt gebruikt zijn de behaalde reducties tot 3.5 dB van breedbandige storingen (0-1 kHz). In geval van de SR-MIMO-opstelling zijn de waarden voor de reducties verkregen door het bepalen van de gemiddelde waarde van alle sensor-uitgangen.

Glossary

List of symbols

\otimes	- Kronecker tensor product
$\sqrt{(\cdot)}$	- square root operation
$(\cdot)^{-1}$	- inverse operator
$(\cdot)^\dagger$	- pseudo inverse operator
$(\cdot)^T$	- transpose operator
$(\cdot)^*$	- complex conjugate transpose operator
$[\cdot]_+$	- causality operator
$[\cdot]_-$	- anti-causality operator
$\ \cdot\ _2$	- 2-norm operator
$\widehat{(\cdot)}$	- estimate of (\cdot)
$\overline{(\cdot)}$	- regression vector composed of elements of (\cdot)
$\bar{(\cdot)}$	- regularized version of (\cdot)
$\arg(\cdot)$	- argument of (\cdot)
$^{10}\log(\cdot)$	- logarithm operator with base 10
$\min(\cdot)$	- minimizing operator
$E(\cdot)$	- stochastic expectation operator with respect to stochastic signals
\mathbb{R}^n	- set of all n dimensional vectors with real coefficients
$\mathbb{R}^{m \times n}$	- set of all $m \times n$ dimensional matrices with real coefficients
\mathbb{C}^n	- set of all n dimensional vectors with complex coefficients
$\mathbb{C}^{m \times n}$	- set of all $m \times n$ dimensional matrices with complex coefficients
$\mathbf{I}_{m \times n}$	- $m \times n$ identity matrix
\mathbf{I}_n	- identity matrix of size n
$\mathbf{0}_{m \times n}$	- matrix with zeros of size $m \times n$
$\mathbf{0}_n$	- n dimensional vector filled with zeros
δ_{k-l}	- Kronecker delta function
μ	- stepsize parameter
γ	- positive real valued scalar tuning parameter
β	- positive real valued scalar tuning parameter
ϖ	- positive real valued scalar tuning parameter
π	- circle constant
ω	- angular frequency in rad/s
\mathbf{x}	- reference (or disturbance) inputs
\mathbf{x}_m	- primary detector output
\mathbf{x}_c	- primary detector output contaminated with noise

η	- random white noise process
ψ	- process noise
ι	- measurement noise
\mathbf{y}	- secondary path output, counteracting disturbance signal
\mathbf{u}	- controller inputs
\mathbf{e}	- error sensor outputs
\mathbf{w}	- controller parameters
k	- discrete time index
f	- frequency (Hz)
j	- complex unit ($j^2 = -1$)
n	- counter variable
l	- counter variable
i	- counter variable
N	- system order
q^{-1}	- backward shift operator
q	- forward shift operator
\mathbf{z}	- system state
\mathbf{P}	- primary path
\mathbf{S}	- secondary path
\mathbf{H}	- primary detector path
\mathbf{W}	- feedforward controller
\mathbf{S}_i	- secondary path inner factor
\mathbf{S}_o	- secondary path outer factor
J	- cost function
\mathbf{Q}	- process noise covariance matrix
\mathbf{R}	- measurement noise covariance matrix
\mathbf{Z}	- cross-covariance between the process and measurement noise
s	- Laplace domain variable
z	- z-domain variable
\mathbf{d}	- disturbance outputs
ζ	- error sensor measurement noise
ϑ	- detection sensor measurement noise

List of abbreviations

dB	-	decibel
rad	-	radians
sec	-	seconds
ADC	-	Analogue to Digital Converter
ALMS	-	Adjoint Least Mean Square
AVIC	-	Active Vibration Isolation Control
DAC	-	Digital to Analogue Converter
DOF	-	Degree-Of-Freedom
FIR(F/M)	-	Finite Impulse Response (Filter/Model)
FLOPS	-	Floating Point OPERations per Sample
FxLMS	-	Filtered-X Least Mean Square
Hz	-	Hertz
IIR(F/M)	-	Infinite Impulse Response (Filter/Model)
IMC	-	Internal Model Control
IOALMS	-	Inner/Outer based Adjoint Least Mean Square
IOFxLMS	-	Inner/Outer based Filtered-X Least Mean Square
LTI	-	Linear Time Invariant
kHz	-	kilo Hertz
LHS	-	Left Hand Side
LMS	-	Least Mean Square
MDS	-	Mass/Damper/Spring
MEX	-	Matlab EXecutable
MHz	-	Mega Hertz
MIMO	-	Multiple Input Multiple Output
MISO	-	Multiple Input Single Output
MSE	-	Mean Square Error
PCLMS	-	Principle Component Least Mean Square
PEM	-	Prediction Error Model
PO-MOESP	-	Past Output Multi-variable Output Error State sPace
PVI	-	Passive Vibration Isolation
RHS	-	Right Hand Side
SD-AVIC	-	Statically Determinate Active Vibration Isolation Control
SISO	-	Single Input Single Output
SLICOT	-	Subroutine Library In Systems and Control Theory
SMI	-	Subspace Model Identification
SNR	-	Signal to Noise Ratio
SR-SISO	-	Single Reference input Single actuator Input Single error sensor Output
SR-MIMO	-	Single Reference input Multiple actuator Input multiple error sensor Output
SS(F/M)	-	State Space (Filter/Model)
SVD	-	Singular Value Decomposition
VAF	-	Variance Accounted For

Patent and publications resulted from this thesis work

In this chapter the patents and conference and journal publications are listed which resulted from this thesis work. Internal reports and master's thesis are omitted for brevity.

Patent

2005

- Nijssen, G. and Berkhoff, A.: A filter apparatus for actively reducing noise. Patent number P74620P00, University of Twente and TNO Science and Industry.

Publications

2005

- Nijssen, G., Super, H., Dijk, J. V. and Jonker, J.: A robust subspace based approach to feedforward control of broadband disturbances on a six-degrees-of-freedom vibration isolation set-up, *Noise Control Engineering Journal* **53**(5), 208–215.

2004

- Nijssen, G., Super, H., Dijk, J. V. and Jonker, J.: Subspace based feedforward control of broadband disturbances on a six-degrees-of-freedom vibration isolation set-up, *ACTIVE 2004: The International Symposium on Active Control of Sound and Vibration*, Williamsburg, Virginia, USA. *Award winning paper: ACTIVE 2004 Outstanding student paper award.*
- Super, H., Nijssen, G., Dijk, J. V. and Jonker, J.: A hybrid vibration isolation approach demonstrated by a six-degrees-of-freedom vibration isolation set-up, *ACTIVE 2004: The International Symposium on Active Control of Sound and Vibration*, Williamsburg, Virginia, USA.
- Nijssen, G., Dijk, J. V. and Jonker, J.: Feedback control of broadband disturbances on a one-degree-of-freedom vibration isolation set-up, *ISMA2004: International Conference on Noise and Vibration Engineering*, Leuven, Belgium.

- Nijssse, G., Dijk, J. V. and Jonker, J.: The multiple reference principle component least mean squares algorithm: a projection based approach, *Proceedings of The International Conference on Acoustics, Speech and Signal Processing*, Montreal, Quebec, Canada.

2003

- Nijssse, G., Dijk, J. V. and Jonker, J.: Subspace based identification and control of a one-degree-of-freedom experimental vibration isolation set-up, *International conference on mechatronics*, Loughborough University, United Kingdom.
- Nijssse, G., Super, H., Dijk, J. V. and Jonker, J.: Subspace based identification and control of a six-degrees-of-freedom experimental vibration isolation set-up, *European conference on noise control engineering*, Napels, Italy.
- Super, H., Nijssse, G., Dijk, J. V. and Jonker, J.: Design and implementation of a six-degrees-of-freedom vibration isolation set-up, *European conference on noise control engineering*, Napels, Italy.

2001

- Oku, H., Nijssse, G., Verhaegen, M. and Verdult, V.: Change detection in the dynamics with recursive subspace identification, *Proceedings of the 40th Conference on Decision and Control*, Orlando, Florida, USA.
- Nijssse, G., Verhaegen, M. and Doelman, N.: A new subspace based approach to iterative learning control, *Proceedings of the European Control Conference*, Seminário de Villar, Porto, Portugal.
- Kanev, S., Verhaegen, M. and Nijssse, G.: A method for the design of fault-tolerant systems in case of sensor and actuator faults, *Proceedings of the European Control Conference*, Seminário de Villar, Porto, Portugal.
- Nijssse, G., Verhaegen, M. and Doelman, N.: State space modeling of acoustics for active control, *Inter-Noise 2001: The 2001 International Congress and Exhibition on Noise Control Engineering*, The Hague, The Netherlands.

1999

- Nijssse, G., Verhaegen, M., Schutter, B. D., Westwick, D. and Doelman, N.: State space modeling in multichannel active control systems, *ACTIVE 99: The 1999 International Symposium on Active Control of Sound and Vibration*, Fort Lauderdale, Florida, USA, pp. 909–920.
- Boel, R., Schutter, B. D., Nijssse, G., Schumacher, J. and van Schuppen, J.: Approaches to modelling, analysis, and control of hybrid systems, *Journal A* **40**(4), 16–27.
- Hu, J., Hu, W., Chou, C., Verhaegen, M., Westwick, D. and Nijssse, G.: Adaptive real-time control of a christmas tree using a novel signal processing approach, *Proceedings of the American Control Conference*, San Diego, California, USA, pp. 2526–2530.

- Wams, B., Nijse, G. and Boom, T. V. D.: A neural-model based robust controller for non-linear systems, *Proceedings of the American Control Conference*, San Diego, California, USA, pp. 4066–4070.

Bibliography

- Åström, K. and Wittenmark, B. (1997), *Computer controlled systems*, Prentice Hall, Inc.
- Basseville, M. and Nikiforov, I. (1993), *Detection of abrupt changes*, Prentice Hall, Inc.
- Bjarnason, E. (1995), ‘Analysis of the Filtered-x LMS Algorithm’, *IEEE Transactions on Speech and Audio Processing* **3**(6), 504–514.
- Brewer, J. W. (1978), ‘Kronecker products and matrix calculus in system theory’, *IEEE Transactions on circuits and systems* (9), 772–781.
- Cabell, R. H. (1998), A Principal Component Algorithm for Feedforward Active Noise and Vibration Control, PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, USA.
- Dewilde, P. and van der Veen, A.-J. (1998), *Time-varying systems and computations*, Kluwer Academic Publishers.
- Doelman, N. (1993), Design of Systems for Active Sound Control, PhD thesis, Delft University of Technology, Delft, The Netherlands.
- Doh, T.-Y., Moon, J.-H. and Chung, M. (1999), ‘An iterative learning control for uncertain systems using structured singular value’, *Transactions of the American Society of Mechanical Engineers* **121**, 660–666.
- Dorf, R. and Bishop, R. (1998), *Modern control systems*, eight edn, Addison-Wesley.
- Douglas, S. (1997), ‘An efficient implementation of the modified Filtered-X LMS algorithm’, *IEEE Signal Processing Letters* **4**(10), 286–288.
- Douglas, S. (1999), ‘Fast implementations of the Filtered-X LMS and LMS algorithms for multichannel active noise control’, *IEEE Transactions on speech and audio processing* **7**(4), 454–465.
- Doyle, J., Francis, B. and Tannenbaum, A. (1992), *Feedback control theory*, Maxwell MacMillan International.
- El-Sinawa, A. and Kashani, R. (2001), ‘Active isolation using a kalman estimator-based controller’, *Journal of vibration control* **7**(8), 1163–1173.

- Elliott, S. (2000a), 'Optimal controllers and adaptive controllers for multichannel feedforward control of stochastic disturbances', *IEEE Transactions on Signal Processing* **48**(4), 1053–1060.
- Elliott, S. (2000b), *Signal processing for active control*, Academic Press.
- Elliott, S. J. (1999), 'Optimal controllers and adaptive controllers for multichannel feedforward control of stochastic disturbances', *IEEE Transactions on signal processing* **48**(4), 1053–1060.
- Enden, A. V. D. (1989), *Discrete-time signal processing*, Prentice Hall, Inc.
- Fraanje, P. (2004), Robust and fast schemes in active noise and vibration control, PhD thesis, University of Twente, P.O. Box 217, 7500 AE, Enschede, The Netherlands.
- Fuller, C., Elliott, S. and Nelson, P. (1996), *Active control of vibration*, Academic Press Limited.
- Golub, G. and Loan, C. V. (1996), *Matrix computations*, The John Hopkins University Press.
- Gustafsson, F. (2000), *Adaptive filtering and change detection*, John Wiley and Sons Ltd.
- Hansen, C. and Snyder, S. (1997), *Active control of noise and vibration*, E & FN Spon.
- Hassibi, B., Sayed, A. and Kailath, T. (1996), 'H[∞] optimality of the lms algorithm', *IEEE Transactions on Signal Processing* **44**(2).
- Haverkamp, B., Chou, C. and Verhaegen, M. (1997), 'SMI toolbox: A matlab toolbox for state space model identification', *Journal A* **38**(3), 34–37.
- Haverkamp, L. (2001), State space identification, PhD thesis, Delft University of Technology, Delft, The Netherlands.
- Haykin, S. (1996), *Adaptive filter theory*, Prentice Hall, Inc.
- Haykin, S. and Widrow, B. (2003), *Least-mean-square adaptive filters*, Wiley-Interscience.
- Horodinca, M., Hanieh, A. A. and Preumont, A. (2002), A soft six degrees of freedom active vibration isolator based on a stewart platform, in 'ACTIVE 2002: The 2002 International Symposium on Active Control of Sound and Vibration', ISVR, Southampton, UK, pp. 1021–1032.
- Hu, J.-S., Yu, S.-H. and HSieh, C.-S. (1998), 'Application of model-matching techniques to feedforward active noise controller design', *IEEE Transactions on Control Systems Technology* **6**(1), 33–42.
- II, L. W. C. (1993), *Digital and analog communication systems*, 4 edn, MacMillan Publishing Company.
- Johansson, S. (1998), Active noise control in aircraft - algorithms and applications, Thesis HKR/R-RES98/3SE, University of Karlskrona/Ronneby, Lund University, Lund/Ronneby, Sweden.

- Kailath, T. and Sayed, A., eds (1999), *Fast reliable algorithms for matrices with structure*, Society for industrial and applied mathematics.
- Kailath, T., Sayed, A. and Hassibi, B. (2000), *Linear estimation*, Prentice Hall, Inc.
- Kim, I.-S., Na, H.-S., Kim, K.-J. and Park, Y. (1994), ‘Constraint Filtered-X and Filtered-U least mean square algorithms for active noise control in ducts’, *Journal of the Acoustical Society of America* **6**(95), 3379–3389.
- Kuo, S. and Morgan, D. (1996), *Active noise control systems*, John Wiley & Sons, Inc.
- Kwakernaak, H. and Sivan, R. (1991), *Modern Signals and Systems*, Prentice-Hall, Inc.
- Leemhuis, A. (2004), Design of the causal wiener controller for active vibration isolation control: a state space approach, Master’s thesis, University of Twente, Faculty of Engineering Technology, Department of Mechanical Engineering, Laboratory of Mechanical Automation.
- Leon-Garcia, A. (1994), *Probability and random processes for electrical engineering*, Addison-Wesley.
- Ljung, L. (1999), *System identification*, Prentice Hall, Inc.
- Ljung, L. and Glad, T. (1994), *Modeling of dynamic systems*, Prentice Hall, Inc.
- Mead, D. J. (1998), *Passive vibration control*, John Wiley and Sons.
- Miyoshi, M. and Kaneda, Y. (1988), ‘Inverse filtering of room acoustics’, *IEEE Transactions on Acoustics, Speech and Signal processing* **36**(2), 145–152.
- Morari, M. and Zafriou, E. (1989), *Robust process control*, Prentice-Hall, Inc.
- Nascimento, V. H. and Sayed, A. H. (1998), Are ensemble-average learning curves reliable in evaluating the performance of adaptive filters?, in ‘Asilomar Conference on Signals, Systems, and Computers’, Pacific Grove, California, USA, pp. 1171–1175.
- Nelson, P. and Elliott, S. (1992), *Active control of sound*, Academic Press Limited.
- Nelson, P., Orduña-Bustamante, F. and Hamada, H. (1995), ‘Inverse filter design and equalization zones in multichannel sound reproduction’, *IEEE Transactions on Speech and Audio Processing* **3**(3), 185–192.
- NICONET, I. S. (2000), ‘The SLICOT package - task III.A: Development of standard software for linear, time-invariant state space model identification’. Katholieke Universiteit Leuven, Department of Electrical Engineering, Kasteelpark Arenberg 10, 3001, Leuven-Heverlee, Belgium.
- Niewold, F. (2004), H_2 controller design with the active vibration control toolbox for Matlab, Master’s thesis, University of Twente, Faculty of Engineering Technology, Department of Mechanical Engineering, Laboratory of Mechanical Automation.

- Nijssse, G., Super, H., Dijk, J. V. and Jonker, J. (2004), Feedforward control of broadband disturbances on a six-degrees-of-freedom vibration isolation set-up, *in* 'ACTIVE 2004: The 2004 International Symposium on Active Control of Sound and Vibration', Williamsburg, Virginia, USA.
- Nijssse, G., Super, H., Dijk, J. V. and Jonker, J. (2005), 'A robust subspace based approach to feedforward control of broadband disturbances on a novel six-degrees-of-freedom vibration isolation set-up', *Noise Control Engineering Journal* **53**(5), 207–215.
- Nijssse, G., Verhaegen, M. and Doelman, N. (2001a), A new subspace based approach to iterative learning control, *in* 'Proceedings of the European Control Conference', Seminário de Villar, Porto, Portugal.
- Nijssse, G., Verhaegen, M. and Doelman, N. (2001b), State space modeling of acoustics for active control, *in* 'Inter-Noise 2001: The 2001 International Congress and Exhibition on Noise Control Engineering', The Hague, The Netherlands.
- Nijssse, G., Verhaegen, M., Schutter, B. D., Westwick, D. and Doelman, N. (1999), State space modeling in multichannel active control systems, *in* 'ACTIVE 99: The 1999 International Symposium on Active Control of Sound and Vibration', Fort Lauderdale, Florida, USA, pp. 909–920.
- Oarâ, C. and Varga, A. (1999), The general inner-outer factorization problem for discrete-time systems, *in* 'Proceedings of The European Control Conference', Karlsruhe, Germany.
- Oppenheim, A. and Shafer, R. (1975), *Digital signal processing*, Prentice Hall, Inc.
- Phan, M. and Frueh, J. (1998), *Iterative Learning Control*, Kluwer Academic Publishers, chapter 15 (System Identification and Learning Control).
- Phillips, C. and Nagle, H. (1995), *Digital Control System Analysis and Design*, Prentice Hall, Inc.
- Preumont, A. (2002), *Vibration control of active structures*, second edn, Kluwer Academic Publishers.
- Price, D., Wilde, J. D., Papadaki, A., Curran, J. and Kitney, R. (2001), 'Investigation of acoustic noise on 15 MRI scanners from 0.2t to 3t', *Journal of magnetic resonance imaging* **13**(2), 288–293.
- Rafaely, B. and Elliott, S. (1999), ' H_2/H_∞ active control of sound in a headrest: Design and implementation', *IEEE Transactions on Control Systems Technology* **7**(1), 79–84.
- Redfern, D. and Campbell, C. (1998), *The MATLAB 5 handbook*, Springer Verlag.
- Regalia, P. A. (1995), *Adaptive IIR filtering in signal processing and control*, Marcel Dekker, Inc.
- Roberts, R. and Mullis, C. (1987), *Digital Signal Processing*, Addison Wesley.
- Roozen, N., Philips, P., Biloen, D., Limpens, P. and Tuithof, H. (2005), Active vibration isolation applied to a magnetic resonance imaging (MRI) system, *in* 'Twelfth International Congress on Sound and Vibration', Lisbon, Portugal.

- Rugh, W. J. (1996), *Linear system theory*, second edn, Prentice Hall, Inc.
- Rupp, M. and Sayed, A. H. (1996), ‘A time-domain feedback analysis of filtered-error adaptive gradient algorithms’, *IEEE Transactions on Signal Processing* **44**(6), 1428–1439.
- Rupp, R. and Sayed, A. (1998), ‘Robust FxLMS algorithms with improved convergence performance’, *IEEE Transactions on Speech and Audio Processing* **6**(1), 78–85.
- Sayed, A. H. (2003), *Fundamentals of adaptive filtering*, John Wiley & Sons.
- Schofield, B. (2003), Subspace based identification for adaptive control, Master’s thesis, Department of Automatic Control, Lund Institute of Technology, Sweden.
- Strang, G. (1988), *Linear algebra and its applications*, Harcourt Brace Jovanovich, Inc.
- Super, H. (2006), Hybrid vibration isolation of a six-degrees-of-freedom experimental setup, PhD thesis, University of Twente, Enschede, The Netherlands.
- Thoeni, U. (1994), *Programming real-time multicomputers for signal processing*, International Series in Acoustics, Speech and Signal Processing, Prentice Hall, Inc.
- Tokuno, H., Kirkeby, O., Nelson, P. A. and Hamada, H. (1997), ‘Inverse filter of sound reproduction systems using regularization’, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* **E80-A**(5), 801–819.
- van der Hoeven, S. (2002), Prefiltering of the fxlms algorithm applied to a vibrating plate, Master’s thesis, University of Twente, Faculty of Applied Physics, Systems and Control engineering Laboratory.
- Van Overschee, P. and De Moor, B. (1994), ‘N4SID: Subspace algorithms for the identification of combined deterministic and stochastic systems’, *Automatica* **30**(1), 75–93.
- van Overschee, P. and de Moor, B. (1996), *Subspace identification for linear systems*, Kluwer Academic Publishers.
- Verhaegen, M. (1993), ‘Subspace model identification part 3. Analysis of the ordinary output-error state-space model identification algorithm’, *International Journal of Control* **56**(3), 555–586.
- Verhaegen, M. (1994), ‘Identification of the deterministic part of MIMO state space models given in innovations form from input-output data’, *Automatica* **30**(1), 61–74.
- Verhaegen, M. and Dewilde, P. (1992), ‘Subspace model identification part 2. Analysis of the elementary output-error state-space model identification algorithm’, *International Journal of Control* **56**(5), 1211–1241.
- Verhaegen, M., Varga, A. and Grbel, G. (1994), Some experience with the MOESP class of subspace model identification methods identifying the bo105 helicopter, Technical report TR R165-94, DLR-Oberpfaffenhofen.
- Viberg, M. (1995), ‘Subspace-based methods for the identification of linear time-invariant systems’, *Automatica* **31**(12), 1835–1851.

- Vidyasagar, M. (1985), *Control system synthesis: a factorization approach*, MIT Press.
- Visser, J., Sridhar, J. and Mulder, J. (1996), Comparison of prediction error and subspace identification methods - aircraft and helicopter applications, Memorandum M-720, Faculty of Aerospace Engineering, Delft University of Technology, Delft, The Netherlands.
- Wan, E. A. (1996), Adjoint LMS: an efficient alternative to the filtered-x LMS and multiple error LMS algorithms, in 'ICASSP: Proceedings of the international conference on acoustics, speech and signal processing', Atlanta, Georgia, USA.
- Wang, A. K. and Ren, W. (1999), 'Convergence analysis of the multi-variable filtered-x lms algorithm with application to active noise control', *IEEE Transactions on Signal Processing* **47**(4), 1166–1169.
- Westbrook, C. (2001), *Handbook of MRI technique*, second edition edn, Blackwell Science.
- Widrow, B. and Walach, E. (1996), *Adaptive inverse control*, Prentice Hall, Inc.
- Wissenburg, C. (1991), *Inleiding in de electronica*, Delftse uitgevers maatschappij B.V.
- Woodward, P. (2001), *MRI for technologists*, second edition edn, McGraw-Hill.
- Youla, D., Bongiorno, J. and Jabr, H. (1976a), 'Modern Wiener-Hopf design of optimal controllers, pt. 1, The single-input-single-output case', *IEEE Transactions on Automatic Control* **3**(21), 3–13.
- Youla, D., Bongiorno, J. and Jabr, H. (1976b), 'Modern Wiener-Hopf design of optimal controllers, pt. 2, The multiple-input-multiple-output case', *IEEE Transactions on Automatic Control* **3**(21), 319–338.

Acknowledgements

Many thanks go to my thesis supervisors professor Ben Jonker and assistant professor Johannes van Dijk for giving me the opportunity for the establishment of the Ph.D. thesis. Thank you Ben and Johannes, for the trust which you put in me and for the pleasant working atmosphere in the group. Especially for Johannes, my daily supervisor, thanks for all the pleasant discussions we had.

I also owe much gratitude to professor Maarten Steinbuch from Eindhoven University of Technology, for motivating me to finish the thesis and for taking place in the thesis committee. Without the discussions I had with you, Maarten, this thesis would not have been accomplished at all. Thanks also for Hiddo Super, my colleague Ph.D. student. I am particularly grateful for providing me with your knowledge about the dSPACE systems, with helping me out with my real-time control problems and for being ‘paranimf’.

Special thanks go to former colleagues Bas Benschop, Ton van den Boom, Niek Doelman, Stoyan Kanev, Hiroshi Oku, Bart De Schutter, Stefano Stramigioli, Henk Verbruggen and Vincent Verdult.

Former supervisor professor Michel Verhaegen, thanks for providing me with research questions. Furthermore, I would like to thank assistant professor Arthur Berkhoff for the time and humor we had together, for the valuable comments on the thesis manuscript and for taking place in the thesis committee.

Former colleague Rufus Fraanje of the SCE group, of course I will not forget you. Thank you for the time we had together and for the numerous research discussions we had. We definitely learned a lot about adaptive algorithms in front of the white board while writing out equations and block diagrams.

A number of master’s thesis students contributed to this thesis: Arjen Duindam, Max Groenendijk, Wouter Hagedoorn, Jonathan Hofman, Klaas Kole, Anton Leemhuis, Frans Niewold, Dick Petersen, Michiel Post and Arno Pross. I appreciate the hard work you put into it!

I also appreciate the work of the external members of the thesis committee: professors André de Boer and Arjan van der Schaft. Also, ‘paranimf’ Gonny van der Zwaag, thanks for helping me out.

Finally, I want to thank my parents for always driving and motivating me to strive for the best in life. Thank’s mum and dad! Also, my sister Géke Jansen has been a valuable ‘asset’ during my Ph.D. time. Together with my parents, she always knew how to motivate me, even when times were tough.

Curriculum Vitae

Gerard Nijse was born on August 19, 1974 in Ede, The Netherlands. He received his secondary education at the Van Lodenstein college in Amersfoort and graduated in 1993. From 1993 to 1998 he worked on his master's degree in Electrical Engineering at the Delft University of Technology and he received his degree with honors. His thesis dealt with the robustness analysis of control systems based on neural network models and resulted in a first publication at the American Control Conference in 1999. After finishing his master's he became a Ph.D. student. Supervised by Michel Verhaegen he started on a new class of active control algorithms, which were based on state space models. He spend one year working in Delft and continued his thesis at the Faculty of Applied physics of the University of Twente in September 1999, where a project started on noise and vibration control in cooperation with the TNO Institute of Applied Physics, Delft. In May 2002 he made a switch to the Faculty of Engineering Technology of the same university, where Ben Jonker became his supervisor. He obtained his Ph.D. degree in February 2006. During his Ph.D. Gerard obtained a certificate from the Dutch Institute of Systems and Control for successfully following five post-graduate courses in systems and control engineering.