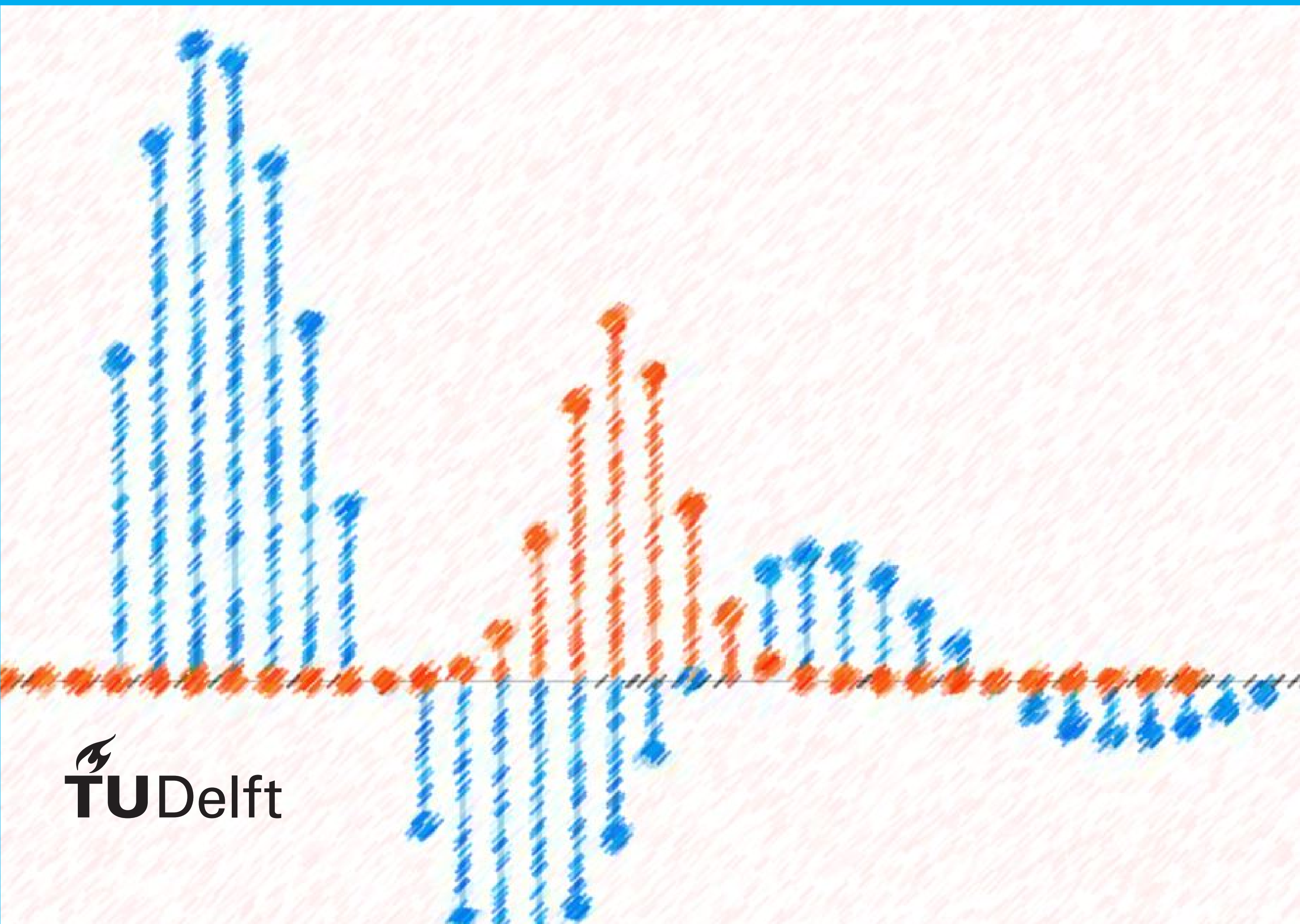
Time Domain Force Identification

for Noise and Vibration Prediction in Vehicles

T.N.J. Geelen

MSc Thesis



TU Delft

Time Domain Force Identification

for Noise and Vibration Prediction in Vehicles

by

T.N.J. Geelen

Master of Science Thesis
at the Delft University of Technology,
to be defended publicly on july 3, 2019

Student number:	4294866	
Project duration:	January 1, 2018 – July 1, 2019	
Thesis committee:	dr. R. Happee,	TU Delft, supervisor
	ir. D.D. van den Bosch,	VIBES.technology
	dr. M. van der Seijs,	VIBES.technology

TUDelft

Abstract

In many engineering fields it is beneficial to obtain information about the force acting on a dynamical system. As measurement of this force is often difficult or impossible a technique that identifies this force in an alternative manner is desired. A wide variety of methods is available in literature. Obtaining this force in the frequency domain is done often. However, in certain cases where the input force is non-stationary a frequency domain technique does not suffice. This thesis therefore focuses on obtaining a reliable force identification method in the time domain.

The force identification problem can be seen as an 'inverse problem' to which a simple analytical solution is not trivial. A more advanced method is required. Methods found in literature can be grouped into three categories which fundamentally differ in the way the dynamics is modelled. Deterministic force identification methods are defined as methods where the dynamics is modelled deterministically. Whenever a methods uses a stochastic model it is considered a stochastic force identification method. A third group of force identification methods uses artificial intelligence to obtain a model of the system when no model is available. In this thesis it is assumed a model of the system dynamics is available and therefore artificial intelligence methods for force identification are not considered.

Deterministic force identification method including regularization methods, recursive methods and iterative methods are compared to stochastic methods which are all based on the Kalman filter. The most relevant methods are evaluated using simulated data of a single and multiple degree of freedom dynamical system and measured performed on an aluminium structure.

It was concluded that the Least Mean Square Adaptive Algorithm outperforms the Joint Input-State Estimator with Artificial Displacement Measurements in identifying forces acting on the simulated single and multiple degree of freedom system as well as the forces acting on the aluminium structure.

Nomenclature

Notation

x	Scalar
$\boldsymbol{x}$	Vector
$\boldsymbol{X}$	Matrix

Fixed symbols

$\boldsymbol{M}$	Mass matrix
$\boldsymbol{C}$	Damping matrix
$\boldsymbol{K}$	Stiffness matrix
$\boldsymbol{u}$	Displacements
$\boldsymbol{f}$	Forces
$\boldsymbol{d}$	Measurements
$\boldsymbol{Y}$	FRFs
$\boldsymbol{H}$	IRFs
$\boldsymbol{x}$	State
$\boldsymbol{A}$	System matrix
$\boldsymbol{B}$	Input matrix
$\boldsymbol{G}$	Observation matrix
$\boldsymbol{J}$	Direct feedthrough matrix
$\boldsymbol{I}$	Identity matrix
$\boldsymbol{w}$	System noise
$\boldsymbol{v}$	Measurement noise
t	Continuous time
k	Discrete time
N	Size of the input vector
O	Size of the IRF

Operators

$\dot{\boldsymbol{x}}$	First order time derivative of $\boldsymbol{x}$
$\ddot{\boldsymbol{x}}$	Second order time derivative of $\boldsymbol{x}$
$\boldsymbol{X}^{-1}$	Inverse of matrix $\boldsymbol{X}$
$\boldsymbol{X}^T$	Transpose of matrix $\boldsymbol{X}$
$rank(\boldsymbol{X})$	Rank of matrix $\boldsymbol{X}$
$tr(\boldsymbol{X})$	Trace of matrix $\boldsymbol{X}$
$\hat{\boldsymbol{x}}$	Estimate of $\boldsymbol{x}$
$cov(\boldsymbol{X})$	Covariance of matrix $\boldsymbol{X}$
$\|\|\boldsymbol{x}\|\|_p$	p-norm of $\boldsymbol{x}$

Abbreviations

ADM	Artificial Displacement Measurement
AKF	Augmented Kalman Filter
CDP	Classical Dynamic Programming
DKF	Dual Kalman Filter
DoF	Degree of Freedom
DoFs	Degrees of Freedom
DP	Dynamic Programming
DS	Dynamic Substructuring
EDP	Extended Dynamic Programming
EKF	Extended Kalman Filter
FRF	Frequency Response Runction
GIRLS	Generalized Iterative Reweighted Least Squares
GTR	Generalized Tikhonov Regularization
IRF	Impulse Fesponse Function
JISE	Joint Input-State Estimator
LMS	Least Mean Square
LS	Least Squares
LTI	Linear Time-Invariant
MDoF	Multi Degree of Freedom
MIMO	Multi-Input-Multi-Output
MLS	Moving Least Squares

MVU	Minimum-Variance Unbiased
RMSE	Root Mean Square Error
SDoF	Single Degree of Freedom
SI	Subspace Identification
SISO	Single-Input-Single-Output
SM	Sensitivity Method
SNR	Signal-to-Noise Ratio
SVD	Singular Value Decomposition
TPA	Transfer Path Analysis
TR	Tikhonov Regularization
TSVD	Truncated Singular Value Decomposition
TVR	Total Variation Regularization
VPT	Virtual Point Transformation

Contents

List of Figures

List of Tables

1 Introduction

In many engineering fields it is desired to obtain information about the forces acting on a dynamical system. As measurement of this force is often difficult or impossible a technique that identifies this force in an alternative manner is required. The forces acting on the system can be seen as the input. The description of the system itself defines its dynamics: the relation between the input and the response or the output of the system. In some cases these forces can be measured directly for example by using load cells. However when direct measurements are impossible or difficult to execute a different approach is needed. Measurements of the dynamic response of the system on these forces often is possible. These measurements can be used to reconstruct the input forces.

In Structural Health Monitoring (SHM), the damage-inducing force is of interest of which direct measurement is often not possible. The location of the measurements are inaccessible and therefore reconstruction is needed [16, 27, 31]. It can also be that, due to the nature of a force (e.g. wind force acting on a wind turbine) measurements are impossible [44]. In fatigue life assessment time history of the external forces is important to determine the remaining fatigue life, however measurement is impossible due to sensor limitations and the unknown nature of the external forces [6, 52]. Level-ice forces are measured to monitor vibrations in the ice. However non-collocated sensors are the only realistic option. Therefore identification of the forces is required [49]. In the automotive industry, the forces acting on a wheel are important to know for example for safety and performance but are expensive to measure. Reconstruction would be advantageous [53]. An aspect that plays a role in measuring forces lies in the nature of the load cell. The measured quantity is strain and the force is determined by the relation between stress and strain (Hooke's law). To avoid large displacements the stiffness of the load cell itself must be relatively large. This extra stiffness may have a significant influence on the dynamics of the system [71].

Finally, in the automotive industry comfort is an aspect that is of increasing importance. Active components such as engines, turbo's and electromotors can cause vibrations resulting in unwanted noise. To eliminate these vibrations in an early stage in the design process effective modeling of the entire system is needed. Dynamic Substructuring (DS) [28] is a technique that analyses the dynamic behavior of a complex system by analyzing elements of the system separately. The vibrations induced in the system can be determined with a method known as Transfer Path Analysis (TPA) [65] describing the transmission of vibrations through the elements. To determine the vibrations propagating from one element in the system to another element, the elements will be coupled for which the forces at the interfaces are required. This can be (and is) done in the frequency domain very effectively. However, non-stationary effects are not fully captured in the frequency domain. A time-domain approach is needed. A time domain method for force identification can also be used for (real time) auralization where the sound produced by a vibration is modeled. With this technique one can hear for example the influence of changing a certain component of a vehicle. This literature research focuses on the state of the art of force identification in the time domain applicable for TPA.

1.1. Requirements

In order for a force identification method to be applicable for TPA in the time domain it has to meet the requirements listed below.

- *The method should be able to handle multi-input-multi-output (MIMO) systems.* As the to be analyzed structures are often complex of nature, their response is captured by more than one sensor resulting in a multi-output system. Also, most components in a vehicle (e.g. an engine) are constrained in more than one position, the forces of this active component result in a multi-input system.
- *It should be able to reconstruct non-stationary forces.* Current technology using frequency domain techniques is not accurate enough when identifying non-stationary forces.
- *It should be able to handle noisy signals.* All measurements inherently include noise which can influence the identification process if not handled correctly. A signal-to-noise ratio (SNR) of 100 is common and an SNR of 10 may occur.
- *It should (at least) be able to handle acceleration response measurements.* TPA is often done with accelerometers as velocity and displacement sensors can give rise to difficulties.
- *It should allow for non-collocated input-measurement pairs.* Often response measurement at the input location is not possible.
- *It should reconstruct forces with acceptable accuracy.* A certain quality of the reconstructed force is needed to be succesfully applied in TPA. A definition of "*acceptable*" will be obtained in the research following this literature research.
- *It should reconstruct forces within acceptable computation time.* A definition of "*acceptable*" will be obtained in the research following this literature research.

1.2. Objectives

The following list consists of the objective of this literature research and its following research.

1. Collect available techniques in literature.
2. Investigate methods to evaluate different techniques objectively.
3. Evaluate available techniques.
4. Implement most applicable techniques for SDoF system.
5. Implement most applicable techniques for MDoF system.
6. Implement most applicable techniques for semi-academic test case.
7. Evaluate results.

1.3. Authors Contributions

1. Artificial Displacement Measurements for the Joint Input-State Estimator
2. Extension to the Least Mean Square Adaptive Algorithm
3. Virtual Point Transformation in state space

I

State of the Art

Part I of this thesis will elaborate on the current state of the art concerning force identification. As such it can be considered a literature study which presents an overview of different force identification methods that can be found in literature.

Identification methods can be grouped into three categories: deterministic methods, stochastic methods and methods that use artificial intelligence. These three groups characterise themselves by the way the system dynamics is defined. Most attention is given to the deterministic and stochastic methods as they are most applicable for TPA.

The first chapter of part I will focus on the different ways a dynamical system can be described. The second chapter sketches the fundamental problem of force identification which can be traced back to a problem referred to as the inverse problem. The following two chapters present deterministic and stochastic force identification methods respectively. Part I is concluded by a comparison of the presented methods with respect to the list of requirement defined in the introduction.

2

Representations of a Dynamical System

This chapter shortly introduces different ways to mathematically describe a dynamical system. Throughout this thesis multiple of these representations will be used in different force identification methods. Also, the system is assumed to be linear and time-invariant (LTI). Eventually these models will have to be implemented using measured data. As this data is never continuous a conversion of the system description to discrete time is needed. Therefore, a description of the discretization is given.

2.1. Physical Domain

A dynamical system can be represented as a combination of masses, springs and dampers on which excitations are present. This is shown mathematically in equation 2.1.

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}\dot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = \boldsymbol{S}_p\mathbf{f}(t) \tag{2.1}$$

where $\mathbf{u}(t)$ is the vector of displacements of the degrees of freedom (DoFs) of the system, $\dot{\mathbf{u}}(t)$ and $\ddot{\mathbf{u}}(t)$ its time-derivatives for the velocity and acceleration. $\mathbf{M}$, $\mathbf{C}$ and $\mathbf{K}$ are the mass, damper and stiffness matrix, respectively. The vector $\mathbf{f}(t)$ is the input force vector and matrix $\boldsymbol{S}_p$ maps the forces of vector $\boldsymbol{f}(t)$ to the degrees of freedom in $\boldsymbol{u}(t)$. Discretization can be done via a numerical integration method such as Euler's method or the Newmark-β method which is done in section 3.1.3

2.2. Frequency Domain

When looking at equation 2.1, applying the Fourier transform on the excitation and force vectors the following equation is obtained where the excitations and force vectors are dependent on the frequency (ω):

$$\boldsymbol{M}\ddot{\boldsymbol{u}}(\omega) + \boldsymbol{C}\dot{\boldsymbol{u}}(\omega) + \boldsymbol{K}\boldsymbol{u}(\omega) = \boldsymbol{f}(\omega) \tag{2.2}$$

Using the fact that $\dot{\boldsymbol{u}}(\omega) = j\omega\boldsymbol{u}(\omega)$ and $\ddot{\boldsymbol{u}}(\omega) = -\omega^2\boldsymbol{u}(\omega)$, a matrix $\boldsymbol{Z}(\omega)$ dependent on frequency alone can be obtained:

$$(-\omega^2\boldsymbol{M} + j\omega\boldsymbol{C} + \boldsymbol{K})\boldsymbol{u}(\omega) = \boldsymbol{f}(\omega) \tag{2.3}$$

$$\boldsymbol{Z}(\omega)\boldsymbol{u}(\omega) = \boldsymbol{f}(\omega) \tag{2.4}$$

Matrix $\boldsymbol{Z}(\omega)$ is called the dynamic stiffness matrix and describes the forces needed for a unit harmonic response on one DoF while all other DoFs are constrained. A much more intuitive and much more used notation uses the inverse of matrix $\boldsymbol{Z}(\omega)$:

$$\boldsymbol{u}(\omega) = \boldsymbol{Y}(\omega)\boldsymbol{f}(\omega), \quad \boldsymbol{Y}(\omega) = \left(\boldsymbol{Z}(\omega)\right)^{-1} \tag{2.5}$$

where $\boldsymbol{Y}(\omega)$ represents the Frequency Response Functions (FRFs) of the system. This can be physically interpreted as the ratio between the amplitude of the input and the output for each frequency.

2.3. Modal Domain

The dynamics of an LTI system can also be described in terms of its natural vibration modes. These natural vibration modes or eigenmodes are shapes at which a system vibrates for which the inertial forces are in harmony with the elastic forces in the absence of damping. Mathematically this translates to:

$$(\boldsymbol{K} - \omega_r^2 \boldsymbol{M})\boldsymbol{\Phi}_r = 0 \tag{2.6}$$

where ω_r are the 1 to r natural frequencies or eigenfrequencies of the system and $\boldsymbol{\Phi}$ is a matrix which columns are the modes shapes. Typically the columns of $\boldsymbol{\Phi}$ are in ascending order of the eigenfrequencies, mass-normalized and orthogonal such that:

$$\begin{cases} \boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\Phi}_1 & \boldsymbol{\Phi}_2 & \cdots & \boldsymbol{\Phi}_n \end{bmatrix} \\ \boldsymbol{\Phi}^T \boldsymbol{M} \boldsymbol{\Phi} = \boldsymbol{I} \\ \boldsymbol{\Phi}^T \boldsymbol{K} \boldsymbol{\Phi} = diag(\omega_1^2, \omega_2^2, \ldots, \omega_n^2) \end{cases} \tag{2.7}$$

Whenever higher frequencies play a less significant role they and can be ignored. This is done by truncating the modal matrix $\boldsymbol{\Phi}$ resulting in a reduction matrix $\boldsymbol{R}$:

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{\Phi}_1 & \boldsymbol{\Phi}_2 & \cdots & \boldsymbol{\Phi}_m \end{bmatrix} \tag{2.8}$$

with $m < n$. The displacement vector can now be approximated by modal displacement vector $\boldsymbol{\eta}(t)$:

$$\boldsymbol{u}(t) \approx \boldsymbol{R}\boldsymbol{\eta}(t) \tag{2.9}$$

Using this in equation 2.1 gives the equations of motion in terms of the modal coordinates:

$$\boldsymbol{M}\boldsymbol{R}\ddot{\boldsymbol{\eta}}(t) + \boldsymbol{C}\boldsymbol{R}\dot{\boldsymbol{\eta}}(t) + \boldsymbol{K}\boldsymbol{R}\boldsymbol{\eta}(t) = \boldsymbol{S}_p \boldsymbol{f}(t) + \boldsymbol{r}(t) \tag{2.10}$$

Because $\boldsymbol{R}\boldsymbol{\eta}(t)$ is an approximation of $\boldsymbol{u}(t)$ it is possible that the external forces do not exactly equal the inertial, damping and elastic forces. Therefore a residual term $\boldsymbol{r}(t)$ is added. To compensate for this residual term, the equations of motion are pre-multiplied by $\boldsymbol{R}^T$ ensuring that $\boldsymbol{R}^T \boldsymbol{r} = 0$. With this pre-multiplication, the external forces are also described in terms of the modal coordinates. This results in the following modally reduced equations of motion:

$$\boldsymbol{M}_m \ddot{\boldsymbol{\eta}}(t) + \boldsymbol{C}_m \dot{\boldsymbol{\eta}}(t) + \boldsymbol{K}_m \boldsymbol{\eta}(t) = \boldsymbol{f}_m(t) \tag{2.11}$$

$$\boldsymbol{M}_m = \boldsymbol{R}^T \boldsymbol{M} \boldsymbol{R} \tag{2.12}$$

$$\boldsymbol{C}_m = \boldsymbol{R}^T \boldsymbol{C} \boldsymbol{R} \tag{2.13}$$

$$\boldsymbol{K}_m = \boldsymbol{R}^T \boldsymbol{K} \boldsymbol{R} \tag{2.14}$$

$$\boldsymbol{f}_m = \boldsymbol{R}^T \boldsymbol{f} \tag{2.15}$$

These modally reduced equations of motion approximate the true equations of motion but reduce the number of degrees of freedom which decreases computation time.

2.4. Time Domain

Another representation that fully describes the behavior of an LTI system is the systems Impulse Response Functions (IRFs). This is defined as the time domain response of the system to a Dirac impulse. The Dirac impulse is a theoretical signal that can be seen as a signal with infinitely high gain and infinitely short duration of which the integral over infinite time is one. The response of an LTI system described by its IRFs is given in the equation below.

$$\mathbf{u}(t) = \mathbf{H}(t) \circledast \mathbf{f}(t) = \int_0^t \mathbf{H}(t-\tau)\mathbf{f}(\tau)d\tau \tag{2.16}$$

where $\mathbf{H}(t)$ is a matrix containing the IRFs of the system and $\circledast$ stands for the mathematical convolution operation. The last part of equation 2.16 is referred to as Duhamel's integral. A way to discretize the convolution integral is as follows:

$$\boldsymbol{u}(k) = \sum_{i=0}^{k-1} \boldsymbol{H}(k-i)\boldsymbol{f}(i)\Delta t \tag{2.17}$$

This implies that the response $\boldsymbol{u}(n)$ depends only on the force at time t_0 whereas, assuming piece-wise linear approximation of the force, the response also depends on the force at time t_1. Averaging the force over t_0 and t_1 would therefore be better to determine an estimation of the response. Another way to discretize the convolution integral therefore is:

$$\boldsymbol{u}(k) = \sum_{i=0}^{k-1} \boldsymbol{H}(k-i) \frac{\boldsymbol{f}(i) + \boldsymbol{f}(i+1)}{2} \Delta t \tag{2.18}$$

This can be conveniently written as:

$$\boldsymbol{u}(k) = \sum_{i=0}^{k-1} \boldsymbol{H}(k-i) \boldsymbol{f}(i + \frac{1}{2}) \Delta t \tag{2.19}$$

This shows that the summation is shifted a half time step forward compared to equation 2.17. An alternative way of approaching the discretized convolution integral is:

$$\boldsymbol{u}(k) = \sum_{i=0}^{k-1} \frac{\boldsymbol{H}(k-i)\boldsymbol{f}(i) + \boldsymbol{H}(k-i-1)\boldsymbol{f}(i+1)}{2} \Delta t \tag{2.20}$$

To show the dependence of the response on the different time steps the summation can be written as:

$$\boldsymbol{u}(k) = \boldsymbol{H}(k)\boldsymbol{f}(0)\frac{\Delta t}{2} + \sum_{i=1}^{k-1} \boldsymbol{H}(k-i)\boldsymbol{f}(i)\Delta t + \boldsymbol{H}(0)\boldsymbol{f}(k)\frac{\Delta t}{2} \tag{2.21}$$

This notation will be used in section 3.1.1 to derive an analytical solution to the force identification problem.

2.5. State Space Domain

As seen in the beginning of this chapter equation 2.1 is a second order differential equation. With the use of the state space theory it is possible to write this equation as a first order differential equation. A state $\boldsymbol{x}$ is defined containing the position and velocity:

$$\boldsymbol{x}_1(t) = \boldsymbol{u}(t) \tag{2.22}$$
$$\boldsymbol{x}_2(t) = \dot{\boldsymbol{u}}(t) \tag{2.23}$$

From this a first order differential equation can be obtained:

$$\dot{\boldsymbol{x}}_1(t) = \boldsymbol{x}_2(t) \tag{2.24}$$

Also:

$$\dot{\boldsymbol{x}}_2(t) = \ddot{\boldsymbol{u}}(t) \tag{2.25}$$

Substituting this into equation 2.1:

$$\boldsymbol{M}\dot{\boldsymbol{x}}_2(t) + \boldsymbol{C}\boldsymbol{x}_2(t) + \boldsymbol{K}\boldsymbol{x}_1(t) = \boldsymbol{S}_p\boldsymbol{f}(t) \tag{2.26}$$

This results in a second first order differential equation:

$$\dot{\boldsymbol{x}}_2 = \boldsymbol{M}^{-1}\boldsymbol{S}_p\boldsymbol{f}(t) - \boldsymbol{M}^{-1}\boldsymbol{C}\boldsymbol{x}_2(t) - \boldsymbol{M}^{-1}\boldsymbol{K}\boldsymbol{x}_1(t) \tag{2.27}$$

Equation 2.24 and 2.27 can be put together in matrix form:

$$\begin{bmatrix} \dot{\boldsymbol{x}}_1(t) \\ \dot{\boldsymbol{x}}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & \boldsymbol{I} \\ -\boldsymbol{M}^{-1}\boldsymbol{K} & -\boldsymbol{M}^{-1}\boldsymbol{C} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1(t) \\ \boldsymbol{x}_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ -\boldsymbol{M}^{-1}\boldsymbol{S}_p \end{bmatrix} \boldsymbol{f}(t) \tag{2.28}$$

This describes the progression of the dynamics of the system over time. The observation or measurement on the system can be described using selection matrices $\boldsymbol{S}_d$, $\boldsymbol{S}_v$ and $\boldsymbol{S}_a$:

$$\boldsymbol{d}(t) = \begin{bmatrix} \boldsymbol{S}_d\boldsymbol{u}(t) \\ \boldsymbol{S}_v\dot{\boldsymbol{u}}(t) \\ \boldsymbol{S}_a\ddot{\boldsymbol{u}}(t) \end{bmatrix} \tag{2.29}$$

Or in terms of the state $\boldsymbol{x}$:

$$\boldsymbol{d}(t) = \begin{bmatrix} \boldsymbol{S}_d \boldsymbol{x}_1 \\ \boldsymbol{S}_v \boldsymbol{x}_2 \\ -\boldsymbol{S}_a \boldsymbol{M}^{-1} \boldsymbol{K} \boldsymbol{x}_1 - \boldsymbol{S}_a \boldsymbol{M}^{-1} \boldsymbol{C} \boldsymbol{x}_2 + \boldsymbol{S}_a \boldsymbol{M}^{-1} \boldsymbol{S}_p \boldsymbol{f}(t) \end{bmatrix} \tag{2.30}$$

$$\boldsymbol{d}(t) = \begin{bmatrix} \boldsymbol{S}_d & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{S}_v \\ -\boldsymbol{S}_a \boldsymbol{M}^{-1} \boldsymbol{K} & -\boldsymbol{S}_a \boldsymbol{M}^{-1} \boldsymbol{C} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1(t) \\ \boldsymbol{x}_2(t) \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{S}_a \boldsymbol{M}^{-1} \boldsymbol{S}_p \end{bmatrix} \boldsymbol{f}(t) \tag{2.31}$$

The dynamics (equation 2.28) and observation (equation 2.31) put together give the state space equations:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}_c \mathbf{x}(t) + \mathbf{B}_c \mathbf{f}(t) \\ \mathbf{d}(t) = \mathbf{G}_c \mathbf{x}(t) + \mathbf{J}_c \mathbf{f}(t) \end{cases} \quad \text{with} \quad \mathbf{x}(\mathbf{t}) = \begin{bmatrix} \mathbf{u}(t) \\ \dot{\mathbf{u}}(t) \end{bmatrix} \tag{2.32}$$

$$\boldsymbol{A}_c = \begin{bmatrix} 0 & \boldsymbol{I} \\ -\boldsymbol{M}^{-1} \boldsymbol{K} & -\boldsymbol{M}^{-1} \boldsymbol{C} \end{bmatrix} \tag{2.33}$$

$$\boldsymbol{B}_c = \begin{bmatrix} 0 \\ \boldsymbol{M}^{-1} \boldsymbol{S}_{\boldsymbol{p}} \end{bmatrix} \tag{2.34}$$

$$\boldsymbol{G}_c = \begin{bmatrix} \boldsymbol{S}_d & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{S}_v \\ -\boldsymbol{S}_a \boldsymbol{M}^{-1} \boldsymbol{K} & -\boldsymbol{S}_a \boldsymbol{M}^{-1} \boldsymbol{C} \end{bmatrix} \tag{2.35}$$

$$\boldsymbol{J}_c = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{S}_a \boldsymbol{M}^{-1} \boldsymbol{S}_p \end{bmatrix} \tag{2.36}$$

Discretized with a time step of h the state space equations become [4]:

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{f}(k) \\ \mathbf{d}(k) = \mathbf{G}\mathbf{x}(k) + \mathbf{J}\mathbf{f}(k) \end{cases} \tag{2.37}$$

with

$$\boldsymbol{A} = e^{\mathbf{A}_c h} \tag{2.38}$$

$$\mathbf{B} = \int_0^h e^{\mathbf{A}_c s} ds \mathbf{B}_c = (\mathbf{A}_c - \mathbf{I}) \mathbf{A}_c^{-1} \mathbf{B}_c \tag{2.39}$$

$$\boldsymbol{G} = \boldsymbol{G}_c \tag{2.40}$$

$$\boldsymbol{J} = \boldsymbol{J}_c \tag{2.41}$$

The state space notation of a dynamical system is convenient because the order of differential (or difference) equation is reduced to a first order equation and because the dynamics and observation are defined separately with which be shown to be convenient when uncertainty is added to the model.

3

The Inverse Problem

The previous chapter describes how the response of a dynamical system can be obtained with the system description and the input force. This operation can be seen as a forward operation for which the solution can be obtained relatively easy. To find however the input force, this operation needs to be inverted and an inverse operation arises. This inverse operation was named the inverse problem which will be introduced in this chapter. It will be shown that the solution to this inverse problem is more difficult to obtain. The previously described representations of a dynamical system are used to derive an analytical solution to the force identification problem (section 3.1). It will be shown that approaching this problem from different sides will result in a similar formulation of the solution. All can be classified as a same type of problem. The solution to this problem give rise to difficulties described in section 3.2. In section 3.2.5 an overview of possible solutions to the inverse problem is given which will be discussed more extensively in the next chapters.

3.1. Analytical Deconvolution

As the forward operation in the time domain is defined by the convolution, the inverse operation can be defined as a deconvolution process. Deconvolution can be approached in the time domain section (3.1.1), in the state space domain section (3.1.2) or using the Newmark-β time integration method section (3.1.3).

3.1.1. Deconvolution in Time Domain

Whereas the convolution product is a straightforward mathematical operation, the inverse of the convolution product (deconvolution) is not. The most simple way to discretize the convolution integral is as follows:

$$\boldsymbol{u} = \boldsymbol{H}\boldsymbol{f} \tag{3.1}$$

where $\bar{\boldsymbol{H}}$ is the convolution matrix:

$$\bar{\boldsymbol{H}} = \begin{bmatrix} \boldsymbol{H}(0) & 0 & \cdots & \cdots & 0 \\ \boldsymbol{H}(1) & \boldsymbol{H}(0) & 0 & \cdots & \vdots \\ \boldsymbol{H}(2) & \boldsymbol{H}(1) & \ddots & 0 & \vdots \\ \vdots & \vdots & \cdots & \ddots & 0 \\ \boldsymbol{H}(O-1) & \boldsymbol{H}(O-2) & \cdots & \boldsymbol{H}(1) & \boldsymbol{H}(0) \end{bmatrix} \tag{3.2}$$

and O is the size of the IRF. To determine $\boldsymbol{f}$ matrix $\bar{\boldsymbol{H}}$ needs to be inverted:

$$\boldsymbol{f} = \bar{\boldsymbol{H}}^{-1}\boldsymbol{u} \tag{3.3}$$

Using the discretized convolution integral of equation 2.21 a different solution is found. In this derivation the complete time history of the response $\boldsymbol{u}(k)$ and the IRFs $\boldsymbol{H}(k)$ are considered known. We are interested in the force at the current time step $\boldsymbol{f}(k)$ and the forces of the previous time-steps are considered known.

From equation 2.21 it can be seen that only the third term contains the unknown force $\boldsymbol{f}(k)$ and the first and second term consist solely of known terms. These known terms can be grouped together and written as $\tilde{\boldsymbol{u}}(k)$:

$$\boldsymbol{u}(k) = \tilde{\boldsymbol{u}}(k) + \boldsymbol{H}(0)\boldsymbol{f}(k)\frac{dt}{2} \tag{3.4}$$

The force at the current time step will then be:

$$\boldsymbol{f}(k) = \left(\boldsymbol{u}(k) - \tilde{\boldsymbol{u}}(k)\right)\frac{2\boldsymbol{H}(0)^{-1}}{dt} \tag{3.5}$$

with:

$$\tilde{\boldsymbol{u}}(k) = \boldsymbol{H}(k)\boldsymbol{f}(0)\frac{dt}{2} + \sum_{i=1}^{k-1}\boldsymbol{H}(k-i)\boldsymbol{f}(i)dt \tag{3.6}$$

In conclusion, if $\boldsymbol{u}(k)$ is known, $\boldsymbol{f}(k)$ can be determined. The to be inverted matrix in this case consists of the first time steps of the IRFs of all DoFs. Both in equation 3.5 and 3.3, a matrix inverse is needed. These two matrices fundamentally differ, the matrix in equation 3.3 consists of all the time steps of the IRF and the matrix in equation 3.5 consist of only the first time step of the IRF. However, the inversion can lead to rank or more often numerical problems which will be discussed in section 3.2.

3.1.2. Deconvolution in State Space

Deconvolution can also be executed in the state-space domain. The discretized state-space expression of equation 2.37 are used. Assuming a known initial state $\boldsymbol{x}(0)$ the evolution of the state and observation can be determined:

$$\boldsymbol{x}(k+1) = \boldsymbol{A}^{k+1}\boldsymbol{x}(0) + \sum_{i=0}^{k}\boldsymbol{A}^{i}\boldsymbol{B}\boldsymbol{f}(k-i) \tag{3.7}$$

$$\boldsymbol{d}(k) = \boldsymbol{G}\boldsymbol{A}^{k}\boldsymbol{x}(0) + \sum_{i=0}^{k-1}\boldsymbol{G}\boldsymbol{A}^{i}\boldsymbol{B}\boldsymbol{f}(k-i-1) \tag{3.8}$$

This results in a simplification of $\boldsymbol{d}(k)$:

$$\boldsymbol{d}(k) = \boldsymbol{H}(0)\boldsymbol{x}(0) + \sum_{i=1}^{k}\boldsymbol{H}(i)\boldsymbol{f}(k-i) \tag{3.9}$$

with:

$$\boldsymbol{H}(0) = \boldsymbol{G} \tag{3.10}$$

$$\boldsymbol{H}(k) = \boldsymbol{G}\boldsymbol{A}^{k-1}\boldsymbol{B} \tag{3.11}$$

which concludes to:

$$\boldsymbol{d}(k) = \sum_{i=0}^{k}\boldsymbol{H}(k)\boldsymbol{f}(k-1) \tag{3.12}$$

This equation can be conveniently written in matrix form as:

$$\boldsymbol{d}(k) = \boldsymbol{H}\boldsymbol{f}(k) \tag{3.13}$$

where:

$$\bar{\boldsymbol{H}} = \begin{bmatrix} \boldsymbol{H}(0) & 0 & \cdots & \cdots & 0 \\ \boldsymbol{H}(1) & \boldsymbol{H}(0) & 0 & \cdots & \vdots \\ \boldsymbol{H}(2) & \boldsymbol{H}(1) & \ddots & 0 & \vdots \\ \vdots & \vdots & \cdots & \ddots & 0 \\ \boldsymbol{H}(O-1) & \boldsymbol{H}(O-2) & \cdots & \boldsymbol{H}(1) & \boldsymbol{H}(0) \end{bmatrix} \tag{3.14}$$

This formulation describes the forward problem and can be numerically solved for the responses $\boldsymbol{d}(k)$ when both the system description $\boldsymbol{H}$ and the force input $\boldsymbol{f}(k)$ are known. Solving this for the unknown forces $\boldsymbol{f}(k)$ gives the inverse problem which involves the inverse of matrix $\boldsymbol{H}$:

$$\boldsymbol{f}(k) = \boldsymbol{H}^{-1}\boldsymbol{d}(k) \tag{3.15}$$

The solution found here is exactly the same as found in equation 3.3

3.1.3. Newmark-Beta Method

An alternative way of describing the forward, and subsequently the inverse problem is by applying the Newmark-β integration method [48], which was done by Lui [35]. The Newmark-β method is an implicit integration method that obtains from the current acceleration and previous acceleration, velocity and position the current velocity and position:

$$\dot{\boldsymbol{u}}(k+1) = \dot{\boldsymbol{u}}(k) + \frac{\Delta t}{2}\left(\ddot{\boldsymbol{u}}(k) + \ddot{\boldsymbol{u}}(k+1)\right) \tag{3.16}$$

$$\boldsymbol{u}(k+1) = \boldsymbol{u}(k) + \Delta t \dot{\boldsymbol{u}}(k) + \frac{1-2\beta}{2}\Delta t^2 \ddot{\boldsymbol{u}}(k) + \beta \Delta t^2 \ddot{\boldsymbol{u}}(k+1) \tag{3.17}$$

This can be applied to the equations of motion of equation 2.1 to determine the position, velocity and acceleration for all time steps. These can then be used to determine the input forces. After some manipulation this results in an explicit form which determines the current state (acceleration, velocity and position) from the previous state only (the full derivation can be found in [35]):

$$\begin{bmatrix} \boldsymbol{u}(k) \\ \dot{\boldsymbol{u}}(k) \\ \ddot{\boldsymbol{u}}(k) \end{bmatrix} = \sum_{j=0}^{k} \begin{bmatrix} \boldsymbol{A}_d & \boldsymbol{A}_v & \boldsymbol{A}_a \\ \boldsymbol{B}_d & \boldsymbol{B}_v & \boldsymbol{B}_a \\ \boldsymbol{C}_d & \boldsymbol{C}_v & \boldsymbol{C}_a \end{bmatrix}^j \begin{bmatrix} \boldsymbol{A}_0 \\ \boldsymbol{B}_0 \\ \boldsymbol{C}_0 \end{bmatrix} \boldsymbol{S}_p \boldsymbol{f}(k-j) + \begin{bmatrix} \boldsymbol{A}_d & \boldsymbol{A}_v & \boldsymbol{A}_a \\ \boldsymbol{B}_d & \boldsymbol{B}_v & \boldsymbol{B}_a \\ \boldsymbol{C}_d & \boldsymbol{C}_v & \boldsymbol{C}_a \end{bmatrix}^k \begin{bmatrix} \boldsymbol{u}(0) \\ \dot{\boldsymbol{u}}(0) \\ \ddot{\boldsymbol{u}}(0) \end{bmatrix} \tag{3.18}$$

A description of $\boldsymbol{A}_0$, $\boldsymbol{A}_d$, $\boldsymbol{A}_v$, $\boldsymbol{A}_a$, $\boldsymbol{B}_0$, $\boldsymbol{B}_d$, $\boldsymbol{B}_v$, $\boldsymbol{B}_a$, $\boldsymbol{C}_0$, $\boldsymbol{C}_d$, $\boldsymbol{C}_v$ and $\boldsymbol{C}_a$ can be found in appendix A. This somewhat complex equation shows that the state of the system (position, velocity and acceleration) at time k is a linear combination of the mass, damping and stiffness matrix and the input force and the mass, damping and stiffness matrix and the initial conditions. Using this formulation the forward problem can be written as:

$$\boldsymbol{d}(k) = \boldsymbol{H}_L \boldsymbol{f}(k) \tag{3.19}$$

where:

$$\boldsymbol{H}_L = \begin{bmatrix} \boldsymbol{H}(0)\boldsymbol{S}_p & 0 & \cdots & \cdots & 0 \\ \boldsymbol{H}(1)\boldsymbol{S}_p & \boldsymbol{H}(0)\boldsymbol{S}_p & 0 & \cdots & \vdots \\ \boldsymbol{H}(2)\boldsymbol{S}_p & \boldsymbol{H}(1)\boldsymbol{S}_p & \ddots & 0 & \vdots \\ \vdots & \vdots & \cdots & \ddots & 0 \\ \boldsymbol{H}(O-1)\boldsymbol{S}_p & \boldsymbol{H}(O-2)\boldsymbol{S}_p & \cdots & \boldsymbol{H}(1)\boldsymbol{S}_p & \boldsymbol{H}(0)\boldsymbol{S}_p \end{bmatrix} \tag{3.20}$$

and:

$$\boldsymbol{H}(k) = \begin{bmatrix} \boldsymbol{S}_a & \boldsymbol{S}_v & \boldsymbol{S}_d \end{bmatrix} \begin{bmatrix} \boldsymbol{A}_d & \boldsymbol{A}_v & \boldsymbol{A}_a \\ \boldsymbol{B}_d & \boldsymbol{B}_v & \boldsymbol{B}_a \\ \boldsymbol{C}_d & \boldsymbol{C}_v & \boldsymbol{C}_a \end{bmatrix}^k \begin{bmatrix} \boldsymbol{A}_0 \\ \boldsymbol{B}_0 \\ \boldsymbol{C}_0 \end{bmatrix} \tag{3.21}$$

The inverse problem for force identification will then be:

$$\boldsymbol{f}(k) = \boldsymbol{H}_L^{-1} \boldsymbol{d}(k) \tag{3.22}$$

Again, a same type of relation between the measurement and input force is found as in equations 3.3, 3.5 and 3.15. Solving these equations leads difficulties explained in the next section.

3.2. The Inverse Problem

In the previous section three different methods of deconvolution have been described which in the end all produces a same type of problem; namely the inverse problem. In section 3.1.1 the to be inverted matrix consists of the IRFs of all DoFs whereas in section 3.1.2 and 3.1.3 the results shows a matrix which consists of all the time steps of the IRFs. Although these two matrices fundamentally differ in the way they are build, a solution to the inverse of these matrices can be approached in the same manner. The notation of the three different approaches is slightly different. For clarity, one notation will be used in this section. The following equation describes the forward problem:

$$\boldsymbol{d} = \boldsymbol{H}\boldsymbol{f} \tag{3.23}$$

resulting in the notation for the inverse problem:

$$\boldsymbol{f} = \boldsymbol{H}^{-1}\boldsymbol{d} \tag{3.24}$$

When this inverse problem can be solved as it is, the problem is said to be well-posed. The solution in that case is unique and stable. However, in practice this is rarely the case as will be shown in the following sections and a solution is either not unique, not stable or does not exist at all. The problem is then said to be ill-posed.

3.2.1. The Singular Value Decomposition

A useful tool to examine to posedness of a problem is the Singular Value Decomposition (SVD) which can be seen as a general form of the eigenvalue decomposition. The SVD decomposes a matrix into two matrices of left and right singular vectors and a diagonal matrix of singular values:

$$\boldsymbol{H} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \tag{3.25}$$

The left singular values $\boldsymbol{U}$ consist of the eigenvectors of $\boldsymbol{H}\boldsymbol{H}^T$ and can be seen as mode shapes of the response. The right singular values $\boldsymbol{V}$ consist of the eigenvectors of $\boldsymbol{H}^T\boldsymbol{H}$ and can be seen as mode shapes of the input. Matrices $\boldsymbol{H}\boldsymbol{H}^T$ and $\boldsymbol{H}^T\boldsymbol{H}$ have the same eigenvalues σ_i^2 where:

$$\boldsymbol{\Sigma} = \sum_1^r \sigma_i \tag{3.26}$$

with $r = rank(\boldsymbol{H})$. The posedness of a matrix can be scored by its condition number which is defined as:

$$cond(\boldsymbol{H}) = \frac{\sigma_1}{\sigma_n} \tag{3.27}$$

where n is the dimension of $\boldsymbol{u}$. Perfect orthogonality of all columns of $\boldsymbol{H}$ (for example an identity matrix) will have a condition number of one. As orthogonality decreases, the condition number will increase. When two or more columns point in the same direction (i.e. linear dependence) the condition number will be infinity. As a rule of thumb, matrices with a condition number above 1×10^3 are said to be ill-conditioned.

3.2.2. Rank Deficiency

Whenever two or more columns of matrix $\boldsymbol{H}$ are linear dependent it is said to be rank deficient. Inverting this matrix is impossible. When looking at:

$$\boldsymbol{d} = \boldsymbol{H}\boldsymbol{f} \tag{3.28}$$

there exists no $\boldsymbol{f}$ that results in a unique solution for $\boldsymbol{d}$. This translates to one or more singular values being zero. An example is illustrated in figure 3.1. It shows the singular values of a $30x30$ identity matrix with the last four columns made equal making them linear dependent. It is clearly visible that the last four singular values are zero which indicates rank deficiency.

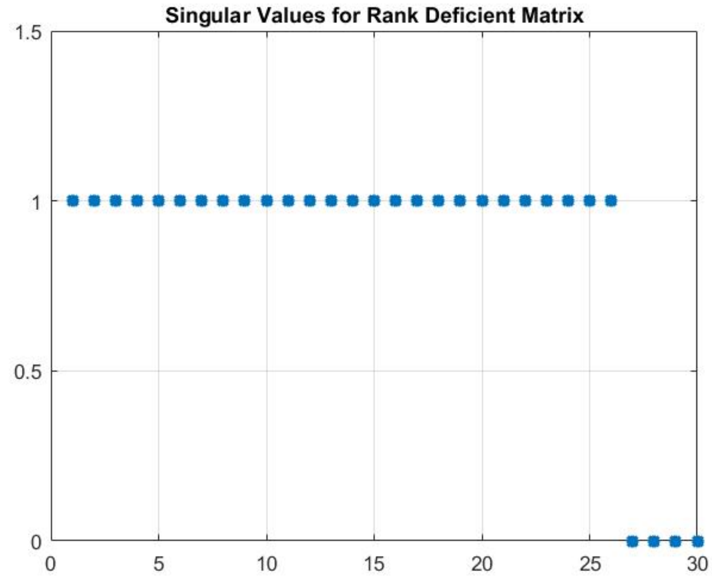


Figure 3.1: Singular Values of a rank deficient matrix

In practice fully linear dependence is rarely the case. However, a similar but different case is shown in the next section.

3.2.3. Numerical Problems

Next to rank deficiency numerical issues can occur when trying to find a solution to the inverse problem. This is similar to the previously described problem but fundamentally different. As the size of $\boldsymbol{H}$ increases its columns will become more alike. Orthogonality will decrease as matrix size will increase. They will most likely never become exactly linear dependent but their dependence will increase. An example of this is shown in figure 3.2 where the condition number of a randon matrix with size $n \times n$ is shown. The average is taken over a large number of random matrices to show the influence of matrix size.

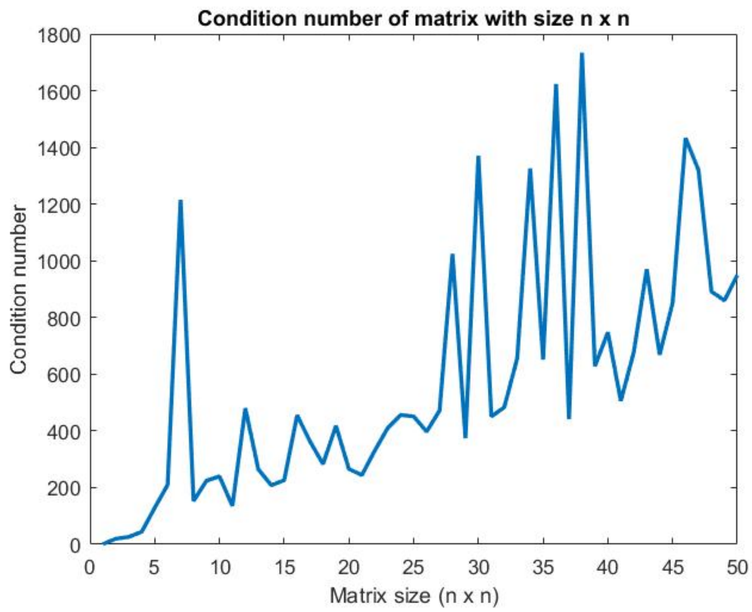


Figure 3.2: Condition number of matrix with increasing size

As opposed to a rank deficient matrix where the singular values drop to zero, the singular values will in this case gradually decay to zero as shown in figure 3.3 for a 30×30 matrix.

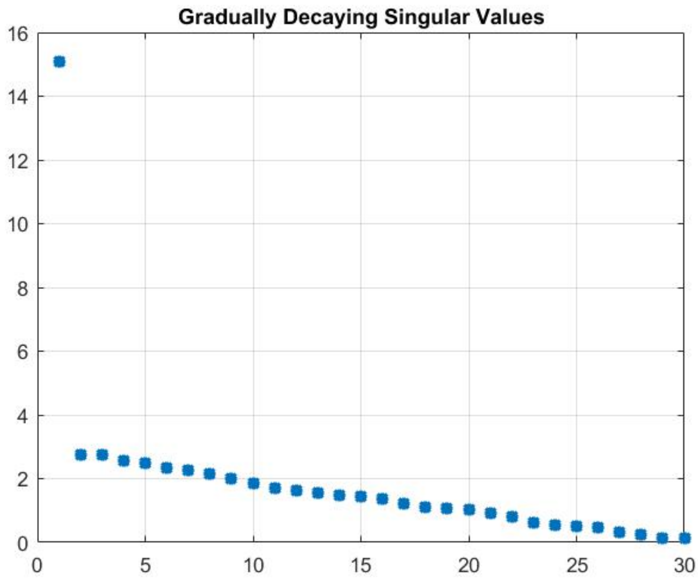


Figure 3.3: Gradually decaying singular values for a random matrix

The problem with the gradually decaying singular values is explained in the next section.

3.2.4. The Least Squares Solution

A solution to the inverse problem can be sought in the least squares (LS) solution where an optimal solution is found by minimizing the distance between the output $\boldsymbol{u}$ and $\boldsymbol{Hf}$. The to be minimized cost function will be:

$$\boldsymbol{J} = ||\boldsymbol{Hf} - \boldsymbol{u}||_2 \tag{3.29}$$

A solution of this minimization problem is:

$$\boldsymbol{f}_{ls} = (\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T\boldsymbol{u} \tag{3.30}$$

or written in terms of the SVD:

$$\boldsymbol{f}_{svd} = (\boldsymbol{V}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^T)\boldsymbol{u} \tag{3.31}$$

where

$$\boldsymbol{\Sigma}^{-1} = diag(\frac{1}{\sigma_i}) \tag{3.32}$$

This can also be written as a summation over all dimensions:

$$\boldsymbol{f}_{svd} = \sum_{i=1}^{r} \frac{\boldsymbol{u}_i\boldsymbol{u}}{\sigma_i}\boldsymbol{v}_i \tag{3.33}$$

This shows that for singular values approaching zero small perturbations in $\boldsymbol{u}$ have a large effect on $\boldsymbol{f}_{svd}$ because it is divided by the small singular value σ_i. This will be illustrated with an example. A system matrix $\boldsymbol{H}$ of size $n \times n$ will be chosen randomly and will be multiplied with a vector $\boldsymbol{f} = \begin{bmatrix}1_1 & ... & 1_n\end{bmatrix}^T$. The output will be perturbed by a small error to create a noisy measurement. Figures 3.4 and 3.5 shows the measurement and a small perturbation above and the least-squares solution of the unperturbed and perturbed output below.

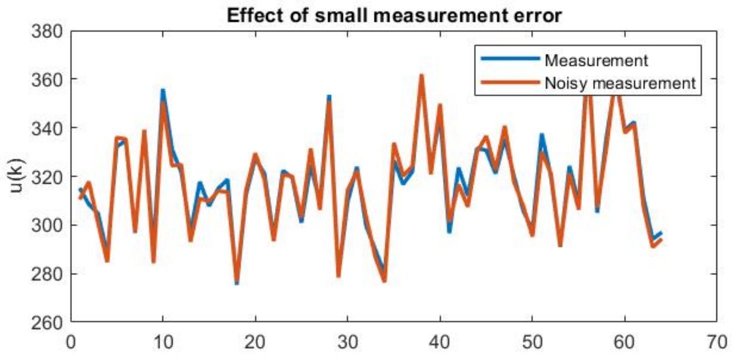


Figure 3.4: Measurement and noisy measurement

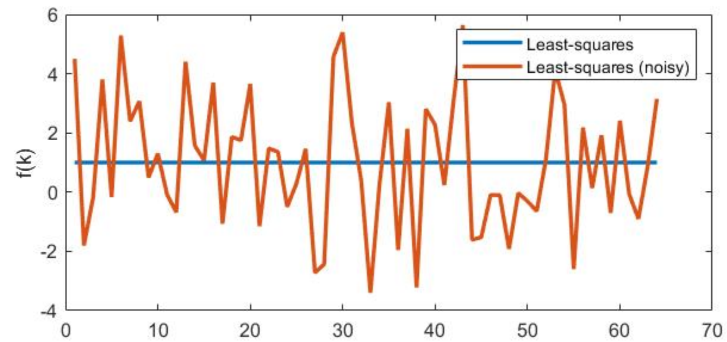


Figure 3.5: Least squares solution using the measurement and noisy measurement

Clearly, a small perturbation in the measurement (figure 3.4) results in a large error in the identified force (figure 3.5) so the solution in unstable. This analytical solution gives an unwanted result and a solution must be sought from a different approach.

3.2.5. Solutions to the Inverse Problem

In the past, numerous approaches have been attempted to obtain a solution to this force identification problem that can be seen as an inverse problem. All methods can be grouped into three categories: Deterministic methods, stochastic methods and artificial intelligence methods. These three categories fundamentally differ in the way the system dynamics is modeled. An overview can be found in figure 3.6

The first category uses a deterministic model of the system. This is most often done with the use of IRFs however other models can be used as well. Deterministic methods include regularization methods where the above described ill-posed problem is replaced by a similar but well-posed problem. A solution can also be found recursively over time or iteratively.

Stochastic methods include uncertainty in the model. When done correctly this can be advantageous as no model perfectly describes the true dynamics and no sensor measures without an error. Stochastic models are mostly described in state space as the equations for the system dynamic and the observation are uncoupled and a definition for uncertainty can be added to both separately. Most stochastic methods are based on or similar to the Kalman filter. This research discusses three different approaches that uses the Kalman filter as a basis: The augmented Kalman filter (AKF), the dual Kalman filter (DKF) and the joint input-state estimator (JISE).

A third category uses no predefined model of the system. Instead, it feeds a large amount of data to an artificial intelligence which uses this data to define an input-output relation [20, 29, 50, 55, 62, 70]. As the dynamics of the system is assumed known in this research there is no need for implementing a complex artificial intelligence algorithm to obtain this system description. Implementation of these methods is much more labour intensive compared the the deterministic and stochastic methods. Therefore this category of methods will not be discussed further in this research.

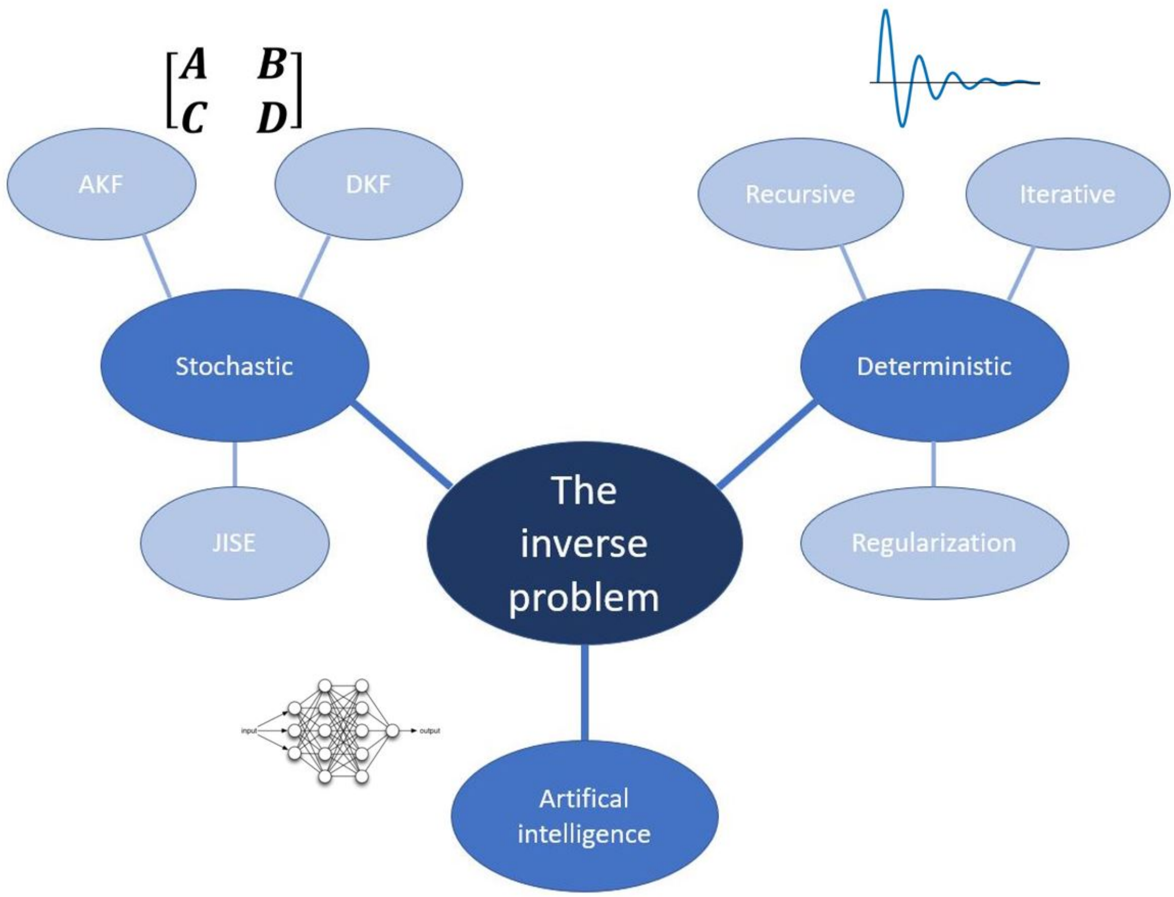


Figure 3.6: Categorization of solutions to the inverse problem

4

Deterministic Force Identification

This chapter describes a deterministic approach to force identification implying that the system, its input and its output are defined deterministically. Result of this is that uncertainties are not taken into account.

Jacquelin and Hamelin developed a method for force identification that uses three strain gauges [23]. This method is successfully applied to recover the axial load in a steel bar. However this method limits to recovering axial loads on simple structures of which the area subject to stress can be determined easily. Law and Chan introduced a method to identify the forces exerted by cars driving over a bridge [33]. The method is based on modal superposition and the bridge is modeled as a Bernoulli-beam. Again, this method is effective for its application but is limited to beam structures. The Sum of Weighted Accelerations Technique (SWAT) was developed by Carne et al. [13]. It uses modal reduction to isolate the accelerations on a body due to its rigid body modes. These accelerations can be used when the mass is known to determine the forces acting on the center of mass on the body. It has to be noted that this technique is only applicable for unconstrained bodies as the contribution of the deforming modes is discarded. Therefore this method is useful in for example aviation and space engineering, however is it not suitable for TPA. Steltzner and Kammer developed the Inverse Structural Filter (ISF) which uses a non-causal moving average filter as a representation of the inverse of the system [57]. This method has been applied successfully but it was found that difficulties arise when identifying moments. Xu proposed a method [69] for force identification that makes use of the moving least-squares algorithm [32]. This algorithm is used to fit the input force to a shape function. By means of the concept of virtual displacement the equations of motion are simplified to improve computational accuracy and time. This results in a system of equations for which the solution is found by an iterative scheme. A disadvantage of this method is that it is better at identifying harmonic forces compared to arbitrary forces. This may imply that it is not suitable for capturing non-stationary effect. Also, it is shown that the error in the identified force increases with an increase in noise and that this increase is faster.

4.1. Regularization

Clearly the in section 3.2.4 described least-squares solution does not yield a desired result, as a small perturbation in the measurement results in a large perturbation in the solution. A method to overcome this amplification of noise is to regularize matrix $\boldsymbol{H}$. Different regularization methods are discussed in this section which all replace the ill-posed problem by a well-posed problem which is similar to the original. Because matrix $\boldsymbol{H}$ is altered, the solution is an approximation of the actual solution. The alteration of matrix $\boldsymbol{H}$ is done by discarding or decreasing the influence of the numerically troublesome small singular values.

Regularization is done by adding a regularization term. This added term comes in the form of a norm which is explained in Appendix B. A disadvantage of regularization methods is that they can only be applied when input and output are collocated.

4.1.1. Truncated Singular Value Decomposition

To prevent the effect small singular values have on the noise on a measurement signal they can simply be discarded which is exactly what the Truncated Singular Value Decomposition does [60]. When looking at the

SVD of matrix $\boldsymbol{H}$:

$$\boldsymbol{H} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \tag{4.1}$$

the LS solution in terms of the SVD is:

$$\boldsymbol{f}_{svd} = (\boldsymbol{V}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^T)\boldsymbol{d} \tag{4.2}$$

where

$$\boldsymbol{\Sigma}^{-1} = diag(\frac{1}{\sigma_i}) \tag{4.3}$$

The TSVD truncates the singular values of matrix $\boldsymbol{H}$. Matrix $\boldsymbol{\Sigma}^{-1}$ will remain as it is up to index p. From thereon the diagonals $\frac{1}{\sigma_i}$ will be set to zero discarding the influence of the singular values from that point:

$$\begin{cases} \boldsymbol{\Sigma}^{-1}_{i,i} = \frac{1}{\sigma_i}, & i < p \\ \boldsymbol{\Sigma}^{-1}_{i,i} = 0, & i > p \end{cases} \tag{4.4}$$

For a rank deficient matrix $\boldsymbol{H}$, choosing p is straightforward as a clear gap in thee singular values is visible. Finding the right p for ill-conditioned matrix $\boldsymbol{H}$ is more difficult as the singular values gradually decay.

4.1.2. Tikhonov Regularization

Whenever the to be solved inverse problem is ill-posed and the LS solution does not suffice, coefficient matrix $\boldsymbol{H}$ will have to be altered in such a way that it yields a solution that is a good approximation of the true solution. A method where a regularization term in the form of the 2-norm of the solution is added to the LS solution is called Tikhonov regularization [61]. It was first introduced by Tikhonov in 1963 and has since been applied in many fields among which force identification [5, 19, 35]. With the addition of the regularization term this results in the following minimization problem:

$$\boldsymbol{f} = \underset{\boldsymbol{f}}{\operatorname{argmin}}\Big(||\boldsymbol{H}\boldsymbol{f} - \boldsymbol{u}||_2^2 + \lambda||\boldsymbol{L}\boldsymbol{f}||_2^2\Big) \tag{4.5}$$

where λ is a positive constant scalar, also referred to as the regularization parameter and $||\cdot||_2$ defines the 2-norm (Euclidean norm). The regularization parameter defines the amount of regularization and can be determined with the L-curve method [21]. For standard Tikhonov Regularization $\boldsymbol{L}$ is the identity matrix. Often $\boldsymbol{L}$ is chosen as the first or second derivative operator (finite difference matrix) [18]. The solution to this minimization problem is:

$$\boldsymbol{f} = (\boldsymbol{H}^T\boldsymbol{H} + \lambda\boldsymbol{L}^T\boldsymbol{L})^{-1}\boldsymbol{H}^T\boldsymbol{d} \tag{4.6}$$

The regularization term can be seen as a smoothing term. Minimizing the 2-norm (or Euclidean distance) of the solution prevents extreme values in the solution. The regularization parameter weights between the data and smoothness of the solution. This solution in terms of the SVD is:

$$\boldsymbol{f}_{svd} = (\boldsymbol{V}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^T)\boldsymbol{d} \tag{4.7}$$

where

$$\boldsymbol{\Sigma}^{-1}_{i,i} = \frac{\sigma_i}{\sigma_i^2 + \lambda^2} \tag{4.8}$$

This shows that for large singular values the influence of the regularization parameter λ is small. As the singular values decrease the influence of λ increases. This shows that Tikhonov regularization gradually decreases the influence of the singular values as they decrease in value. This in contrast to the TSVD where the influence of the singular values is completely discarded after a certain point. A drawback of Tikhonov regularization as well as for the TSVD method is that sensor collocation is required [26]. An illustrative example of Tikhonov regularization in two dimensions is given in Appendix C.

Besides the least squares solution of this minimization problem, it can also be solved with for example a gradient based minimization method [19].

4.1.3. Total Variation Regularization

Whereas Tikhonov Regularization minimizes the Euclidean norm of the solution, a different approach is minimizing the 1-norm of the regularization term. This method is called Total Variation Regularization (TVR).

$$\boldsymbol{f} = \underset{\boldsymbol{f}}{\operatorname{argmin}}\Big(||\boldsymbol{Hf}-\boldsymbol{u}||_2^2 + \lambda||\boldsymbol{Lf}||_1\Big) \tag{4.9}$$

An advantage of total variation regularization is that the solution can take a non-smooth form. A disadvantage is that the 1-norm may introduce non-smoothness in the objective function and a gradient based optimization method is not applicable. However, the solution is convex so a non-gradient-based convex optimization method will suffice. Efficient methods for solving this problem have been proposed by Beck [10] and Goldstein [17].

4.1.4. Generalized Tikhonov Regularization

Generalized Tikhonov regularization (GTR) is also called $L_p - L_q$ regularization and is a regularization method for which the norms are not predefined [5]. This results in the following minimization problem:

$$\boldsymbol{f} = \underset{\boldsymbol{f}}{\operatorname{argmin}}\Big(\frac{1}{p}||\boldsymbol{Hf}-\boldsymbol{u}||_p^p + \frac{\lambda}{q}||\boldsymbol{Lf}||_q^q\Big) \tag{4.10}$$

where p and q are parameters to be optimized for force reconstruction. It can be seen that for $p = q = 2$ the standard Tikhonov regularization problem is found. The solution to this problem can be found by the Generalized Iteratively Reweighted Least-Squares (GIRLS) algorithm [5].

4.2. Recursive Methods

In the previous section the focus lies on regularizing the analytical solution to the force identification problem. A disadvantage of such methods is that the square coefficient matrix $\boldsymbol{H}$ increases with the number of time steps. Computation time therefore increases quadratically with respect to the number of time steps. A way around this is by means of recurrence. A recurrence algorithm uses the results of the previous time step as an input and therefore its computation time scales linearly with respect to the number of time steps. This section introduces two recursive methods, both based in Dynamic Programming.

4.2.1. Classical Dynamic Programming

The Dynamic Programming (DP) algorithm was introduced by Bellman [11] and first applied for force identification by Trujillo dating back to 1978 [63]. Since then is has been applied numerous times, for example in [1, 22, 36, 72]. The working principle behind DP is the fact that a sub element of an optimal solution is an optimal solution of the corresponding sub problem. When this is applied recursively starting at the end, an optimal solution can be found if the problem has optimal substructure. The optimum is defined as the solution that minimizes the least-squares error between the measured data and the response of the system on the reconstructed force. Looking at the system in state space (equation 2.37), the relation between the measurement and the states is:

$$\boldsymbol{d}(k) = \boldsymbol{Gx}(k), \quad \boldsymbol{x}(0) = \boldsymbol{c} \tag{4.11}$$

where the observation matrix $\boldsymbol{J} = \boldsymbol{0}$. For the sake of notation discrete time in this section is noted by a subscript as apposed to the bracket notation used throughout the rest of this research. The systems reaction to the reconstructed force is aimed to match the measurement. The weighted least squares error on this observation depending on the force input and the initial condition of the state written out in vector form and summed over all time steps can then be defined as:

$$E(\boldsymbol{c}, \boldsymbol{f}(k)) = \sum_{k=1}^{N}\Big(\big(\boldsymbol{Gx}(k) - \boldsymbol{d}(k)\big)^T \boldsymbol{W}\big(\boldsymbol{Gx}(k) - \boldsymbol{d}(k)\big)\Big) \tag{4.12}$$

where $\boldsymbol{W}$ is a weight matrix to give preference to certain elements of the state. Minimizing this error results in an optimal least squares solution for the reconstructed force. However, the regular least squares solution gives rise to the problem of ill-posedness. To account for this, a Tikhonov regularization term is added:

$$E(\boldsymbol{c}, \boldsymbol{f}(k)) = \sum_{k=1}^{N}\big((\boldsymbol{Gx}(k) - \boldsymbol{d}(k)^T \boldsymbol{W}(\boldsymbol{Gx}(k) - \boldsymbol{d}(k) + \lambda\boldsymbol{f}(k)^T\boldsymbol{Df}(k)\big) \tag{4.13}$$

where $\boldsymbol{D}$ is a weight matrix. Minimizing this least square error gives an optimal estimate of the input force. The solution to this problem can be written as a minimization problem for each time step:

$$g_k(\boldsymbol{c}) = \min_{\boldsymbol{f}_k} E(\boldsymbol{c}, \boldsymbol{f}_k) \tag{4.14}$$

where $g_k(\boldsymbol{c})$ is the objective function to be minimized. This minimization problem can be solved with the use of Bellman's Principle of Optimality [11]:

"An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision."

This states that when an optimum has been found for a certain time step, this optimum is part of the complete solution and can be used as initial condition ($\boldsymbol{c} = \boldsymbol{x}_k$) for the previous time step. This implies a backwards recursive relation in time (filling equation 4.13 into 4.14):

$$g_{k-1}(\boldsymbol{c}) = \min_{\boldsymbol{f}_{k-1}} \Big((\boldsymbol{Gc} - \boldsymbol{d})_{k-1}^T \boldsymbol{W} (\boldsymbol{Gc} - \boldsymbol{d})_{k-1} + \lambda \boldsymbol{f}_{k-1}^T \boldsymbol{D} \boldsymbol{f}_{k-1} + g_k(\boldsymbol{Ac} + \boldsymbol{B}\boldsymbol{f}_{k-1}) \Big) \tag{4.15}$$

with the recursion starting at the end at $k = N$. The first and second term minimizes for the current time step the least square error with an added Tikhonov regularization term. The third term minimizes the remaining cost. Choosing $\boldsymbol{f}_N = \boldsymbol{0}$ the minimization becomes:

$$g_N = \boldsymbol{c}^T \boldsymbol{G}^T \boldsymbol{W} \boldsymbol{G} \boldsymbol{c} - 2\boldsymbol{c}^T \boldsymbol{G}^T \boldsymbol{W} \boldsymbol{d}_N + \boldsymbol{d}_N^T \boldsymbol{W} \boldsymbol{d}_N \tag{4.16}$$

which shows that g_N is quadratic in $\boldsymbol{c}$ and in the same way is can be proven inductively that for all n, $\boldsymbol{f}_n$ is quadratic in $\boldsymbol{c}$. The quadratic expression of the minimization problem can then be written as:

$$g_k = \boldsymbol{c}^T \boldsymbol{R}_k \boldsymbol{c} + \boldsymbol{c}^T \boldsymbol{S}_k + q_k \tag{4.17}$$

where $\boldsymbol{R}_k$ is a symmetric matrix containing the first and quadratic term in $\boldsymbol{c}$, $\boldsymbol{S}_n$ a vector containing the second and linear term in $\boldsymbol{c}$ and q_n a scalar containing the constant term resulting from the found quadratic equation that minimizes the least square error. The initial values (for $k = N$) can be obtained with equation 4.16 and are:

$$\boldsymbol{R}_N = \boldsymbol{G}^T \boldsymbol{W} \boldsymbol{G} \tag{4.18}$$

$$\boldsymbol{S}_N = -2\boldsymbol{G}^T \boldsymbol{W} \boldsymbol{d}_N \tag{4.19}$$

$$q_N = \boldsymbol{d}_N^T \boldsymbol{W} \boldsymbol{d}_N \tag{4.20}$$

The recursive relations can be found by filling out equation 4.17 into equation 4.15 which gives:

$$\begin{aligned} \boldsymbol{c}^T \boldsymbol{R}_k \boldsymbol{c} + \boldsymbol{c}^T \boldsymbol{S}_k + q_k = \min_{\boldsymbol{f}} \Big(& (\boldsymbol{Gc} - \boldsymbol{d})_{k-1}^T \boldsymbol{W} (\boldsymbol{Gc} - \boldsymbol{d})_{k-1} + \lambda \boldsymbol{f}_{k-1}^T \boldsymbol{D} \boldsymbol{f}_{k-1} + \\ & (\boldsymbol{Ac} + \boldsymbol{B}\boldsymbol{f}_{k-1})^T \boldsymbol{R}_k (\boldsymbol{Ac} + \boldsymbol{B}\boldsymbol{f}_{k-1}) + (\boldsymbol{Ac} + \boldsymbol{B}\boldsymbol{f}_{k-1})^T \boldsymbol{S}_k + q_k \Big) \end{aligned} \tag{4.21}$$

The minimization problem is quadratic in $\boldsymbol{f}_k$ so the optimum is found by taking the derivative with respect to $\boldsymbol{f}_{k-1}$:

$$\frac{\partial g_{k-1}}{\partial \boldsymbol{f}_{k-1}} = 2\boldsymbol{H}\boldsymbol{f}_{k-1} + 2\boldsymbol{B}^T \boldsymbol{R}_k (\boldsymbol{Ac} + \boldsymbol{B}\boldsymbol{f}_{k-1}) + \boldsymbol{B}^T \boldsymbol{S}_k = 0 \tag{4.22}$$

$$\boldsymbol{f}_{k-1} = -\boldsymbol{D}_k \boldsymbol{B}^T \boldsymbol{S}_k - \boldsymbol{D}_k \boldsymbol{H}_k \boldsymbol{Ac} \tag{4.23}$$

where

$$\boldsymbol{D}_k = (2\boldsymbol{H}_k + 2\boldsymbol{B}^T \boldsymbol{R}_k \boldsymbol{B})^{-1} \tag{4.24}$$

$$\boldsymbol{H}_k = 2\boldsymbol{B}^T \boldsymbol{R}_k \tag{4.25}$$

The recurrence steps for $\boldsymbol{R}_n$ and $\boldsymbol{S}_n$ are then:

$$\boldsymbol{R}_{k-1} = \boldsymbol{C}^T\boldsymbol{W}\boldsymbol{C} + \boldsymbol{A}^T(\boldsymbol{R}_{\boldsymbol{k}} - \boldsymbol{H}_k^T\boldsymbol{D}_k\boldsymbol{H}_k/2)\boldsymbol{A} \tag{4.26}$$

$$\boldsymbol{S}_{k-1} = -2\boldsymbol{G}^T\boldsymbol{W}\boldsymbol{d}_{k-1} + \boldsymbol{A}^T(\boldsymbol{I} - \boldsymbol{H}_k^T\boldsymbol{D}_k\boldsymbol{B}^T)\boldsymbol{S}_k \tag{4.27}$$

The recurrence steps of equation 4.26 and 4.27 can be solved starting at the end at $k = N$. These can be used in equations 4.24 and 4.25 to solve equation 4.23 resulting in the reconstructed force. It is shown that the DP algorithm recovers forces successfully when measurement and input are collocated but whenever the distance increases, the quality of the identification decreases [22]. Also, noisy measurements are not filtered by the DP method itself, there will be a relative sensitivity to weighting parameter $\boldsymbol{H}$. When the data is extremely noisy, it will have to be filtered before using it in the DP algorithm. This can be done by for example a Butterworth filter. Another disadvantage of this method is that computation time increases significantly with an increase of the order of the model. To limit this effect, modal reduction can be applied.

4.2.2. Extended Dynamic Programming

A method to include acceleration measurements was developed by Lourens [37] which is an extension to the classical DP algorithm. It uses the following extended observation equation:

$$\boldsymbol{D}(k) = \boldsymbol{G}\boldsymbol{x}(k) + \boldsymbol{J}\boldsymbol{f}(k) \tag{4.28}$$

Using the same approach as the derivation for the recursive equation of the classical DP algorithm the recursion equations are found. A detailed description of this derivation can be found in [37]. The initial conditions for $\boldsymbol{R}_n$, $\boldsymbol{S}_n$ and q_n are:

$$\boldsymbol{R}_N = \boldsymbol{G}^T\boldsymbol{W}\boldsymbol{G} \tag{4.29}$$

$$\boldsymbol{S}_N = -2\boldsymbol{G}^T\boldsymbol{W}\boldsymbol{d}_N \tag{4.30}$$

$$q_N = \boldsymbol{d}_N^T\boldsymbol{W}\boldsymbol{d}_N \tag{4.31}$$

The identified force is:

$$\boldsymbol{f}_{k-1} = -\boldsymbol{L}_k\boldsymbol{T}_k\boldsymbol{c} + \boldsymbol{L}_k\boldsymbol{F}_k - \boldsymbol{L}_k\boldsymbol{B}^T\boldsymbol{S}_k \tag{4.32}$$

where:

$$\boldsymbol{L}_k = (2\boldsymbol{J}^T\boldsymbol{W}\boldsymbol{J} + 2\boldsymbol{H} + 2\boldsymbol{B}^T\boldsymbol{R}_k\boldsymbol{B})^{-1} \tag{4.33}$$

$$\boldsymbol{T}_k = 2\boldsymbol{J}^T\boldsymbol{W}\boldsymbol{G} + \boldsymbol{B}^T(\boldsymbol{R}_k^T + \boldsymbol{R}_k)\boldsymbol{A} \tag{4.34}$$

$$\boldsymbol{F}_k = 2\boldsymbol{J}^T\boldsymbol{W}\boldsymbol{d}_{k-1} \tag{4.35}$$

The recursion equations for $\boldsymbol{R}_n$ and $\boldsymbol{S}_n$ are:

$$\boldsymbol{R}_{k-1} = \boldsymbol{G}^T\boldsymbol{W}(\boldsymbol{G} - \boldsymbol{J}\boldsymbol{L}_k\boldsymbol{T}_k) + \boldsymbol{A}^T(\boldsymbol{R}_k\boldsymbol{A} - (\boldsymbol{R}_k + \boldsymbol{R}_k^T)\boldsymbol{B}\boldsymbol{L}_k\boldsymbol{T}_k/2) \tag{4.36}$$

$$\begin{aligned}\boldsymbol{S}_{k-1} = \boldsymbol{G}^T\boldsymbol{W}\boldsymbol{J}\boldsymbol{L}_k(\boldsymbol{F}_k - \boldsymbol{B}^T\boldsymbol{S}_k) - 2\boldsymbol{G}^T\boldsymbol{W}\boldsymbol{d}_{k-1} + \boldsymbol{T}_k^T\boldsymbol{L}^T(\boldsymbol{F}_k/2 - \boldsymbol{B}^T\boldsymbol{S}_k/2) + \\ \boldsymbol{A}^T(\boldsymbol{R}_k + \boldsymbol{R}_k^T)\boldsymbol{B}\boldsymbol{L}_k(\boldsymbol{F}_k/2 - \boldsymbol{B}^T\boldsymbol{S}_k/2) + \boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{S}_{\boldsymbol{k}}\end{aligned} \tag{4.37}$$

where $\boldsymbol{R}_n$ is no longer symmetric. The extended DP algorithm is solved in the same manner as the classical DP algorithm starting at the end at $k = N$.

4.3. Iterative Methods

Two deterministic iterative methods for force identification are described in this section. An iterative method is necessary whenever an algorithm is not converged over one iteration and an error to be minimized can be described.

4.3.1. Least-Mean-Square Adaptive Algorithm

The Least Mean Square (LMS) algorithm was first introduced by Widrow [68] in 1960 and was later applied for force identification by Sturm [59]. Originally the algorithm was designed to identify an unknown system (figure4.1a). For clarity table 4.1 shows all indexing symbols used in this section.

Symbol	Meaning
k	Discrete time
i	Dummy time variable
N	Size of the input vector
O	Number of time steps of the IRFs
S	Number of input positions
M	Number of response positions

Table 4.1: Indexing symbols used in this section

The unknown system, to be defined by its IRF and an adaptive filter are both given the same input $\boldsymbol{x}(k)$. The response of the filter can be determined by convolution:

$$y(k) = \sum_{i=0}^{N-1} h(k) f(k-i) = \boldsymbol{h}(k)^T \boldsymbol{f}(k) \tag{4.38}$$

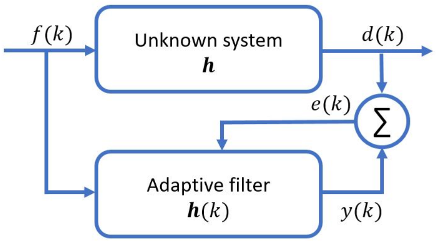
where $\boldsymbol{f}(k)$ is the input vector defined as:

$$\boldsymbol{f}(k) = \begin{bmatrix} f(k) & f(k-1) & \dots & f(k-N+1) \end{bmatrix}^T \tag{4.39}$$

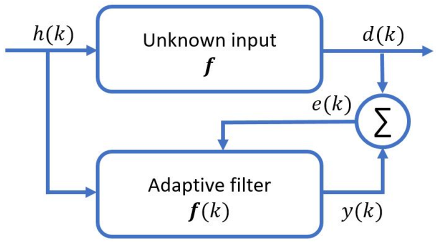
The response of the adaptive filter ($y(k)$) is compared to the measurement ($d(k)$) and the adaptive filter is adjusted by means of the error converging the adaptive filter towards the unknown system. The error is defined as:

$$e(k) = d(k) - y(k) \tag{4.40}$$

As the response of the filter is determined by means of the convolution product, which is commutative, the input and system description can be interchanged. This makes the IRF the input and the adaptive filter correspond to the input force (figure 4.1b).



(a) Schematics of the original LMS algorithm



(b) Schematics of the adjusted LMS algorithm

The definition of the error does not change with this adjustment but has to be minimized with respect to the force vector. The mean-square of the error is:

$$\zeta(k) = E\left[e^2(k)\right] = E\left[d^2(k)\right] - 2\boldsymbol{h}^T E\left[d(k)\boldsymbol{f}(k)\right] + \boldsymbol{h}^T E\left[\boldsymbol{f}(k)\boldsymbol{f}^T(k)\right]\boldsymbol{h} \tag{4.41}$$

Clearly the mean-square error is quadratic in $\boldsymbol{f}(k)$ for which the minimum can be found by steepest descent optimization:

$$\boldsymbol{f}(k+1) = \boldsymbol{f}(k) - \mu\nabla\zeta(k) = \boldsymbol{f}(k) + 2\mu E\left[e(k)\boldsymbol{h}\right] \tag{4.42}$$

where $\nabla\zeta(k)$ denoted the gradient of $\zeta(k)$ and μ is a parameter to control the step size. In the original algorithm, the input vector $\boldsymbol{f}(k)$ changes over time and the to be identified object $\boldsymbol{h}(k)$ consists of the same time steps over the whole identification process. When, however the to be identified object is the input force which changes over time convergence is no longer guaranteed as every time step in the filter can only be updated O times where O is the number of time steps of the IRFs. For convergence an iteration step is introduced. After O time steps, the next iteration is started repeating the process until a convergence criteria is met. By the definition of $\boldsymbol{x}(k)$ (equation 4.39) the algorithm has to start at $k = N$ resulting in the update step for the force vector:

$$\boldsymbol{f}(k+1) = \boldsymbol{f}(k) + 2\mu E\left[e(k)\boldsymbol{h}\right], \quad O \geq k \geq N, \quad \forall N \geq 2O \tag{4.43}$$

The algorithm can be extended to a MDoF system. The response will simply be a summation over all corresponding convolutions:

$$y_m(k) = \sum_{s=1}^{S}\sum_{i=0}^{k-1} \boldsymbol{H}_s(k-i)\boldsymbol{f}_s(k) \tag{4.44}$$

where s is the current input force, S the total number of input forces and m is the current response position. This results in an error of:

$$\boldsymbol{e}_m(k) = \boldsymbol{d}(k) - \boldsymbol{y}(k) \tag{4.45}$$

where $\boldsymbol{d}(k)$ is the vector of measurements at time k and $\boldsymbol{e}(k)$ the vector of errors. The response and error is determined for all M response positions and is used to update the input force estimate:

$$\boldsymbol{f}(k+1) = \boldsymbol{f}(k) + 2\mu\big\langle \boldsymbol{e}(k)\boldsymbol{H}(k)\big\rangle \tag{4.46}$$

where the angled brackets represent the averaged error gradient which is averaged over all M response positions. No regularization is needed for this method however, the convergence parameter μ is crucial for the performance of the algorithm and needs the be chosen correctly. An upper bound is given for this parameter to ensure convergence:

$$0 < \mu < M\left(\sum_{m=1}^{M}\sum_{i=0}^{I-1} |h_{ms}(i)|^2\right)^{-1} \tag{4.47}$$

The LMS adaptive algorithm has been numerically verified with a SNR of 5 and experimentally verified (SNR unknown) yielding satisfactory results for complex MDoF structures with noisy measurements.

4.3.2. Sensitivity Method

Lu and Law proposed an iterative method using sensitivities of the response on a number of parameters describing the input force as a Fourier series [39]. Iteratively these parameters are found using a gradient based optimization technique. This method uses the physical model description (section 2.1):

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}\dot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = \boldsymbol{S}_p\mathbf{f}(t) \tag{4.48}$$

The input force is parametrized in the form:

$$f^j(t) = \sum_{l=1}^{n}(f_0^j + f_l^j sin(\omega_l^j t)) \tag{4.49}$$

where j accounts for the j-th input force and f_0, f_l and ω_l are the to be identified force parameters. A series of l periodic functions and one constant function is used to represent the input force. Substitution of equation 4.49 into equation 4.48 gives:

$$\boldsymbol{M}^j\ddot{u}^j(t) + \boldsymbol{C}^j\dot{u}^j(t) + \boldsymbol{K}^j u^j(t) = S_p^j\sum_{i=1}^{n}(f_0^j + f_1^j sin(\omega^j t)) \tag{4.50}$$

where $\boldsymbol{M}^j$ is the j-th row of matrix $\boldsymbol{M}$, likewise for $\boldsymbol{C}^j$ and $\boldsymbol{K}^j$. To determine the sensitivities of the response the derivative of equation 4.50 is taken with respect to the force parameters f_0^j, f_1^j and ω^j:

$$\boldsymbol{M}\left(\frac{\partial \ddot{\boldsymbol{u}}}{\partial f_0^j}\right)+\boldsymbol{C}\left(\frac{\partial \dot{\boldsymbol{u}}}{\partial f_0^j}\right)+\boldsymbol{K}\left(\frac{\partial \boldsymbol{u}}{\partial f_0^j}\right)=\boldsymbol{S}_p \tag{4.51}$$

$$\boldsymbol{M}\left(\frac{\partial \ddot{\boldsymbol{u}}}{\partial f_l^j}\right)+\boldsymbol{C}\left(\frac{\partial \dot{\boldsymbol{u}}}{\partial f_l^j}\right)+\boldsymbol{K}\left(\frac{\partial \boldsymbol{u}}{\partial f_l^j}\right)=\boldsymbol{S}_p\sum_{i=1}^{n} sin(\omega_l^j t) \tag{4.52}$$

$$\boldsymbol{M}\left(\frac{\partial \ddot{\boldsymbol{u}}}{\partial \omega_l^j}\right)+\boldsymbol{C}\left(\frac{\partial \dot{\boldsymbol{u}}}{\partial \omega_l^j}\right)+\boldsymbol{K}\left(\frac{\partial \boldsymbol{u}}{\partial \omega_l^j}\right)=\boldsymbol{S}_p\sum_{i=1}^{n} f_l^j\, t cos(\omega_l^j t) \tag{4.53}$$

The sensitivities can be obtained by integration using the Newmark-β method. The goal of this method is to find a vector of force parameters $\boldsymbol{F}=\begin{bmatrix} f_0 & \cdots & f_n, & \omega_1 & \cdots & \omega_n\end{bmatrix}^T$ such that the measured response matches that of the determined response with the reconstructed force. This translates to:

$$\boldsymbol{Q}\hat{\boldsymbol{d}}(k)=\boldsymbol{d}(k) \tag{4.54}$$

where $\hat{\boldsymbol{d}}(k)$ is the estimated response using the reconstructed input force, $\boldsymbol{d}(k)$ the measured response and $\boldsymbol{Q}$ a selection matrix matching the DoFs of the estimated data to the measured data. The error is then defined as:

$$\delta\boldsymbol{z}=\hat{\boldsymbol{d}}(k)-\boldsymbol{d}(k) \tag{4.55}$$

The error can also be written in terms of the sensitivities and the change in force parameters:

$$\delta\boldsymbol{z}=\boldsymbol{S\delta F} \tag{4.56}$$

where $\boldsymbol{S}$ has the form:

$$\boldsymbol{S}=\begin{bmatrix} \frac{\partial \ddot{\boldsymbol{u}}(1)}{\partial f_0^j} & \cdots & \frac{\partial \ddot{\boldsymbol{u}}(1)}{\partial f_n^j} & \frac{\partial \ddot{\boldsymbol{u}}(1)}{\partial \omega_1^j} & \cdots & \frac{\partial \ddot{\boldsymbol{u}}(1)}{\partial \omega_n^j} \\ \vdots & & \vdots & \vdots & & \vdots \\ \frac{\partial \ddot{\boldsymbol{u}}(N)}{\partial f_0^j} & \cdots & \frac{\partial \ddot{\boldsymbol{u}}(N)}{\partial f_n^j} & \frac{\partial \ddot{\boldsymbol{u}}(N)}{\partial \omega_1^j} & \cdots & \frac{\partial \ddot{\boldsymbol{u}}(1)}{\partial \omega_n^j} \end{bmatrix} \tag{4.57}$$

where N is the number of time steps. Equation 4.55 can be solved by applying the regularized least-squares method:

$$\delta\boldsymbol{F}=(\boldsymbol{S}^T\boldsymbol{S}-\lambda\boldsymbol{I})^{-1}\boldsymbol{S}^T\delta\boldsymbol{z} \tag{4.58}$$

$$\boldsymbol{F}_{k+1}=\boldsymbol{F}_k+\delta\boldsymbol{F} \tag{4.59}$$

The solution can then be found using the following steps:

1. Solve equation 4.50 for the responses and compute the error $\delta\boldsymbol{z}$ for $k+1$
2. Solve equations 4.51 - 4.53 to obtain the sensitivity matrix $\boldsymbol{S}$
3. Obtain $\boldsymbol{F}$ by solving equations 4.58 and 4.59
4. Repeat step 1-3 until convergence is reached

The iterative process is comparable to the LMS adaptive algorithm as it uses a gradient to converge iteratively towards the solution. However the sensitivity method uses the gradient of the response error with respect to three force parameters whereas the LMS adaptive algorithm uses the averaged gradient over all response positions. An advantage of this method is that only a short duration of the measured response (in theory as least as many time steps as to be identified forces) as well as few sensors are needed. When a system description is not available, this method can be extended to a system and input identification method [40]. A disadvantage of this method is that it was found that it is better at identifying harmonic forces than random forces. This may be atributed to the fact that the force is described as a Fourier series. Increasing the number of series would be a solution however this would also increase computation time.

5 Stochastic Force Identification

All of the above described methods use some sort of formulation for modeling the dynamics of the system. This model never completely represents the actual dynamics and thus always deviates in some way. The same can be said for a sensor that measures a certain quantity. A measurement error will always be present. This chapter presents a number of methods where for both the system and the measurement uncertainty is added to the model. In all of these methods the dynamics is modeled in state-space. The uncertainty is modeled as a white Gaussian noise signal and is added to the state and observation equations:

$$\begin{cases} \mathbf{x}(k+1) & = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{f}(k) + \mathbf{w}(k) \\ \mathbf{d}(k) & = \mathbf{G}\mathbf{x}(k) + \mathbf{J}\mathbf{f}(k) + \mathbf{v}(k) \end{cases} \tag{5.1}$$

where $\boldsymbol{w}(k)$ and $\boldsymbol{v}(k)$ are the process and measurement uncertainty, respectively. These uncertainty signals are assumed to be uncorrelated. Methods for estimation in systems with correlated uncertainty can be found in [2] and [51]. The state space description of equation 5.1 describes the system linearly. However, a system might show non-linear behavior, for example when damping is dependent on position or velocity. The Extended Kalman filter (EKF) can be used in this case. It linearizes the dynamics of the system at each time step around the current state. An implementation of this with the recursive relations for the Kalman filter was given by Ma and Ho [42].

5.1. Kalman Filter

The Kalman Filter is widely used in engineering as a linear quadratic state estimator. The state of a state space system is estimated by weighting between a prediction and a measurement. The prediction is done by a known dynamical model of the system. This prediction is updated by information from the measurement. The value of the prediction and measurement (how much the prediction and measurement are trusted) is determined by the Kalman gain which depends on the uncertainty in the prediction and measurement. This produces an estimate of the state that is Minimum-Variance Unbiased (MVU). This says the average of the error between the true state and the predicted state is zero and its variance is minimized. A derivation of the Kalman gain is given in appendix D. Mathematically, the Kalman Filter can be written down as follows:

Prediction step:

$$\hat{\mathbf{x}}(k|k-1) = \mathbf{A}\hat{\mathbf{x}}(k-1|k-1) + \mathbf{B}\mathbf{f}(k) \tag{5.2}$$

$$\mathbf{P}(k|k-1) = \mathbf{A}\mathbf{P}(k-1|k-1)\mathbf{A}^T + \mathbf{Q} \tag{5.3}$$

where $\hat{\mathbf{x}}(k|k-1)$ is the estimated state at time k given the information at time $k-1$, and $\hat{\mathbf{x}}(k-1|k-1)$ the estimated state at time $k-1$ given the information at time $k-1$. $\mathbf{P}$ is the covariance matrix of the error between the true state and the estimated state defined as:

$$\mathbf{P} = E\big((\mathbf{x}-\hat{\mathbf{x}})(\mathbf{x}-\hat{\mathbf{x}})^T\big) \tag{5.4}$$

$\mathbf{Q}$ is defined as the covariance matrix of the process uncertainty.

Measurement step:

$$\mathbf{K}(k) = \mathbf{P}(k|k-1)\mathbf{C}^T(\mathbf{C}\mathbf{P}(k|k-1)\mathbf{C}^T + \mathbf{R})^{-1} \tag{5.5}$$

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{K}(k)(\mathbf{y}(k) - \mathbf{C}\hat{\mathbf{x}}(k|k-1)) \tag{5.6}$$

$$\mathbf{P}(k|k) = (\mathbf{I} - \mathbf{K}(k)\mathbf{C})\mathbf{P}(k|k-1) \tag{5.7}$$

where $\mathbf{K}(k)$ is the Kalman Gain and $\mathbf{R}$ is the covariance matrix of the measurement noise.

5.1.1. Augmented Kalman Filter

The equations for the Kalman filter described above optimally estimate the state of the system, not the input force. To also estimate the input force, the state can be augmented with the input force [38, 41, 46, 53]. The augmented state will then become:

$$\mathbf{x}_a(k) = \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{f}(k) \end{bmatrix} \tag{5.8}$$

where $\mathbf{x}_a(k)$ is the augmented state and $\mathbf{f}(k)$ is the input force. The progression of the input force over time can be modelled as a zeroth-order random walk process which is a process that progresses randomly over time. It is defined as:

$$\mathbf{f}(k+1) = \mathbf{f}(k) + \boldsymbol{\eta}(k) \tag{5.9}$$

where $\boldsymbol{\eta}(k)$ is a zero-mean stochastic process that when chosen correctly allows for reconstruction of the input force. This equation shows that $\boldsymbol{f}$ can vary over time by parameter $\boldsymbol{\eta}$. This parameter can be seen as a smoothness term in the solution. A low $\boldsymbol{\eta}$ allows for little variation in $\boldsymbol{f}$ giving a smooth solution. A high $\boldsymbol{\eta}$ allows for high variation in $\boldsymbol{f}$ giving a less smooth solution. Allowing the input force estimate the progress randomly over time, only restricted by this smoothness term results in an optimal estimate of the input force as it should also comply with the system dynamics in the Kalman filter. The definition of the progression over time of the input force results in the augmented system matrix Φ_a and augmented observation matrix $\mathbf{C}_a$:

$$\mathbf{A}_a = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad \mathbf{G}_a = \begin{bmatrix} \mathbf{G} & \mathbf{J} \end{bmatrix} \tag{5.10}$$

where $\mathbf{J}$ represents an observation matrix that directly links the input force to the measurement vector. This matrix is also called the direct feedthrough matrix. The state and observation equations then become:

$$\begin{cases} \boldsymbol{x}_a(k+1) = \boldsymbol{A}_a\boldsymbol{x}_a(k) + \boldsymbol{\zeta}(k) \\ \boldsymbol{d}(k) = \boldsymbol{G}_a\boldsymbol{x}_a(k) + \boldsymbol{v}(k) \end{cases} \tag{5.11}$$

where

$$\boldsymbol{\zeta}(k) = \begin{bmatrix} \boldsymbol{w}(k) \\ \boldsymbol{\eta}(k) \end{bmatrix} \tag{5.12}$$

The covariance matrix $\boldsymbol{Q}$ is augmented with the covariance of the covariance of the input force $\boldsymbol{S}$:

$$\boldsymbol{Q}_a = \begin{bmatrix} \boldsymbol{Q} & 0 \\ 0 & \boldsymbol{S} \end{bmatrix} \tag{5.13}$$

As this $\boldsymbol{S}$ is related to $\boldsymbol{\eta}$ it can be adjusted in the algorithm the control the smoothness of the solution. This is comparable to the regularization parameter λ in the regularization method. The recursive algorithm of the Augmented Kalman Filter is:

Prediction step:

$$\hat{\mathbf{x}}_a(k|k-1) = \mathbf{A_a}\hat{\mathbf{x}}_a(k-1|k-1) \tag{5.14}$$

$$\mathbf{P}(k|k-1) = \mathbf{A_a}\mathbf{P}(k-1|k-1)\mathbf{A_a}^T + \mathbf{Q}_a \tag{5.15}$$

Measurement step:

$$\mathbf{K}(k) = \mathbf{P}(k|k-1)\mathbf{G_a}^T(\mathbf{G_a}\mathbf{P}(k|k-1)\mathbf{G_a}^T + \mathbf{R})^{-1} \tag{5.16}$$

$$\hat{\mathbf{x}}_a(k|k) = \hat{\mathbf{x}}_a(k|k-1) + \mathbf{K}(k)(\mathbf{d}(k) - \mathbf{G_a}\hat{\mathbf{x}}_a(k|k-1) \tag{5.17}$$

$$\mathbf{P}(k|k) = (\mathbf{I} - \mathbf{K}(k)\mathbf{G_a})\mathbf{P}(k|k-1) \tag{5.18}$$

Whereas the conventional Kalman filter can be applied in real-time, the augmented Kalman filter cannot because of this regularization term $\boldsymbol{S}$. This method has been proven to be successful when collocated measurements that include strain are used. However, when only acceleration measurements are used, the system and observation matrix suffer from unobservability issues. This can be seen by simply writing out the observability of the pair $(\boldsymbol{A}_a, \boldsymbol{G}_a)$ [25]:

$$\mathcal{O} = \begin{bmatrix} \boldsymbol{G}_a \\ \boldsymbol{G}_a \boldsymbol{A}_a \end{bmatrix} = \begin{bmatrix} \boldsymbol{G} & \boldsymbol{J} \\ \boldsymbol{GA} & \boldsymbol{GB} + \boldsymbol{JI} \end{bmatrix} \tag{5.19}$$

When only acceleration measurements are used, $\boldsymbol{G} = \mathbf{0}$. Therefore the first column of the observability matrix $\mathcal{O}$ is zero making the matrix rank deficient. This implies unobservability of the pair $(\boldsymbol{A}_a, \boldsymbol{G}_a)$. Physically this can be interpreted as low frequency or static components of the force that are not observed by acceleration measurement resulting in a drifting behavior in the identified force. Also, when the distance between the location of the force and measurement increases, identification results will deteriorate. This can be attributed to the fact that not only the forces, but also the complete state vector is estimated, resulting in a rapidly increasing level of ill-conditioning as the sensors are moved away from the force location [38].

5.1.2. Dual Kalman Filter

Azam [8] introduces another method based on the Kalman filter. This method is called the Dual Kalman Filter (DKF) as it applies the Kalman filter in two stages to first make an estimate of the force input after which the normal Kalman filter is applied estimating the state of the system. Again the input force is modelled as a zeroth-order random walk:

$$\mathbf{f}(k+1) = \mathbf{f}(k) + \eta(k) \tag{5.20}$$

In the first stage of the DKF this model of the input force can be seen as the state equation. The measurement equation remains unchanged resulting in the following system:

$$\begin{cases} \boldsymbol{f}(k+1) & = \boldsymbol{f}(k) + \boldsymbol{\eta}(k) \\ \mathbf{d}(k) & = \mathbf{Gx}(k) + \mathbf{Jf}(k) + \mathbf{v}(k) \end{cases} \tag{5.21}$$

Applying the Kalman filter to this set of equations results in a MVU estimate of the input force. In the second stage of the DKF the optimal estimate is used as an input in the original set of equation. The Kalman filter is again applied resulting in an optimal estimate of the state. These two recursive steps are applied subsequently resulting in the following recursive equations:

Input estimation:

$$\hat{\mathbf{f}}(k|k-1) = \hat{\mathbf{f}}(k-1|k-1) \tag{5.22}$$

$$\mathbf{P}_f(k|k-1) = \mathbf{P}_f(k-1|k-1) + \mathbf{Q}_f \tag{5.23}$$

$$\mathbf{K}_f(k) = \mathbf{P}_f(k|k-1)\mathbf{J}^T(\mathbf{J}\mathbf{P}_f(k|k-1)\mathbf{J}^T + \mathbf{R})^{-1} \tag{5.24}$$

$$\hat{\mathbf{f}}(k|k) = \hat{\mathbf{f}}(k|k-1) + \mathbf{K}_f(k)(\mathbf{d}(k) - \mathbf{G}\hat{\mathbf{x}}(k-1|k-1) - \mathbf{J}\hat{\mathbf{f}}(k-1|k-1)) \tag{5.25}$$

$$\mathbf{P}_f(k|k) = \mathbf{P}_f(k|k-1) - \mathbf{K}_f(k)\mathbf{J}\mathbf{P}_f(k|k-1) \tag{5.26}$$

State estimation:

$$\hat{\mathbf{x}}(k|k-1) = \mathbf{A}\hat{\mathbf{x}}(k-1|k-1) + \mathbf{B}\hat{\mathbf{f}}(k|k) \tag{5.27}$$

$$\mathbf{P}_x(k|k-1) = \mathbf{A}\mathbf{P}_x(k-1|k-1)\mathbf{A}^T + \mathbf{Q}_x \tag{5.28}$$

$$\mathbf{K}_x(k) = \mathbf{P}_x(k|k-1)\mathbf{G}^T(\mathbf{G}\mathbf{P}_x(k|k-1)\mathbf{G}^T + \mathbf{R})^{-1} \tag{5.29}$$

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{K}_x(k)(\mathbf{d}(k) - \mathbf{G}\hat{\mathbf{x}}(k|k-1) - \mathbf{J}\hat{\mathbf{f}}(k|k)) \tag{5.30}$$

$$\mathbf{P}_x(k|k) = \mathbf{P}_x(k|k-1) - \mathbf{K}_x(k)\mathbf{G}\mathbf{P}_x(k|k-1) \tag{5.31}$$

Whereas the AKF is subject to unobservability issues when input and measurement are non-collocated, the DKF is shown to yield results in these circumstances [6]. Also, it is shown in [8] that the unwanted drifting effect of the AFK can be mitigated with the DKF. However, a model of the propagation of the force is still needed and thus a regularization term ($\boldsymbol{Q}_f$) is needed.

5.2. Joint Input-State Estimation Algorithm

The Joint Input-State Estimation Algorithm was introduced by Gillijns [15] and later studied by Maes [44]. It is comparable to the Kalman filter but uses an optimal estimate as an input. Again the state-space notation for modelling the dynamics of the system and measurement is used:

$$\begin{cases} \mathbf{x}(k+1) & = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{f}(k) + \mathbf{w}(k) \\ \mathbf{d}(k) & = \mathbf{G}\mathbf{x}(k) + \mathbf{J}\mathbf{f}(k) + \mathbf{v}(k) \end{cases} \tag{5.32}$$

An important assumption for this method is that the rank of matrix $\boldsymbol{J}$ should equal the number of applied forces. The Joint Input-State Estimation Algorithm consists of three steps. An input estimation step where an MVU estimate of the input force is produced and a second and third step which are equal to the two steps of the original Kalman filter:

Input estimation:

$$\tilde{\boldsymbol{R}}(k) = \boldsymbol{C}\boldsymbol{P}(k|k-1)\boldsymbol{G}^T + \boldsymbol{R} \tag{5.33}$$

$$\boldsymbol{M}(k) = (\boldsymbol{D}^T\tilde{\boldsymbol{R}}(k)^{-1}\boldsymbol{D})^{-1}\boldsymbol{D}^T\tilde{\boldsymbol{R}}(k)^{-1} \tag{5.34}$$

$$\hat{\boldsymbol{p}}(k|k) = \boldsymbol{M}(k)\big(\boldsymbol{y}(k) - \boldsymbol{C}\hat{\boldsymbol{x}}(k|k-1)\big) \tag{5.35}$$

$$\boldsymbol{P}_{\boldsymbol{p}(k|k)} = (\boldsymbol{D}^T\tilde{\boldsymbol{R}}(k)^{-1}\boldsymbol{D})^{-1} \tag{5.36}$$

Measurement update:

$$\boldsymbol{L}(k) = \boldsymbol{P}(k|k-1)\boldsymbol{C}^T\tilde{\boldsymbol{R}}(k)^{-1} \tag{5.37}$$

$$\hat{\boldsymbol{x}}(k|k) = \hat{\boldsymbol{x}}(k|k-1) + \boldsymbol{L}(k)\big(\boldsymbol{y}(k) - \boldsymbol{C}\hat{\boldsymbol{x}}(k|k-1) - \boldsymbol{D}\boldsymbol{p}(k|k)\big) \tag{5.38}$$

$$\boldsymbol{P}(k|k) = \boldsymbol{P}(k|k-1) - \boldsymbol{L}(k)(\tilde{\boldsymbol{R}}(k)^{-1} - \boldsymbol{D}\boldsymbol{P}_{\boldsymbol{p}(k|k)}\boldsymbol{D}^T)\boldsymbol{L}(k)^T \tag{5.39}$$

$$\boldsymbol{P}_{\boldsymbol{xp}(k|k)} = \boldsymbol{P}^T_{\boldsymbol{xp}(k|k)} = -\boldsymbol{L}(k)\boldsymbol{P}_{\boldsymbol{p}(k|k)} \tag{5.40}$$

Time update:

$$\boldsymbol{x}(k+1|k) = \boldsymbol{\Phi}\hat{\boldsymbol{x}}(k|k) + \boldsymbol{\Gamma}\hat{\boldsymbol{p}}(k|k) \tag{5.41}$$

$$\boldsymbol{P}(k+1|k) = \begin{bmatrix}\boldsymbol{\Phi} & \boldsymbol{\Gamma}\end{bmatrix}\begin{bmatrix}\boldsymbol{P}(k|k) & \boldsymbol{P}_{\boldsymbol{xp}(k|k)} \\ \boldsymbol{P}_{\boldsymbol{xp}(k|k)} & \boldsymbol{P}_{\boldsymbol{p}(k|k)}\end{bmatrix}\begin{bmatrix}\boldsymbol{\Phi}^T \\ \boldsymbol{\Gamma}^T\end{bmatrix} + \boldsymbol{Q} \tag{5.42}$$

An advantage of the joint state-input estimation algorithm is that it does not require any regularization. Result of this is that it can be applied in real-time. When $\boldsymbol{B} = \boldsymbol{J} = 0$ the conventional Kalman filter is obtained. It is noted that the joint input-state estimator as well as the AFK and the DKF can be used for time-varying systems when the time-varying system equations are known.

5.3. Conclusions

The main conclusions from this literature research are stated here. Next, all given methods are assembled in a table and evaluated for the requirements listed in the introduction. The "*accuracy*" requirement is left out as comparative implementation will give answer to this question and will be done in following research. The computation time is evaluated according to the Big-O norm stating the order between the input size and the number of computations.

5.3.1. Regularization Methods

It is found that when an analytical deconvolution method is used it has to be regularized when dealing with an ill-posed problem. This approach scales quadratic with the number of time steps. The regularization term can be seen as a smoothing term and thus accounts for noise. Also, regularization methods can only be applied when input and output are collocated.

5.3.2. Dynamic Programming

Both Classical Dynamic Programming and Extented Dynamic Programming only yield good results when the input and output are collocated. Dynamic Programming methods are sensitive to noise and might need pre-filtering before used in the Dynamic Programming algorithm. Computation time increases significantly with an increase in system order.

5.3.3. Least Mean Square Adaptive Algorithm

The LMS adapative algorithm yields satisfactory results, is able to handle noise and non-collocated input-output pairs. As is uses the IRFs directly is handles non-stationary effect and computation time scales linearly with the size of the input.

5.3.4. Sensitivity Method

The sensitivity method is insensitive to measurement noise or model error. An advantage is that in theory only a short duration of the measurement is needed. As the force is described as a Fourier series, non-stationary effect might be somewhat more difficult to capture. When no model of the system is available, an extension to this method can be used to simultaneously identify the system and input force.

5.3.5. Augmented Kalman Filter

When only acceleration measurements are used, the AKF shows drift in the identified force as low frequency or static input are not observed. It also suffers from unobservability issues when non-collocated sensors are used. The AKF requires a regularization term to be chosen.

5.3.6. Dual Kalman Filter

The drift and unobservability issues experienced by the AKF are solved with the DKF, however it still requires regularization.

5.3.7. Joint Input-State Estimator

The JISE is applicable for noisy measurements from non-collocated acceleration sensors. Also, it is real-time applicable. All Kalman filter methods are appicable for time-varying systems when the time-varying system matrices are known.

5.3.8. Table of Conclusions

The conclusions are shown in the table below.

Method	MIMO	Non-stat. effects	Noise	Acc. measurement	Non-coll. sensors	Computation time
TSVD	✓	✓	✓	✓	✗	$\mathcal{O}(n^2)$
TR	✓	✓	✓	✓	✗	$\mathcal{O}(n^2)$
TVR	✓	✓	✓	✓	✗	$\mathcal{O}(n^2)$
GTR	✓	✓	✓	✓	✗	$\mathcal{O}(n^2)$
CDP	✓	✓	✗	✗	✗	$\mathcal{O}(n)$
EDP	✓	✓	✗	✓	✗	$\mathcal{O}(n)$
LMS	✓	✓	✓	✓	✓	$\mathcal{O}(n)$
SM	✓	✗	✓	✓	✓	$\mathcal{O}(n)$
AKF	✓	✓	✓	✗	✗	$\mathcal{O}(n)$
DFK	✓	✓	✓	✓	✓	$\mathcal{O}(n)$
JISE	✓	✓	✓	✓	✓	$\mathcal{O}(n)$

Table 5.1: Force identification method comparison

From this table it can be concluded that regularization methods are not suitable for force identification for TPA because these methods do not allow for non-collocated input-output pairs as well as their quadratic relation between input size and computation time. Dynamic programming method do not handle noise well and do not allow for non-collocated input-output pairs. Therefore they are not to be used for force identification for TPA. The AKF shows drift when only accelerations measurement is used, and again it does not allow for non-collocated input-output pairs. It is not suitable for force identification for TPA. Promising methods are the LMS adaptive algorithm, the DKF and the JISE.

II

Evaluation

In part II of this thesis the most suitable force identification methods are evaluated and compared in order to test their identification performance and to identify their strengths and weaknesses.

The most suitable identification methods are obtained by the conclusions of part I presented in the table of chapter 5.3.8. These methods are the Dual Kalman Filter, the Joint Input-State Estimator and the Least-Mean-Square Adaptive Algorithm.

To test these methods in an effective manner, firstly the most simple system is used: a simulated one DoF mass-spring-damper system. Complexity of simulation is then increased to a nine DoF system after which the methods are exposed to experimental data.

During simulation different weaknesses of the methods were found. A number of adjustments to the methods are therefore presented with the goal to counteract these weaknesses.

6

Simulated Evaluation

In order to obtain an answer to the question which method is most applicable for Transfer Path Analysis the Dual Kalman Filter, Joint Input-State Estimator and the Least Mean Square Adaptive Algorithm are studied numerically in this chapter. Their performance concerning stability, accuracy and speed is compared. To evaluate the above described methods, simulation of two different dynamical systems is executed in MATLAB. Firstly, the performance of the methods are evaluated on an SDoF mass-spring-damper-system. The behavior of an SDoF system is predictable and intuitive and this can be used to analyze the different methods. Next, complexity is increased to a nine DoF system to analyze the performance of the algorithms on a more complex and MDoF system.

The system will be excited by three different input forces. Firstly an impulsive force as in theory an impulse excites all frequencies and thus all dynamics are captured in one signal even though the input itself is not harmonic. In practice only frequencies up to a certain frequency are excited as an input in discrete time can only approach a Dirac impulse and is limited by the time step used. Secondly an harmonic sweep is used as an input. This signal will test the algorithms on its capability to identify harmonic signals as well as a time varying signal which mimics for example an engine run-up. Also, a harmonic sweep excites all frequencies up to its highest frequency. Lastly, a random input is used as it is commonly encountered in the industry.

The response to the excitation is determined using the state space matrices and equation 2.37. Gaussian white noise is added to account for measurement uncertainty with a SNR of 100.

6.1. Single Degree of Freedom Mass-Spring-Damper

A SDoF system is used to evaluate the different algorithms in the most basic manner. The schematics of the SDoF system is given in figure 6.1.

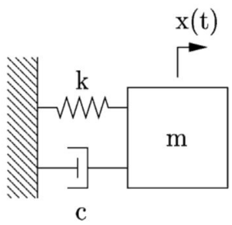


Figure 6.1: Schematics of SDoF mass-spring-damper system

The system is characterized by the following values:

$$m = 10\ kg \tag{6.1}$$

$$c = 1 * 10^2\ Ns/m \tag{6.2}$$

$$k = 1 * 10^4\ N/m \tag{6.3}$$

With a time step of 0.002 s this results in the following Impulse Response Function (IRF):

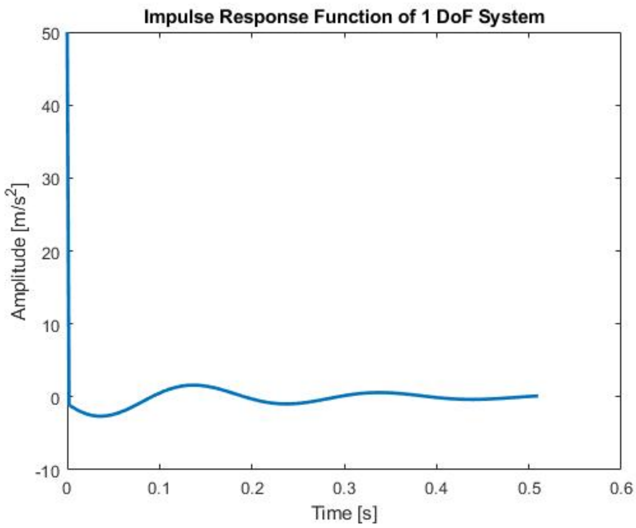


Figure 6.2: Impulse Response Function of SDoF mass-spring-damper system

This IRF will be used as the system dynamics description in the least mean square (LMS) adaptive algorithm. The state space matrices used for the dual Kalman filter (DKF) and the joint input-state estimator (JISE) will be derived using equations from section 2.5. Only acceleration observation will be used giving the following selection matrices:

$$S_a = 1 \tag{6.4}$$

$$S_v = 0 \tag{6.5}$$

$$S_d = 0 \tag{6.6}$$

As the noise level is known exactly the covariance matrix R used in the DKF and the JISE is equal to the covariance of the added noise:

$$R = \left(\frac{max(|\boldsymbol{d}|)}{SNR} \right)^2 \tag{6.7}$$

System uncertainty is neglected in this research.

6.1.1. Dual Kalman Filter

The DKF (section 5.1.2) is used to identify the three forces described above. The identified forces are shown in figure 6.3. This figure shows that the DKF is capable of partly reconstructing the input force on an SDoF system. However, a drift from the true force is present in all three identified forces. Even though stated in [6] that the DKF should resolve the drift issues experienced by the augmented Kalman filter, this unstable drift behavior is visible in the reconstructed forces. This implies that the DKF as it is is not suitable for force identification when only acceleration measurements are used.

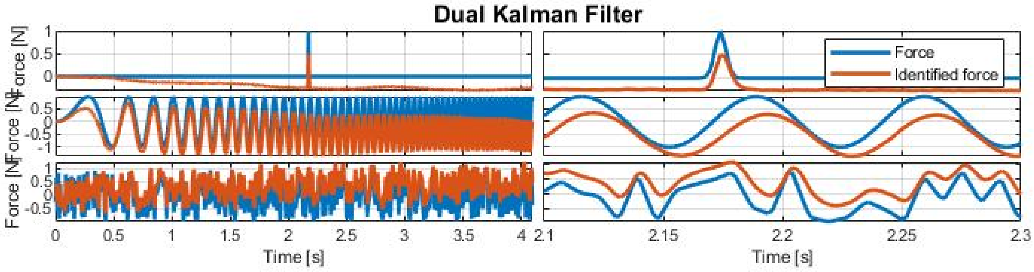


Figure 6.3: Identification of impulsive, harmonic sweep and random force by the DKF

6.1.2. Joint Input-State Estimation Algorithm

The performance of the JISE (section 5.2) is comparable to the performance of the DKF. It is able to reconstruct the input force however, it also is subject to unstable drift behavior. This unwanted instability makes the JISE unsuitable of force identification when only acceleration measurements are used.

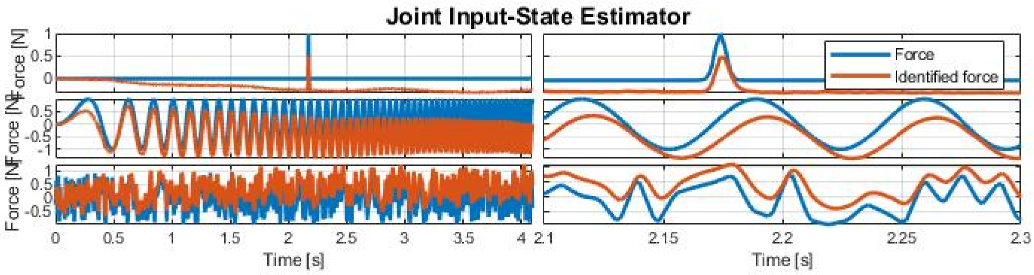


Figure 6.4: Identification of impulsive, harmonic sweep and random force by the JISE

6.1.3. Stability of the Kalman Filter

As observed in the results of the DKF and the JISE the Kalman filter is not unconditionally stable. Therefore the stability of these methods requires to be studied. The stability of the inversion of a system depends on the poles of the inverse system and therefore on the zeros of the system itself [43]. The definition of a zero can be given by:

$$rank\left(\begin{bmatrix} \boldsymbol{A}-\lambda_j\boldsymbol{I} & \boldsymbol{B} \\ \boldsymbol{G} & \boldsymbol{J} \end{bmatrix}\right) < n_s + min(n_p, n_d) \tag{6.8}$$

where λ_j represents the j^{th} zero, n_s the number of states of the state space system, n_p the number of input forces and n_d the number of measurements. For all zeros there exist a vector of states and forces such that:

$$\begin{bmatrix} \boldsymbol{A}-\lambda_j\boldsymbol{I} & \boldsymbol{B} \\ \boldsymbol{G} & \boldsymbol{J} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{f} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \tag{6.9}$$

An input $\boldsymbol{f}$ on the system will thus result in a response of zero making the input unobservable by the measurement. If the zeros of the discrete time system are located within the unit circle the inversion is stable. If the zeros are outside the unit circle the inversion is unstable. If one or more zeros lie on the unit circle the inversion is said to be marginally stable. If only acceleration measurements are used there will always be at least one marginally stable zero corresponding to static excitations. Or in physical terms: static excitations cannot be observed by acceleration measurements. This causes instability in the Kalman filter which expresses itself as drift behavior in the identified force.

Another way to look as this instability is by looking at the intergratory nature of the Kalman filter. Within the Kalman filter, the state of the system is estimated. When only acceleration is measured, the Kalman filter will integrate this measurement over time. The integration of a noisy signal over time known the be subject to drift. This can be demonstrated by leaving out the noise on the measurement. As there is no noise on the signal, there is no noise to accumulate and the identification is very accurate. The figure below shows the identification of a sinusoid sweep by the DKF using a measurement signal without noise.

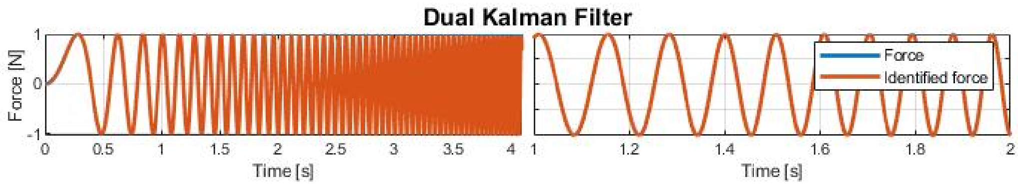


Figure 6.5: Identified sinusoid sweep by DKF without the presence of noise

The same results are found for the JISE:

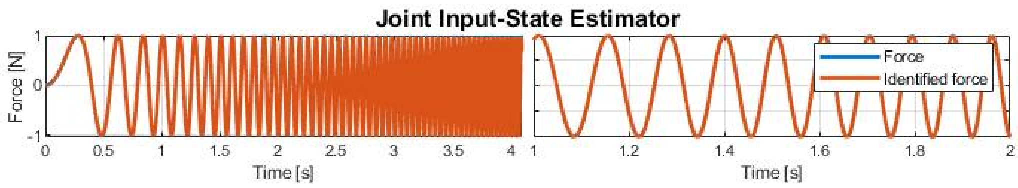


Figure 6.6: Identified sinusoid sweep by JISE without the presence of noise

It can thus be concluded that the noise on a measured acceleration signal inherently in the cause of instability of the DKF and JISE. Measurements can be very accurate but noise will never be completely eliminated. The DKF and JISE as described above can thus not be used for force identification with only acceleration measurements, however it is possible to stabilize both filters. This will be described in the following sections.

6.1.4. Resolving Drift

In the previous section it is mentioned that drift occurs when only acceleration and/or velocity measurements are used. A straightforward solution to this problem then would seem to (also) use displacement measurements. As stated in [46] the number of displacement measurements should be equal of larger than the number of input forces to ensure stable inversion. The figure below shows the results of the JISE for a sine sweep when both a displacement and acceleration measurement is used. This result confirms that a displacement measurement resolves the drift issue experienced by the Kalman filters.

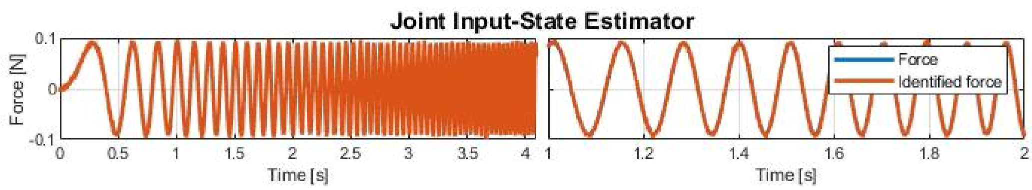


Figure 6.7: Identification of sinusoid sweep by the JISE with displacement and acceleration measurement

However, the implementation of displacement measurements is not straightforward. The implementation of various displacement measurement methods using for example strain gauges, LVDTs or optics is either costly or requires intensive installation and calibration. Displacement measurement for structural dynamics is being researched extensively and possibly valuable methods are proposed by for example [14, 34]. Exploration of these methods is beyond the scope of this research.

6.1.5. Artificial Displacement Measurements

As a displacement measurement resolves the drift issue in the Kalman filter and a physical displacement measurement is complex to implement an artificial displacement measurements (ADM) can be used. The

Kalman filter is given a vector of zeros next to the physical acceleration measurement telling it displacement is restricted. However, to comply with the vibrations measured by the accelerometers a predefined amount of freedom is given to the ADM. This freedom is defined in the form of a covariance matrix $\boldsymbol{R}_{adm}$. The state space matrices of the observation equation will become:

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \\ -\boldsymbol{S}_a\boldsymbol{M}^{-1}\boldsymbol{K} & -\boldsymbol{S}_a\boldsymbol{M}^{-1}\boldsymbol{C} \end{bmatrix} \tag{6.10}$$

The covariance matrix on the measurement will be extended with matrix $\boldsymbol{R}_{adm}$:

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{R}_{adm} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R} \end{bmatrix} \tag{6.11}$$

Covariance matrix R_{adm} must be chosen such that it minimizes drift whilst preserving the dynamics observed by the acceleration measurement. This approach is similar to the method used in [46] where dummy measurements are added to stabilize the augmented Kalman filter. The above described method can be applied to the DKF as well as the JISE. Figure 6.8 shows the results of of the JISE with ADM and acceleration measurement for a impulsive, harmonic sweep and random force. It can be concluded that results have improved compared to the results without ADM presented in figure 6.4 and that the ADM have stabilized the JISE. The same improvement can be found with the DKF. It must be noted that the performance of these filters in combination with ADM highly depends on the covariance matrix R_{adm}. The optimal values in this diagonal matrix depend on the measured accelerations and can therefore be scaled to these acceleration measurements.

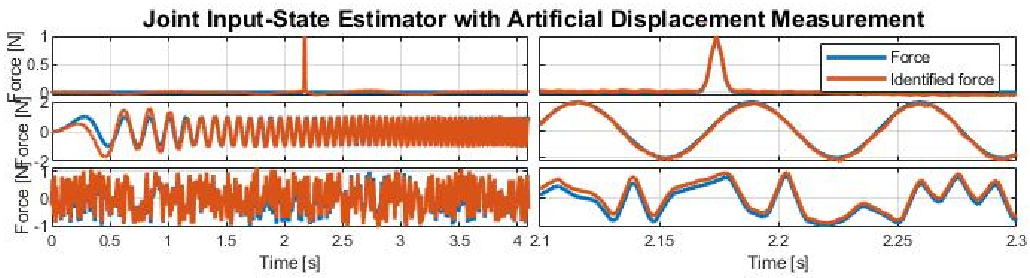


Figure 6.8: Identification of impulsive, harmonic sweep and random force by the JISE with ADM

6.1.6. Dual Kalman Filter vs Joint Input-state Estimator

The performance of the DKF and the JISE is comparable and shows the same drift behavior which can be resolved with adding ADM. It is found that their performance matches exactly when the regularization parameters Q_f is infinity. Figure 6.9 shows the solution found by the DKF with the the regularization parameters Q_f set to $1 * 10^{30}$. Figure 6.10 shows the same solution has been found with the JISE.

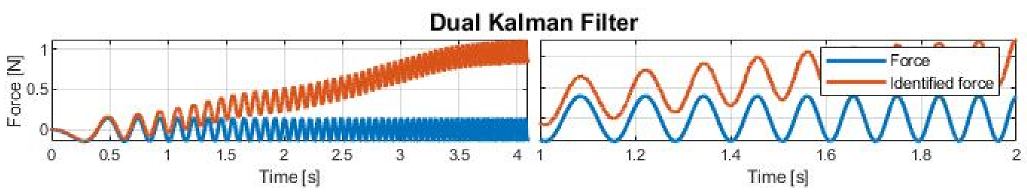


Figure 6.9: Identification of harmonic sweep with the DKF with $Q_f = 1 * 10^{30}$

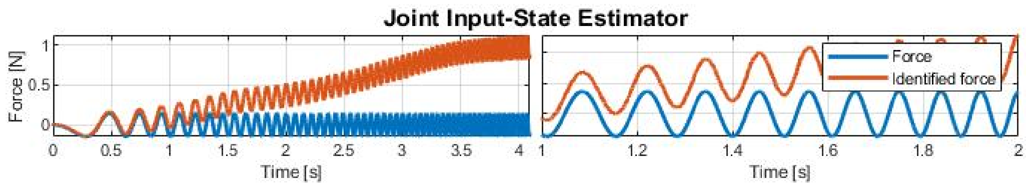


Figure 6.10: Identification of harmonic sweep with the JISE

This can be confirmed mathematically with the definition of the Kalman gain for the DKF and the JISE. The Kalman gain for input estimation for the DKF is determined with equations 5.23 and 5.24 repeated here for convenience:

$$\mathbf{P}_f(k|k-1) = \mathbf{P}_f(k-1|k-1) + \mathbf{Q}_f \tag{6.12}$$

$$\mathbf{K}_f(k) = \mathbf{P}_f(k|k-1)\mathbf{J}^T(\mathbf{J}\mathbf{P}_f(k|k-1)\mathbf{J}^T + \mathbf{R})^{-1} \tag{6.13}$$

When looking at the SDoF case all matrices in both equations are scalars. The definition of the Kalman gain $\boldsymbol{K}_f$ can be rewritten as:

$$\boldsymbol{K}_f = \frac{\left(\boldsymbol{P}_f(k-1|k-1) + \boldsymbol{Q}_f\right)\boldsymbol{J}}{\boldsymbol{J}\left(\boldsymbol{P}_f(k-1|k-1) + \boldsymbol{Q}_f\right)\boldsymbol{J} + \boldsymbol{R}} \tag{6.14}$$

under the condition that:

$$\boldsymbol{Q}_f >> \boldsymbol{P}_f(k-1|k-1) \tag{6.15}$$

which results in:

$$\boldsymbol{K}_f = \frac{\boldsymbol{Q}_f\boldsymbol{J}}{\boldsymbol{J}\boldsymbol{Q}_f\boldsymbol{J} + \boldsymbol{R}} \tag{6.16}$$

Also:

$$\boldsymbol{J}\boldsymbol{Q}_f\boldsymbol{J} >> \boldsymbol{R} \tag{6.17}$$

which leaves:

$$\boldsymbol{K}_f = \frac{1}{\boldsymbol{J}} \tag{6.18}$$

The definition of the Kalman gain for input estimation of the JISE are stated in equations 5.33 and 5.34, repeated here for convenience:

$$\tilde{\boldsymbol{R}}(k) = \boldsymbol{G}\boldsymbol{P}(k|k-1)\boldsymbol{G}^T + \boldsymbol{R} \tag{6.19}$$

$$\boldsymbol{M}(k) = (\boldsymbol{J}^T\tilde{\boldsymbol{R}}(k)^{-1}\boldsymbol{J})^{-1}\boldsymbol{J}^T\tilde{\boldsymbol{R}}(k)^{-1} \tag{6.20}$$

In the SDoF case G is a matrix. However, as P is a scalar the term GPG^T becomes a scalar. With this taken into account the Kalman gain M can be determined with scalar math operations:

$$\boldsymbol{M}(k) = \frac{\boldsymbol{G}\boldsymbol{P}(k|k-1)\boldsymbol{G}^T + \boldsymbol{R}}{\boldsymbol{J}^2}\frac{\boldsymbol{J}}{\boldsymbol{G}\boldsymbol{P}(k|k-1)\boldsymbol{G}^T + \boldsymbol{R}} \tag{6.21}$$

which results in the same definition for the Kalman gain for input estimation:

$$\boldsymbol{M} = \frac{1}{\boldsymbol{J}} \tag{6.22}$$

This can be extended to MDoF systems as all terms ($\boldsymbol{P}$, $\boldsymbol{P}_f$, $\boldsymbol{Q}_f$, $\boldsymbol{J}$ and $\boldsymbol{R}$) are diagonal, $\boldsymbol{Q}_f$ by definition, $\boldsymbol{P}_f$ by the nature of the equations of the algorithm, $\boldsymbol{J}$ by definition and $\boldsymbol{R}$ by the assumption that the noise is uncorrelated. With this convenient property of these matrices the equations above hold.

To conclude, it is shown numerically and mathematically that when the DKF is given infinite freedom it yields the same results as the JISE. Improvements on this results by optimizing the regularization parameter $\boldsymbol{Q}_f$ are found the be minimal (figures 6.3 and 6.4). Therefore preference is given to the JISE as it does not require the optimization of a parameter. From hereon only the JISE will be used.

6.1.7. Least Mean Square Adaptive Algorithm

The same simulated responses as used in sections 6.1.1 and 6.1.2 are used to determine the input force with the LMS algorithm (4.3.1). The results are shown in figure 6.11. Apart from the low frequency input at the start of the harmonic sweep, the LMS algorithm is capable of retrieving the input signals on an SDoF system. The lesser results for low frequency inputs can be attributed to the fact that only acceleration measurements are used which capture low frequency content (below the lowest eigenfrequency) less than higher frequency content. The eigenfrequency of this SDoF system is at 31 rad/s and the harmonic sweep starts at 20 rad/s thus this results is expected.

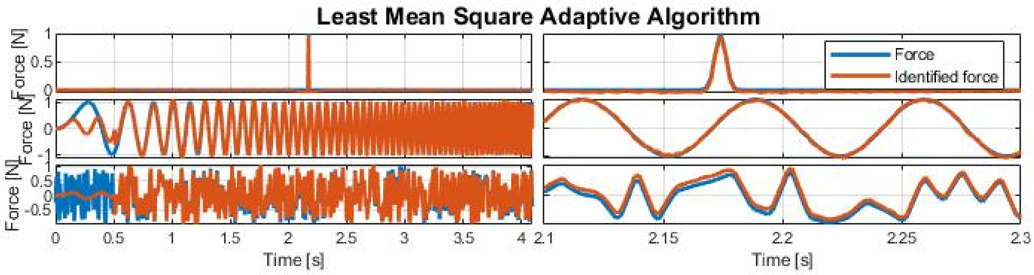
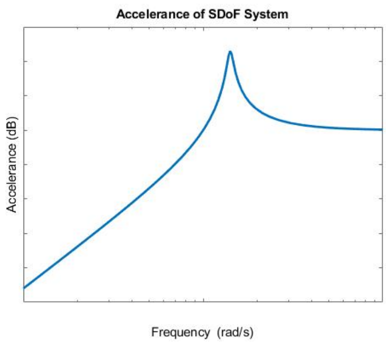
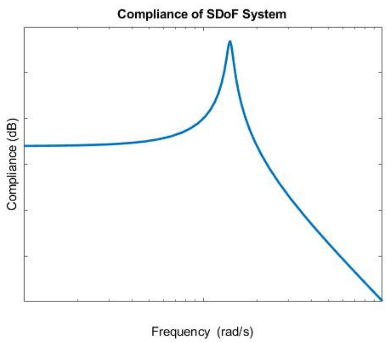


Figure 6.11: Identification of impulsive, harmonic sweep and random force by the LMS algorithm

In general it can be said that beyond the ultimate eigenfrequency the performance of the LMS algorithm deteriorates as the input is not observed by the measurement. In the case of acceleration measurements an input below the lowest eigenfrequency will become more unobservable as the frequency of the input decreases. This is shown in figure 6.12a which shows the accelerance of a SDoF system. When displacement measurements are used the performance of the LMS algorithm deteriorates as the frequency of the input is larger than the largest eigenfrequency of the system. This is shown in figure 6.12b which shows the compliance of a SDoF system.



(a) Accelerance of SDoF system



(b) Compliance of SDoF system

Figure 6.12: Accelerance and compliance of SDoF system

6.1.8. Stability of the LMS Algorithm

The LMS algorithm is based on steepest descent. The step size of the steepest descent algorithm can be controlled by the parameter $\boldsymbol{\mu}$ which determines the speed of convergence. However, when the step size is chosen too large the algorithm becomes unstable. A stability criterion is given in [58]:

$$0 < \mu_s < M\left(\sum_{m=1}^{M}\sum^{I-1}|h_m(i)|^2\right)^{-1} \tag{6.23}$$

In the case of an MDoF system the vector μ can be normalized to corresponding impulse responses to make full use of this convergence criterion. This ensures equal convergence over the different inputs. The normalized convergence criterion can be written as:

$$\mu_s = cM\left(\sum_{m=1}^{M}\sum^{I-1}|h_m(i)|^2\right)^{-1} \tag{6.24}$$

with c the parameter to control the speed of convergence which has to comply to

$$0 < c < 1 \tag{6.25}$$

to ensure stability.

6.1.9. Extension to the Least Mean Square Adaptive Algorithm

When a given response signal is used to identify the force that caused it this force can only be identified from time index O where O is the size of the impulse response [58]. This is due to the definition of the input force in the algorithm:

$$\boldsymbol{f}(k) = \begin{bmatrix} f(k) & f(k-1) & \dots & f(k-O+1) \end{bmatrix}^T \tag{6.26}$$

The algorithm starts at time index k where the input force vector is from 1 to O. When this vector is multiplied with the impulse response it results in the response at time step O. Thus, all response data in time steps 1 to $O-1$ is not used. Also, due to the nature of the algorithm, all time indexes of $f(k)$ for $k < O$ are updated less than O times. These two factors results in unreliable identification of the input force for time indexes smaller than O.

This can however be solved fairly easy by adding an array of zeros with size O to the beginning of the input vector. The input force vector for an MDoF system will be:

$$\boldsymbol{f}(k) = \begin{bmatrix} \boldsymbol{O}_{n_p,O} & \boldsymbol{f}(k) \end{bmatrix} \tag{6.27}$$

In stead of starting the algorithm at time index O it can now be started at time index 1 to ensure identification of the full input signal. Figure 6.13 shows the result of the LMS algorithm with input extension.

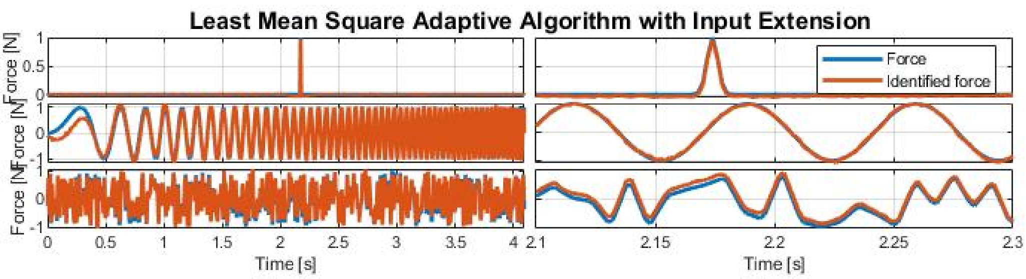


Figure 6.13: Identification of impulsive, harmonic sweep and random force by the LMS algorithm with input extension

It can be concluded that extending the input vector with an array of zeros the size of the IRF resolves the problem of not being able to identify the full input signal. As can be seen from the figure above, full reconstruction of the input is achieved. A downside of this method is that it will increase computation time with an added O time steps per iteration.

6.1.10. Conclusions

In this section the Dual Kalman Filter, Joint Input-State Estimator and the Least Mean Square Adaptive algorithm where compared in terms of their performance on stability and accuracy to test their suitability for Transfer Path Analysis. This was done by means of simulations on a SDoF mass spring damper system. Results have shown that the DKF and JISE showed unstable behavior when only acceleration measurements are used. From this observation can be concluded that these method are unsuitable for force identification in the case that only acceleration measurements are used. This instability can be resolved by adding displacement measurements to the acceleration measurements. If however, displacement measurements are not available it is possible to stabilize both filters by adding Artificial Displacement Measurements (ADM). It is shown that for both true and artificial displacement measurements results are adequate. It is also shown numerically and mathematically that when the regularization parameter of the DKF is set to infinity both the DKF and JISE show the same behavior. An extension to the LMS algorithm is proposed of which the results show that the full input signal can be accurately reconstructed. The LMS algorithm holds two major disadvantages: its incapability to identify forces with a response below the lowest eigenfrequency (in the case of acceleration measurements) and its computation time which will be elaborated on in sections 6.3 and 6.4.1.

6.2. Multi Degree of Freedom Mass-Spring-Damper System

To evaluate the performance of the algorithms in a more complex scenario which becomes more like a TPA scenario a nine DoF system is used of which the schematics are shown in figure 6.14. The connection between masses m_{A6} and m_{B1} and between m_{A7} and m_{B2} are considered rigid.

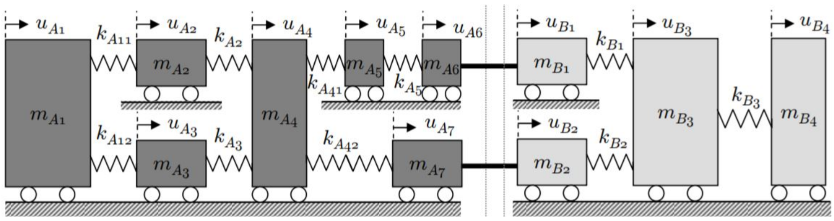


Figure 6.14: Schematics of MDoF mass-spring-damper system [67]

This system is excited at masses m_{A1} and m_{A7} simultaneously by an impulsive, harmonic sweep and random force as done in section 6.1. To the response Gaussian white noise is added with a SNR of 100 and is used as an input to the JISE with ADM and the LMS algorithm.

6.2.1. Joint Input-State Estimation Algorithm with Artificial Displacement Measurement

Figure 6.15 shows the impulsive force on first and seventh mass of the nine DoF mass-spring-damper system as identified by the JISE with ADM. The results show that it is capable of correctly identifying this force on this system.

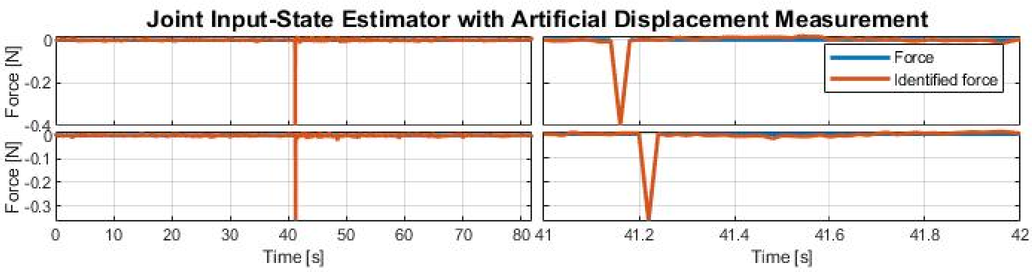


Figure 6.15: Identification of impulsive force by the JISE with ADM

Figure 6.16 shows this forces found by the JISE with ADM when the nine DoF system is excited by a harmonic sweep input. It can be observed that the the sweep is for the largest part reconstructed accurately however results in the low frequencies are less accurate. This can be attributed to the setting of the covariance matrix for ADM. The optimal setting for this matrix depends on the change of the acceleration response. Recalling the function of this matrix, it determines the freedom the displacement can be on the locations of measurement. Therefore setting this matrix too high results in the unstable and unwanted drift behavior. On the other hand, setting it too low results in too little freedom in terms of displacement which results in a situation where no force can be found that matches both the acceleration and artificial displacement measurements. A trade-off has to be made between allowing enough freedom while still stabilizing the filter. Therefore this method is less suitable for input signals that excite both very high and low frequencies.

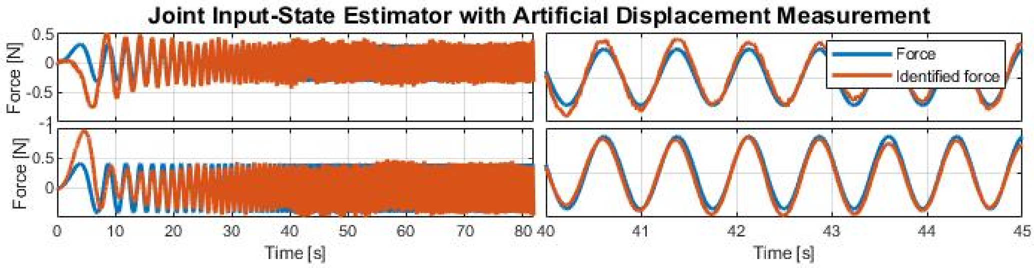


Figure 6.16: Identification of sinusoidal sweep force by the JISE with ADM

The result of the JISE with ADM for a random excitation is shown in figure 6.17. It can be seen that drift behavior cannot be eliminated completely.

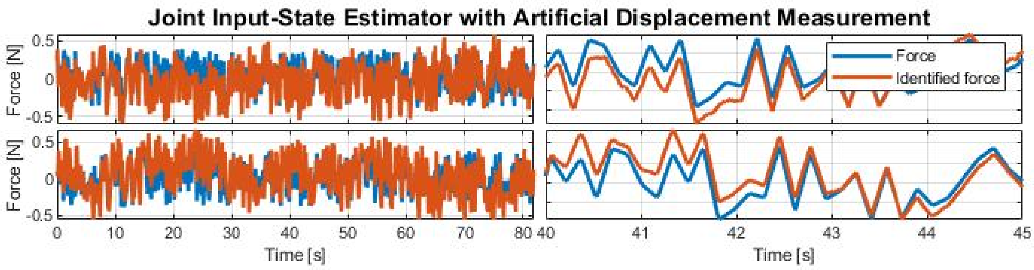


Figure 6.17: Identification of random force by the JISE with ADM

6.2.2. Least Mean Square Adaptive Algorithm

Figure 6.18 shows the results of the LMS algorithm on an impulsive input force. Results are satisfactory.

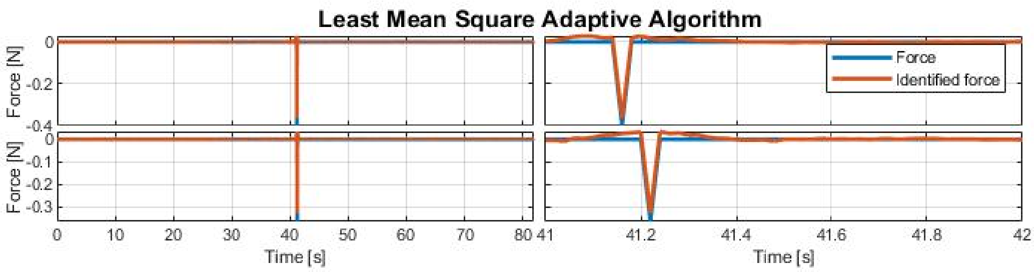


Figure 6.18: Identification of impulsive force by the LMS algorithm

Figure 6.18 shows the results of the LMS algorithm on an harmonic sweep input force. For high frequency content this algorithm performs well. However, as visible in the plot low frequency inputs are not identified. This is due to the fact that only acceleration measurements are used where low frequencies are less visible.

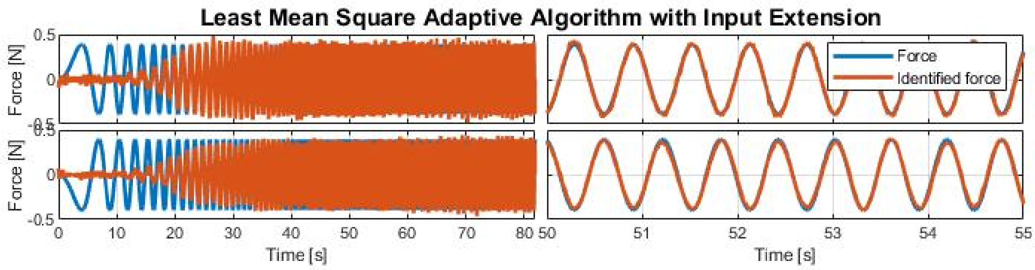


Figure 6.19: Identification of sinusoidal sweep force by the LMS algorithm

Are random force has been used as an input to the LMS algorithm. This identification results are shown in figure 6.20. Performance of this algorithm on a random input force is satisfactory.

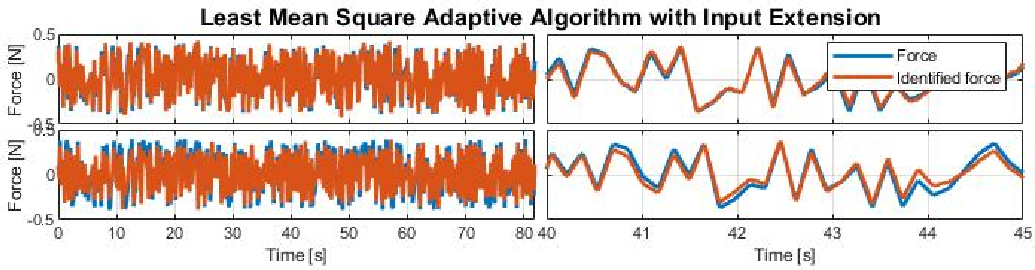


Figure 6.20: Identification of random force by the LMS algorithm

6.2.3. Conclusions

The JISE with ADM and the LMS algorithm are exposed to a MDoF system to test their performance. The results show that the JISE with ADM gives satisfactory results on impulsive forces, however when the system is excited with an harmonic sweep and a random force its performance is somewhat less satisfactory. Results are highly dependent on the covariance matrix for ADM. It can be concluded that this method is less suitable for inputs that excite both very high and low frequencies as identifying them requires a different setting for the covariance matrix. As long as this condition is met, the JISE with ADM is suitable to reconstruct forces on a numerical MDoF system and therefore has the potential to be suitable for TPA. The results for the LMS algorithm shows that this method is very accurate, apart from the unobservable low frequency content which was also seen in the SDoF case. It can therefore be concluded that the LMS algorithm it suitable to identify forces on an MDoF system of which the responses are obtained numerically. This implies that it has high potential to be suitable for TPA.

6.3. Convergence Speed of the Least Mean Square Adaptive Algorithm

The results described above show that the LMS algorithm is able to retrieve the input forces on an SDoF and MDoF system whenever the input force does not contain low frequency content. The biggest disadvantage of this method is its computation time as it is an iterative method. This section will elaborate of different methods to increase the convergence speed of the LMS algorithm to decrease computation time. All methods described below are based adjusting the step size of the algorithm.

6.3.1. Normalized Step Size

A first method to investigate is the normalized step size which was first introduced in [47]. The step size is normalized with the power of the input signal. When this is applied to the original LMS algorithm (figure 4.1a) where the input signal changes over time, this method can be considered a variable step size method. However, in this research the input signal of the LMS algorithm is the IRF which is constant over time and the the step size will thus not change over time. The step size will only be scaled to the power of the corresponding

IRF, averaged over the response positions. In mathematical terms:

$$\mu_s = \tilde{\mu} M \left(\sum_{m=1}^{M} \sum^{I-1} |h_m(i)|^2 \right)^{-1} \tag{6.28}$$

with:

$$0 < \tilde{\mu} < 1 \tag{6.29}$$

It can be observed that when $\tilde{\mu} = 1$ the upper bound of the stability criterion (equation 4.47) of the LMS algorithm as proposed by [58] is obtained.

6.3.2. Variable Step Size

A large number of variable step size method are available. An overview of these methods can be found in [12]. These method aim to adjust the step size over time with the use of the measured data to improve the speed of convergence. Most methods use the error (equation 4.45) to define an optimal step size. A disadvantage of these methods is that they require multiple parameters to be set. As the objective is to save time only methods that use two parameters or less are tested in this research. These methods and their update step to adjust the step size are shown below.

Linear:
A variable step size method is proposed that linearly relates the step size to the averaged error. The step size is updated using the following equation:

$$\mu(k) = \alpha \frac{1}{M} \sum_{m=1}^{M} |\boldsymbol{e}(k)| \tag{6.30}$$

where α is a scaling factor.

Power:
The linear relation described above can be extended to obtain a power relation between the step size and the error:

$$\mu(k) = \alpha \frac{1}{M} \sum_{m=1}^{M} |\boldsymbol{e}^{\beta}(k)| \tag{6.31}$$

where α and β are scaling factors.

Barzilai and Borwein:
As the LMS algorithm is essentially a steepest descent algorithm the variable step size method for steepest descent by Barzilai and Borwein can be applied [9]. Its approach is to find the step size that minimizes:

$$\mu_s(k) = \underset{\boldsymbol{x}}{\operatorname{argmin}} ||\Delta \boldsymbol{x} - \mu_s \Delta \boldsymbol{g}(x)||^2 \tag{6.32}$$

where:

$$\Delta \boldsymbol{x} = \boldsymbol{f}_s(k) - \boldsymbol{f}_s(k-1) \tag{6.33}$$

$$\Delta \boldsymbol{g}(x) = \langle \boldsymbol{e}_m(k) \boldsymbol{H}_{ms} \rangle - \langle \boldsymbol{e}_m(k-1) \boldsymbol{H}_{ms} \rangle \tag{6.34}$$

The solution to this minimization problem is found by differentiating with respect to λ:

$$0 = \Delta \boldsymbol{g}(x)^T (\Delta \boldsymbol{x} - \mu \Delta \boldsymbol{g}(x)) \tag{6.35}$$

which results in the optimal step size:

$$\mu = \frac{\Delta \boldsymbol{g}(x)^T \Delta \boldsymbol{x}}{\Delta \boldsymbol{g}(x)^T \Delta \boldsymbol{g}(x)} \tag{6.36}$$

Shan:
A variable step size algorithm has been proposed by Shan [56] which makes use of the 'orthogonality principle' which states that the minimum of the mean-square error is reached when the input signal (of the LMS algorithm :$\boldsymbol{H}$) and the error signal ($e(k)$) are orthogonal. An estimate of this correlation can be determined by the multiplication of the current error and the input signal.

$$\rho(k) = \lambda\rho(k) + (1-\lambda)u(k)e(k) \tag{6.37}$$

The step size is then updated with:

$$\mu(k+1) = \frac{\alpha|\rho(k)|}{\boldsymbol{u}^T\boldsymbol{u}} \tag{6.38}$$

where ρ is a scaling factor and λ a forgetting factor.
Kwong:
Kwong [30] proposed a variable step size method that updates using the following equation:

$$\boldsymbol{\mu}(k+1) = \alpha\boldsymbol{\mu}(k) + \frac{\gamma}{M}\sum_{m=1}^{M} e^2(k) \tag{6.39}$$

where α is a scaling factor and γ is a forgetting factor.
Ang:
A method by Ang [3] uses the following equations to update the step size:

$$\boldsymbol{\mu}(k) = \boldsymbol{\mu}(k-1) + \rho e(k)\boldsymbol{u}^T(k)\boldsymbol{\phi}(k) \tag{6.40}$$

$$\boldsymbol{\phi}(k+1) = \alpha\boldsymbol{\phi}(k) + \boldsymbol{u}(k)e(k) \tag{6.41}$$

where α is a forgetting factor and ρ a scaling factor.
Mathews:
Mathews [45] introduced a variable step size method that uses the following step size update:

$$\mu(k) = \mu(k-1) + \rho e(k)e(k-1)\boldsymbol{u}^T(k-1)\boldsymbol{u}(k) \tag{6.42}$$

where ρ is a scaling factor.

6.3.3. Comparison of Step Size Methods

The methods described above are tested using the MDoF system which is excited by a random force. The convergence of the algorithms is evaluated using the 2-norm of the gradient estimate averaged over all time steps. Figure 6.21 shows that no method increases the speed of convergence compared to the fixed step size LMS algorithm significantly. It must be noted that the methods proposed by Ang and Shan use the input signal to update the step size. The original LMS algorithm uses the input to the system as input to the algorithm which changes over time. The adjusted LMS algorithm for force identification uses the IRFs and input to the algorithm which does not change over time. This reduces the influence of these variable step size methods.

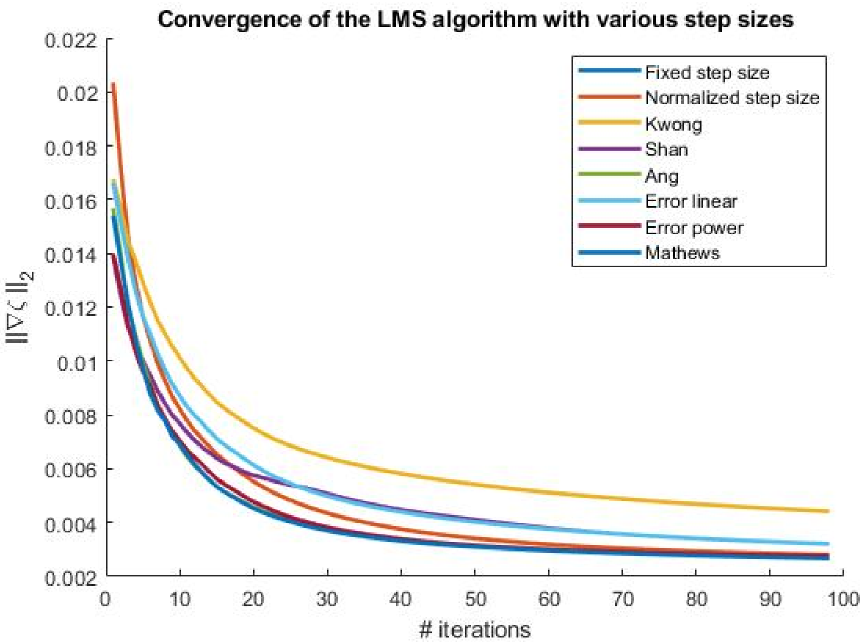


Figure 6.21: Convergence of the LMS algorithm for various step sizes

6.4. Time Aspects

This section addresses two different aspects concerning the time required for both algorithms to come to its solution. Firstly, the computation time itself is discussed and times for both methods in different scenario's are compared. Secondly, the possibility of applying force identification in real-time is addressed. This can be useful in for example control applications.

6.4.1. Computation Time

This section will elaborate on the computation time needed for both methods to find an optimal solution. Computation time is known to be a disadvantage of the LMS algorithm and attempts to improve its convergence speed have been unsuccessful (section 6.3. This section will give a quantitative comparison of the true computation times needed. It must be noted that even though quantitative results are given these results should be looked at relatively only as the computation times obtained are affected by the programming skills of the author.

Both methods are fundamentally different in the way the optimal solution is found. The JISE will recursively use the measurement data for find the optimal Kalman gain relatively fast and will use this optimal Kalman gain to find the optimal solution over one recursion of the full measurement data set. The LMS algorithm however does not find its optimal solution in one recursion. Rather, its need multiple iterations to converge to the solution. In fact, it will always be able to improve its solution with a following iteration and thus will never find its optimal solution. To be able to obtain a reasonable comparison of computation times the LMS algorithm will be stopped when it has found a solution similar to the solution found by the JISE with ADM. As a measure for the correctness of the solution the Root Mean Square Error (RMSE) of the input force is used:

$$RMSE = \left(\frac{\sum_{i=1}^{N} \left(\boldsymbol{f}(t) - \hat{\boldsymbol{f}}(t) \right)^2}{N} \right)^{1/2} \tag{6.43}$$

An measurement signal with 2048 samples and an Impulse Response Function with 1024 samples is used. This fairly small number of samples is possible because of the relatively simple systems that are used for these simulation. When measurements are performed on real structures, typically a larger number of samples is required for both the measurement signal and the Impulse Response Function. The computation times scale linearly with the number of samples. Table 6.1 shows the computation times of the JISE with ADM and the LMS algorithm they needed to obtain a solution with a similar RMSE. Firstly, the computation times of the

JISE with ADM are approximately the same. This is expected as this the time needed to recursively process all the data points by the filter. The second conclusion that can be drawn from this observation is that the LMS algorithm indeed needs more computation time than the JISE with ADM, which was expected. The type of signal that is identified is affecting the computation time. An impulsive force requires the least computation time. This is due to simplicity of the signal. The LMS algorithm starts with an initial vector of zeros for the input force which is correct for the most part of the true input signal. The algorithm only needs to converge a small number of time step of the input vector to the correct value.

Input	JISE with ADM	LMS
Impulse (SDoF)	1.07 sec	4.15 sec
Sweep (SDoF)	1.31 sec	4.32 min
Random (SDoF)	1.05 sec	4.51 min
Impulse (MDoF)	1.15 sec	6.08 sec
Sweep (MDoF)	1.31 sec	11.4 min
Random (MDoF)	1.02 sec	50.9 sec

Table 6.1: Computation times for the JISE with ADM and the LMS algorithm

6.4.2. Real-Time Applicability

In for example control applications it is useful to apply force identification in real-time. To do so, a trustworthy solution must be found within limited computation time.

As the Joint Input-State Estimator finds its solution over one recursion it is fundamentally suitable for real-time applications. Conveniently, it uses a state space system description often used in control applications. However, as stated earlier, when only acceleration measurements are used the JISE is not stable and thus does not produce a trustworthy solution in this case. Extending the algorithm with Artificial Displacement Measurements to stabilize the filter is risky to do for real-time application. The stability of the filter depends on the parameter to be set, the covariance matrix of the displacement. As the optimal setting of this parameter is dependent on the input itself this method is less suitable for real-time applications. If, and only if, prior knowledge on the input signal that can guarantee stability is available, this method can be applied in real-time. It may be worth researching a possible relation between the optimal setting of the parameter and the measured acceleration. If such a relation exists it can be implemented in the filter resulting in a method that can trustworthy find a solution for force identification in real-time. Also, true displacement measurements would allow for real-time application of the JISE.

The Least Mean Square Adaptive Algorithm is inherently not suitable for force identification in real-time. Whereas the JISE finds its solution over one recursion, the LMS algorithm requires multiple iterative recursions to obtain a trustworthy solution. This property lies at the fundament of the method and makes it unsuitable for real-time applications.

7
Experimental Evaluation

The previous chapter tested a number of force identification methods by feeding them with simulated data. The performance was analyzed and two method were selected. The Joint Input-State Estimator with Artificial Displacement Measurement (JISE with ADM) and the Least Mean Square adaptive algorithm (LMS) will be exposed to data obtained by experiment. Both methods will reconstruct the forces exerted by a stepper motor on an aluminium structure. This chapter will present the method and results of this experiment.

The goal of this chapter is to test the performance of both methods in a set-up close to what can be expected from a real application in noise and vibration prediction in the automotive industry. In the case of simulated measurements, measurement uncertainty was assumed to be Gaussian white noise and system uncertainty was assumed to be not present. In the case of a real application measurement errors may behave differently and system uncertainty will not be zero in practice.

7.1. Set-up

A aluminium structure is used to which four accelerometers are attached at locations throughout the structure (figure 7.1). All four accelerometers measure the acceleration in three orthogonal directions resulting in twelve response signals. The structure is suspended using elastic bands to create free-free boundary conditions. An impact hammer was used to excite the structure in three orthogonal directions. The measured force and response are used to obtain a description of the system dynamics. A stepper motor attached with a bolted connection to an end of the structure is used as an operational excitation. A sampling frequency of 16384 Hz is used. To evaluate and compare the forces found by both methods one of the twelve response signals will be used as a validation signal and will be not used in force identification. A response prediction obtained with the identified force and system description will be compared to the validation signal to obtain a indication of the performance of the algorithm.

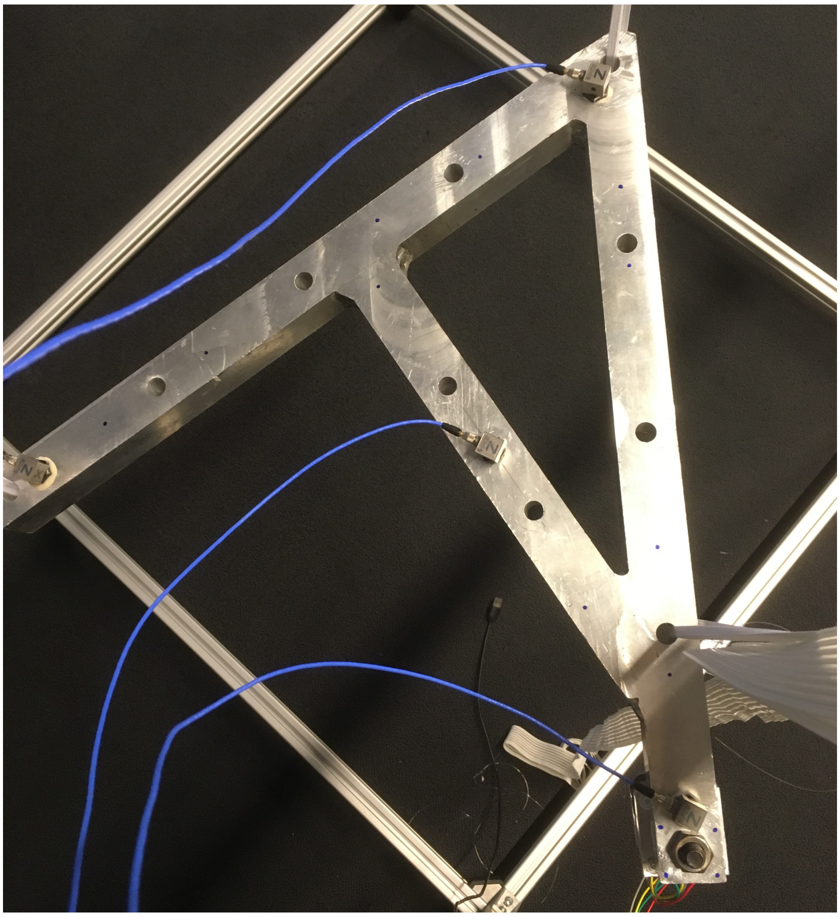


Figure 7.1: Set-up of measurement

7.2. System Description

This section will show the how the system descriptions were obtained. The LMS algorithm uses the Impulse Response Function (IRF) which is obtained using a H_v estimator shown in section 7.2.1. The JISE with ADM uses state space for which the matrices are obtained using Subspace Identification shown in section 7.2.2.

7.2.1. Impulse Response Functions

To obtain the IRFs the force measured by the impact hammer and the responses measured by the accelerometers are used to determine the Frequency Response Functions (FRFs). The acceleration measurement is transformed to the frequency domain using the Fast Fourier Transform with a Hann window. The FRFs are then obtained using an H_v estimator [54]. This estimator assumes noise on the input and output and minimizes it.

$$H_v = \frac{\frac{1}{q}\sum_1^q G_{xf}}{\left|\frac{1}{q}\sum_1^q G_{xf}\right|}\sqrt{\frac{\frac{1}{q}\sum_1^q G_{xx}}{\frac{1}{q}\sum_1^q G_{ff}}} \tag{7.1}$$

In this equation the cross power spectrum (G_{xf}) is defined as the complex conjugate of the input and the output. G_{xx} and G_{ff} are the autopower spectrum and defined as the complex conjugate of the input and output and the spectrum of the input and output itself, respectively. When more impacts per input locations are used the spectra are averaged over all measurement, q is the number of impacts per input location. The IRFs can be determined by transforming the FRFs back to the time domain using the Inverse Fast Fourier Transform with adequate zero-padding. It may seem inefficient to transform time domain data to the frequency domain after which to transform it back to the time domain. However, it is necessary as the division in the H_v estimator in the frequency domain is a deconvolution in the time domain. As explained in section 3.1, deconvolution is not straightforward.

7.2.2. State Space System

The state space matrices are obtained with an identification technique referred to as Subspace Identification (SI) . This method is able to find state space matrices up to a similarity transformation from input and output data. When the input and output data is rearranged in structured block Hankel matrices it can be used to find subspaces that are related to the system matrices. Examples of such subspaces are the column space of the observability matrix and the row space of the state sequence of a Kalman filter. This non-iterative system identification method uses three linear algebra operations: RQ factorization, the Singular Value Decomposition and the solution of a linear least square problem. For a detailed description of this method the reader is referred to the book published by Verhaegen and Verdult [66]. These authors have also made available the LTI System Identification Toolbox that implements the methods described in their book. This toolbox has been used in this research.

7.3. Pseudo-forces

Operational excitation of the structure is done via a stepper motor which is connected to the structure with a bolt. The location of the excitation is therefore not a single point but rather two area's of contact where the connection is made. For practical reasons, modeling through this contact area is not possible as measurements at such an area are not possible. Therefore an alternative to the physical input location is needed.

A method was introduced by Janssens and Verheij [24] referred to as the pseudo-forces method. It's working principle is that a set of forces can be found that cancels out the response by the excitation. If these forces are then applied to the structure in the opposite direction they should result in an equal response. These forces, called pseudo-forces should now represent the operational excitation. If the location of the pseudo-forces is conveniently chosen at the edge of the structure they can be easily accessed with an impact hammer and a system description from these locations to the response positions can be obtained.

For this research the structure is excited by an impact hammer on three locations around the stepper motor in three orthogonal directions to describe the operational excitation as complete as possible.

The measured impacts and their responses result in a system with three inputs and eleven outputs (the twelfth output is used as a validation channel). The Frequency Response Functions related to this system are shown in figure 7.2. In this figure the blue lines correspond to the FRFs directly from the measured data, determined with equation 7.1. The red lines show the FRFs of the state space system obtained through Subspace Identification. Clearly the FRFs do not match. The frequencies of the resonances and anti-resonances are the same however the state space appears to show dynamics at a higher amplitude. The Subspace Identification method thus does not produce a correct state space system that matches the true dynamics of the system. This can have various causes, however the reason Subspace Identification fails is unknown to the author at the time of writing this thesis. Attempts to improve to the solution include varying the number of states used and adjusting damping. From these attempts can be concluded that the number of states it not the cause of the problems experienced. The information added when using more states has converged and thus adding more states would not results in a better defined system. It is suspected that damping may be the cause of the state space system incorrectly describing the dynamics as this is a common difficulty in system identification. It requires more research to find out if this is the case. It is found that the stability of the system is highly sensitive to the damping values in the state space matrices.

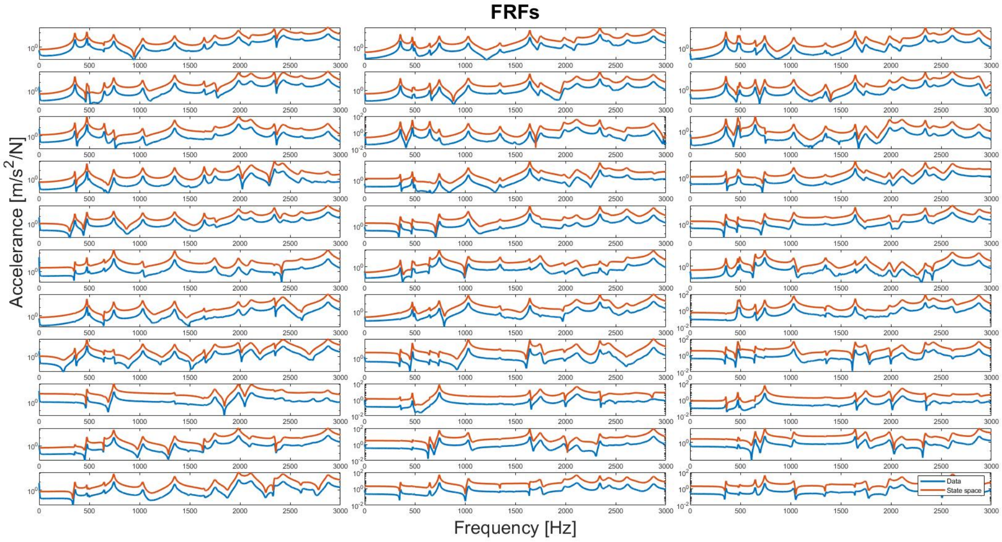


Figure 7.2: Frequency Response Functions of the measurement and the state space system where the columns represent pseudo-forces and the rows response channels

As the offset is merely a linear discrepancy in dynamics it can be solved by adjusting the $\boldsymbol{B}$ and $\boldsymbol{J}$ matrix accordingly:

$$\boldsymbol{B}_{new} = \boldsymbol{B}\begin{bmatrix} c_1 & c_2 & c_3 \end{bmatrix} \tag{7.2}$$

$$\boldsymbol{J}_{new} = \boldsymbol{J}\begin{bmatrix} c_1 & c_2 & c_3 \end{bmatrix} \tag{7.3}$$

where c_1, c_2 and c_3 are scalars that shift the FRFs to the correct amplitude. The shifted FRFs are shown in figure 7.3. It can be seen that the adjusted $\boldsymbol{B}$ and $\boldsymbol{D}$ matrix effectively correct for the linear offset found in the results of Subspace Identification. The state space system now correctly predicts the input-output relation for the measurements of the impact hammer and its responses measured by the accelerometers. These state space matrices are now to be used in the Joint Input-State Estimator for force identification.

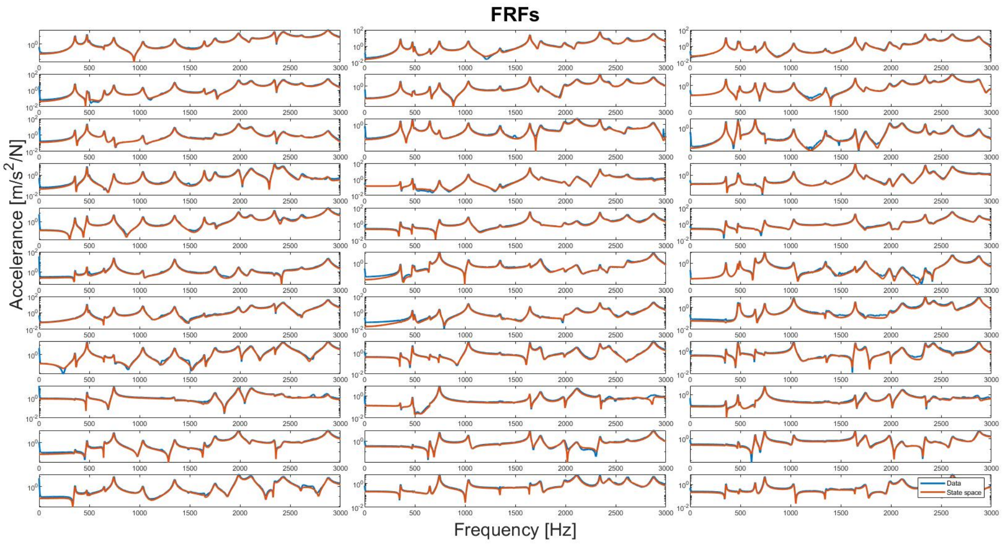


Figure 7.3: Linearly corrected Frequency Response Functions of the measurement and the state space system where the columns represent pseudo-forces and the rows response channels

7.3.1. Joint Input-state Estimator with Artificial Displacement Measurement

The JISE with ADM has been used to reconstruct the forces exerted by the operational excitation in terms of pseudo-forces. The twelfth response channels has not been used for the identification of the force but for verification of the identified forces. Figure 7.4 shows the measured signal of the verification channels (top left), zoomed in (top right) and its frequencies (bottom) and the response of the system when numerically excited with the identified forces. It shows that the frequency content of the response is captured by the JISE with ADM, however the verification channel does not completely match. It appears that the damping in the two verification channels are different. This may confirm the earlier suspected statement that damping is incorrectly captured in the state space system. In the previous chapter it was found that the JISE with ADM does not perform perfectly when a wide range of frequencies are present in the input which is the case in this measurement. It is to be expected that both the troublesome state space system and the wide range of frequencies play at least some role in the result obtained by the JISE with ADM. Another explanation for the result not to completely match the verification is that the Subspace Identification method only find the system up to a similarity transformation. This implies that the identified system produces the correct input-output relation however the states of the system do not represent the true physical states. Therefore, extending the $\boldsymbol{C}$ and $\boldsymbol{D}$ matrix to add Artificial Displacement Measurement may not restrict the true physical displacement. The Joint Input-State Estimator has been successfully implemented in combination with Subspace Identification by Azam [7] of which the results are said to be promising. It also proposed an approach to project the obtained states of the state space system back to the physical states. It is worth researching whether this projection of states would also resolve the issues experienced with applying the JISE with ADM in combination with Subspace Identification.

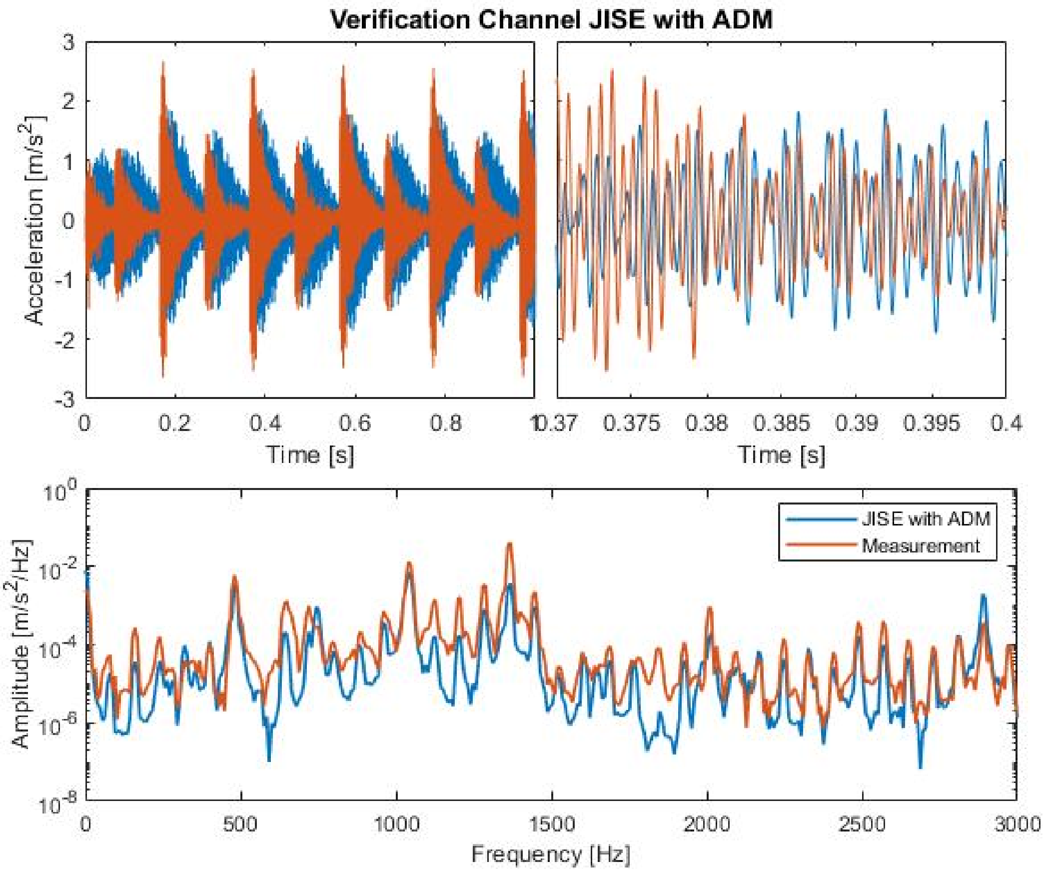


Figure 7.4: Time and frequency domain data of verification channel obtained with JISE with ADM

7.3.2. Least Mean Square Adaptive Algorithm

The same operational measurement has been used to identify the pseudo-forces with the LMS algorithm. Again, the twelfth channels is used as a channel to verify the obtained forces. Figure 7.5 shows the measured and the predicted response of the verification channel when the LMS algorithm is used. It shows that the forces identified by the LMS algorithm correctly predict the response of the verification channels. It can therefore be concluded that the found pseudo-forces represent the operational excitation by the stepper motor and that the performance of the LMS algorithm in this scenario is satisfactory.

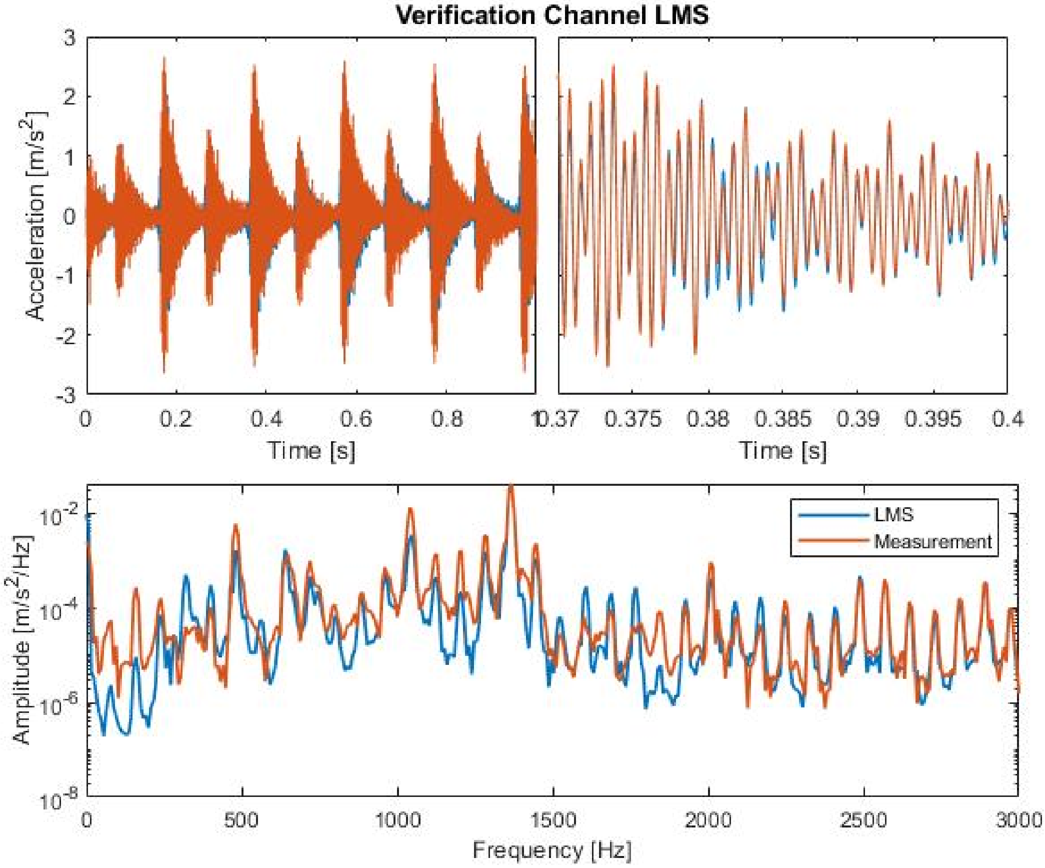


Figure 7.5: Time and frequency domain data of verification channel obtained with LMS

7.3.3. LMS vs JISE with ADM

To observe the relative performance of both algorithms the identified pseudo-forces and their frequency content are plotted in figure 7.6. Surprisingly the forces obtained by the JISE with ADM and the LMS algorithm are comparable which was not expected when observing the results in terms of the verification channel in the previous two sections. This results suggests that the results obtained by the JISE with ADM are correct. A possibility is that the incorrectness of the obtained forces by the JISE with ADM are not visible in these plot and they lie in the phase of the signal. As no problem with these phase of the signal have been observed in the numerical test cases the phase must not be correctly captured by the state space system. It requires more research to find out if this is the case.

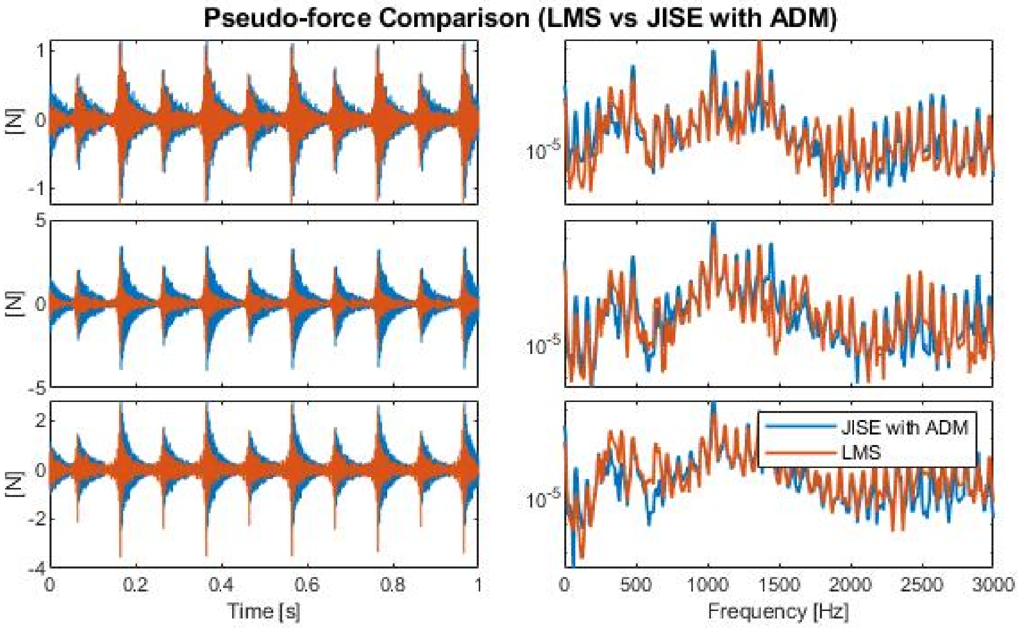


Figure 7.6: Comparison of the identified pseudo-forces with the JISE with ADM and the LMS algorithm

To compare the result of both methods the Root Mean Square Error (RMSE) of the verification channel is used which can be determined with the following equation:

$$RMSE = \left(\frac{\sum_{i=1}^{N} \left(\boldsymbol{d}(t) - \hat{\boldsymbol{u}}(t) \right)^2}{N} \right)^{1/2} \tag{7.4}$$

where $\boldsymbol{d}(t)$ is the measured data of the verification channel, $\hat{\boldsymbol{u}}(t)$ is the response of the system on the identified forces and N the number of data points. The RMSE for both method is given in table 7.1. The RMSE shows that the result obtained with the LMS algorithm is significantly better than the results obtained with the JISE with ADM.

Method	RMSE
JISE with ADM	0.7301
LMS	0.1777

Table 7.1: Root Mean Square Error of the verification channel for the pseudo forces obtain with the JISE with ADM and the LMS algorithm

It can be concluded that the LMS algorithm reliably reconstructs pseudo-forces for Transfer Path Analysis. The results obtained with the JISE with ADM are less reliable. It can therefore be concluded that the LMS algorithm outperforms the JISE with ADM in terms of accuracy. It is expected that an incorrect state space system plays at least some role in this result.

7.4. Virtual Point Forces

In the previous section the identified force was described using three orthogonal forces. To give a full description of forces through one point, for example in coupling two structures, three forces and three moment are necessary. This can be achieved by using the Virtual Point Transformation (VPT) [64]. With this method multiple translational measurements on different input and output locations can be transformed to one single point resulting in three forces and three moments. These three forces and moment can fully describe the Virtual Point (VP). For the VPT to be applied it is assumed that the structure is rigid in the area where the transformation takes place. Transformation can be done on the input, output or both. For this research, transformation on the input is considered only. The blue dots in figure 7.7 show the hammer impact locations around the the connection between the aluminium.

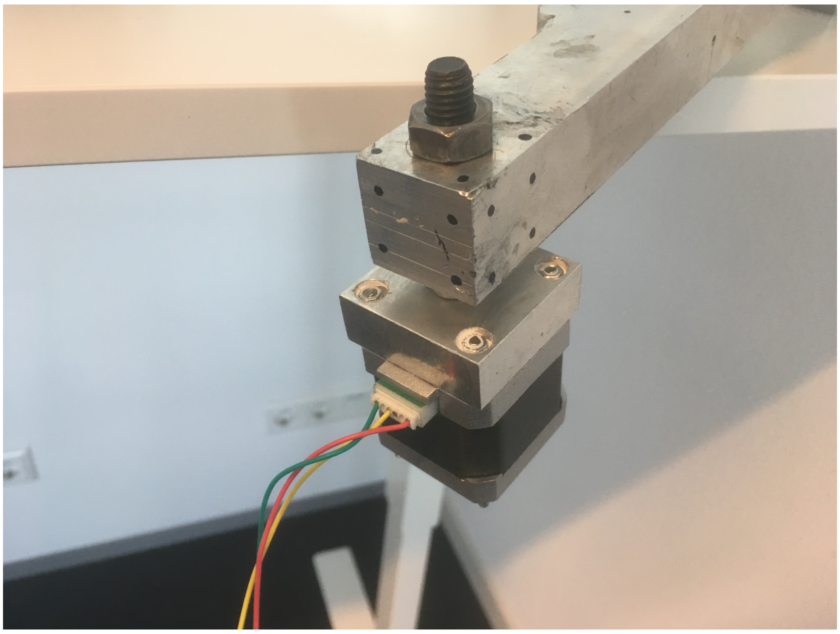


Figure 7.7: Impacts locations on the aluminium structure around the stepper motor

Twelve input locations are used to describe the input in the VP. After transformation a system of six inputs and eleven output remains. The FRFs of the measured data (blue) and the state space system obtained through Subspace Identification (red) are shown in figure 7.8. As with the pseudo-forces, the same problem occurs: the state space system holds the dynamic information however at a different amplitude. Noteworthy is that whereas for the forces (first three rows) the amplitude has increased for the state space system, it has drastically decreased for the moment (last three rows). As this offset is linear it can be solved by linearly scaling on the input side of the state space system:

$$\boldsymbol{B}_{new} = \boldsymbol{B}\begin{bmatrix} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 \end{bmatrix} \tag{7.5}$$

$$\boldsymbol{J}_{new} = \boldsymbol{J}\begin{bmatrix} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 \end{bmatrix} \tag{7.6}$$

where c_1, c_2, c_3, c_4, c_5 and c_6 are scalars that shift the FRFs to the correct amplitude. Figure 7.9 shows that the corrected FRFs match the FRFs of the measured data. It does appear that the state space system does not capture all anti-resonances, especially in the low frequency range.

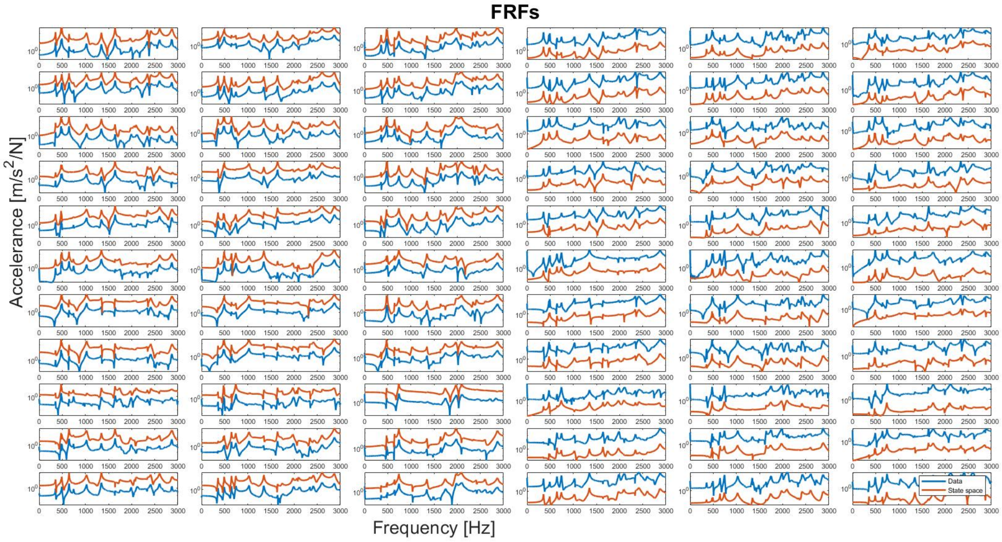


Figure 7.8: Frequency Response Functions of the measurement and the state space system where the columns represent virtual point forces and the rows response channels

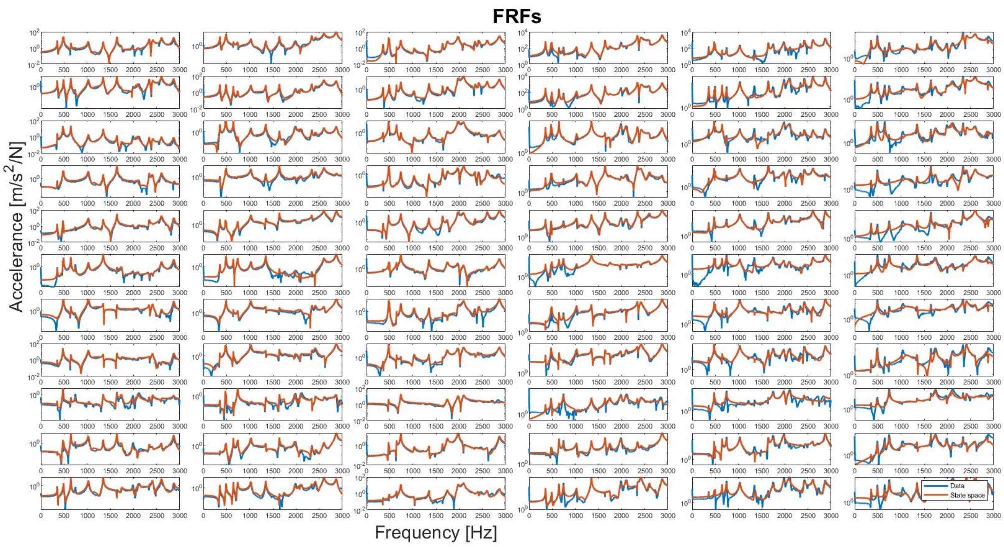


Figure 7.9: Linearly corrected Frequency Response Functions of the measurement and the state space system where the columns represent virtual point forces and the rows response channels

7.4.1. Virtual Point Transformation for Impulse Response Functions

The VPT for FRFs is described in [64] and the conversion to the time domain is straightforward as we are solely dealing with a spacial transformation. To obtain the IRFs for which the inputs are transformed to the VP it needs to be multiplied with transformation matrix $\boldsymbol{T}$. This matrix results from the locations and directions of the impact locations and the location of the VP. A detailed description of how transformation matrix $\boldsymbol{T}$ is obtained is given in appendix E.

$$\boldsymbol{H}_{VP} = \boldsymbol{H}\boldsymbol{T}^T \tag{7.7}$$

7.4.2. Virtual Point Transformation for State Space Systems

The state space matrices can be adjusted to that they are compatible with the VPT. When transformation takes place on the input matrices $\boldsymbol{B}$ and $\boldsymbol{J}$ concerning the input are transformed. The impact forces relate to the virtual point forces and moment according to the following equation:

$$\boldsymbol{f} = \boldsymbol{T}^T \boldsymbol{m} \tag{7.8}$$

where $\boldsymbol{f}$ are the impact forces and $\boldsymbol{m}$ the virtual point forces and moments. This can be entered in the state space equations to obtain the equations that describe the VPT on the input in state space:

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{T}^T \boldsymbol{m} \tag{7.9}$$

$$\boldsymbol{d}(k) = \boldsymbol{G}\boldsymbol{x} + \boldsymbol{J}\boldsymbol{T}^T \boldsymbol{m} \tag{7.10}$$

The VPT can be extended to the case where transformation takes place on both the input and the output. The relation between measured output and the virtual point transformed output can be written as:

$$\boldsymbol{q} = \boldsymbol{T}\boldsymbol{d} \tag{7.11}$$

Entering this in the state space equations gives the equations that describe the full VPT in state space:

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{T}^T \boldsymbol{m} \tag{7.12}$$

$$\boldsymbol{q}(k) = \boldsymbol{T}\boldsymbol{G}\boldsymbol{x} + \boldsymbol{T}\boldsymbol{J}\boldsymbol{T}^T \boldsymbol{m} \tag{7.13}$$

The description of the full VPT in state space will not be used further in this thesis but has been added for completeness.

7.4.3. Joint Input-state Estimator with Artificial Displacement Measurement

The same verification channels as used in the previous sections is used to verify the virtual point forces obtained with the JISE with ADM. It can be seen that the virtual point forces found by the JISE with ADM predict the measurement by the verification channel somewhat correctly. When looking at the time domain data it can be seen that the signals show resemblance however the amplitude is not always correct. The same can be seen in the frequency domain data where the most frequency peaks are captured by the JISE with ADM however some of them at different amplitude. Also multiple anti-resonances where not captured by the filter. This occurs at the same frequencies as where anti-resonances were not present in the state space system (see figure 7.9). It can thus be concluded that this behavior originates from the state space system and not from the force identification method. Results have improved compared to the pseudo-forces found by the JISE with ADM so it can be concluded that the addition of moments by the VPT improves the result. Inconsistencies in these results, however less present compared to the results obtained with pseudo-forces, can be attributed to the same causes that are explained in section 7.3.1. The results obtained by the JISE with ADM show that its produces reliable virtual point forces to be used in TPA.

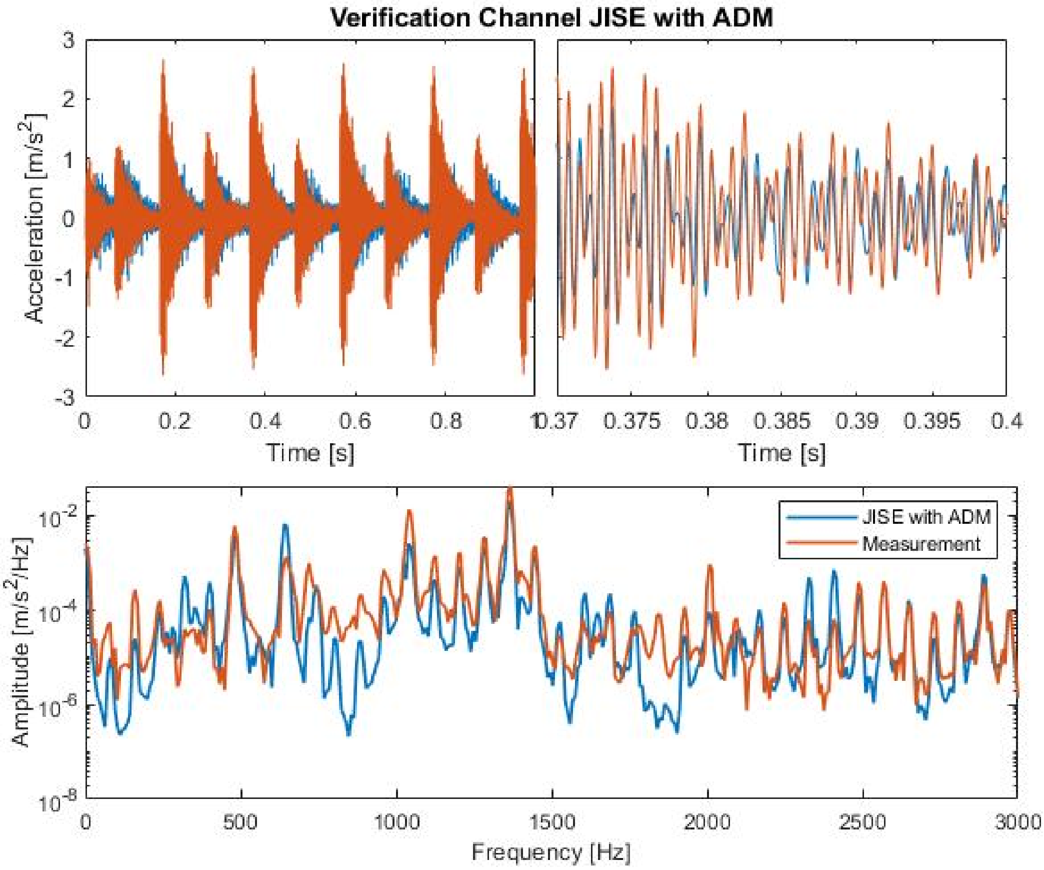


Figure 7.10: Time and frequency domain data of verification channel obtained with JISE with ADM

7.4.4. Least Mean Square Adaptive Algorithm

The virtual point forces found by the LMS algorithm predict the response of the verification channels correctly. The time domain signal closely matches the verification signal and the peaks of the frequency domain data are captured by the LMS algorithm. A few number of peaks in the low frequency area are less present which can be expected as low frequency behavior is less visible in acceleration measurements. It can be concluded that the LMS algorithm is suitable for TPA in combination with the VPT.

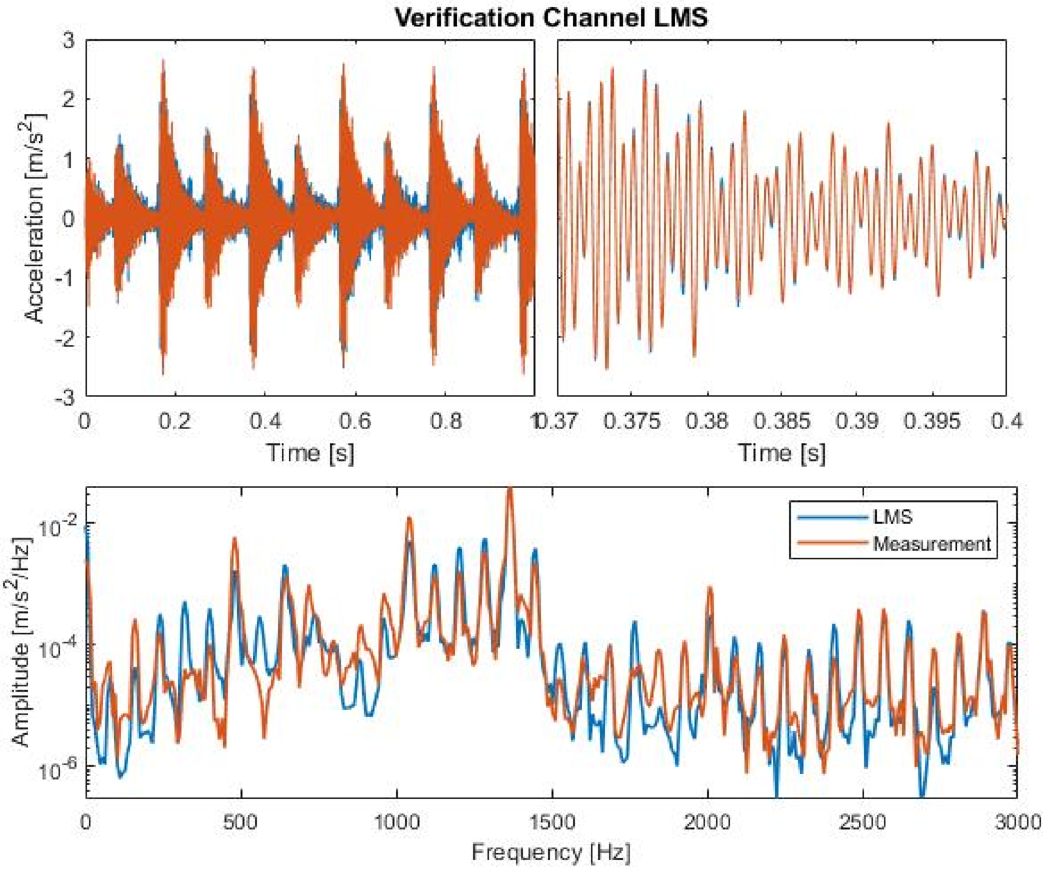


Figure 7.11: Time and frequency domain data of verification channel obtained with LMS

7.4.5. LMS vs JISE with ADM

As concluded the LMS algorithm is able to correctly reconstruct the forces and moment acting on the virtual point that represent the operational excitation by the stepper motor. When these results are compared with the results obtained by the JISE with ADM it can be seen that results are comparable from which can be concluded that the obtained results are reliable.

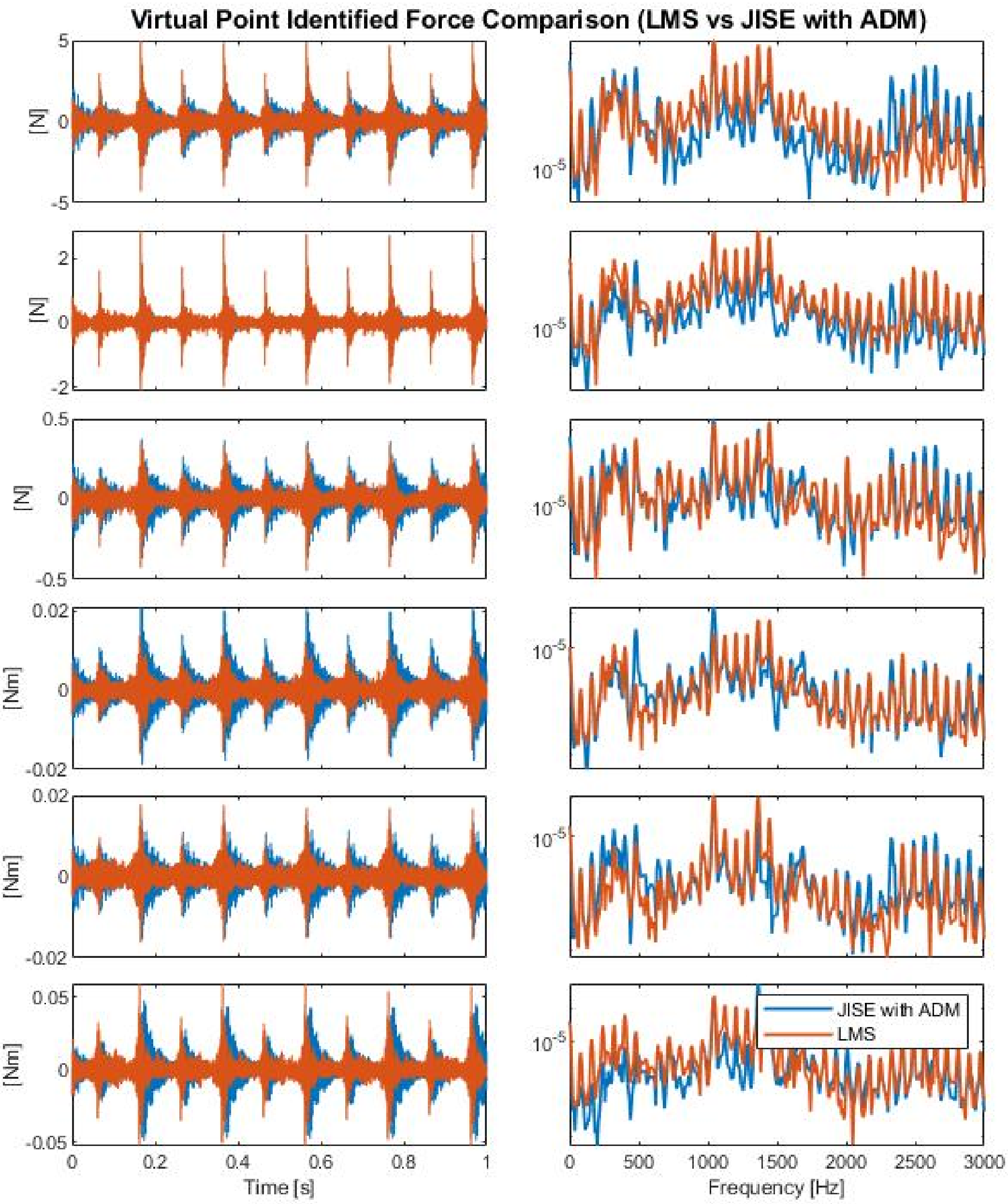


Figure 7.12: Comparison of identified forces and moments acting on the virtual point

For a quantitative comparison between both methods the RMSE of the verification channel is used, for which the results are shown in table 7.2. This confirms the conclusion that the VPT has improved the result found by the JISE with ADM. Results for the LMS algorithm have also slightly improved after the VPT. Both methods are thus suitable to be applied for TPA when the VPT is used. Even though the results for the JISE with ADM have improved with the addition of moment in the virtual point, the LMS algorithm outperforms the JISE with ADM in terms of accuracy.

Method	RMSE
JISE with ADM	0.3938
LMS	0.1585

Table 7.2: Root Mean Square Error of the verification channel for the virtual point forces obtain with the JISE with ADM and the LMS algorithm

8 Conslusions

In this chapter the conclusions of this research are summarized.

- Literature study shows that from the available methods the Dual Kalman Filter, Joint-Input State Estimator and the Least Mean Square Adaptive Algorithm are most suitable for force identification when applied for Transfer Path Analysis.
- The Dual Kalman Filter and the Joint Input-State Estimator are shown to show unstable behavior when only acceleration measurements are used. This is due to the fact that a state space system which only observes acceleration will always have a marginally stable zero corresponding to static excitations. This marginally stable zero determines the stability of the inverse of the system. In physical terms: static excitation cannot be observed by acceleration measurements.
- The Dual Kalman Filter and the Joint Input-State Estimator can be stabilized while only using acceleration measurements by using Artificial Displacement Measurements. This methods adds an artificial displacement measurement of zero to every acceleration measurement. To comply with the acceleration measurement, freedom is given around the displacement measurement by means of a covariance matrix. This matrix can be used to control the stability of the identification method.
- It is mentioned in literature that the Least Mean Square Adaptive algorithm is only able to reliably reconstruct forces from time index O where O is the size of the Impulse Response Function. A simple extension to the input vector with zeros the size of O will resolve this issue. This method will make the Least Mean Square Adaptive algorithm able to reliably reconstruct the complete input signal.
- The Least Mean Square Adaptive algorithm yields satisfactory results when applied to a simulated single and multiple degree of freedom system as well as to measurement taken from an aluminium structure. It is able to reliably reconstruct pseudo-forces and virtual point forces.
- A downside of the Least Mean Square Adaptive algorithm is its computation time. Various variable step size methods for the Least Mean Square Adaptive algorithm have been implemented in an attempt to improve its speed of convergence. It has been found that none of these methods improves the speed of convergence of this algorithm. The stability bound for the Least Mean Square Adaptive algorithm for force identification given in literature provides the largest stable step size possible and varying it will only decrease convergence speed or results in instability.
- Another downside of the Least Mean Square Adaptive algorithm is the fact that it is blind to static and quasi-static inputs when only acceleration measurement are used as an estimation of velocity and displacement is not incorporated in the algorithm.
- The Joint Input-State Estimator with Artificial Displacement Measurements is able to reliably reconstruct the input forces on a simulated single and multiple degree of freedom system. When applied to measurements on a aluminium structure the pseudo-forces found are less reliable. This is at least for some part due the an incorrect state space system. Virtual point forces are reliably reconstructed when using measurement on an aluminium structure.

- Even though the Joint Input-State Estimator with Artificial Displacement Measurements uses an estimation of velocity and position in the algorithm, results for static and quasi-static inputs are not without fail reliable. The results are sensitive to covariance matrix that determines the freedom of the displacement measurement.
- The results obtained by the Joint Input-State Estimator with Artificial Displacement Measurements for measurements on a aluminium structure depend heavily on the state space system found through Subspace Identification.
- The Least Mean Square Adaptive algorithm outperforms the Joint Input-State Estimator with Artificial Displacement Measurements in terms of accuracy.
- The computation time of the Least Mean Square Algorithm is significantly larger than the computation time of the Joint Input-State Estimator with Artificial Displacement Measurements
- The Least Mean Square Adaptive Algorithm is not suitable for real-time applications
- The Joint Input-State Estimator with Artificial Displacement Measurements is not unconditionally stable for real-time applications as the optimal setting for its stability parameters depends on the input itself. The Joint Input-State estimator with true displacement measurements would allow for real-time applications.

9 Recommendations

This chapter present recommendations for further research.

- The results obtained by the Joint Input-State Estimator with Artificial Displacement Measurements depend heavily on the state space system. As the matrices of the system have been adjusted manually in this research to fit the data, an error in the state space system obtained through Subspace Identification may be the cause of the lesser performance of the Joint Input-State Estimator with Artificial Displacement Measurements. Further research is needed to show if this is the case.
- As stated by earlier research [7] the Joint Input-State Estimator (without Artificial Displacement Measurements) is effective in combination with Subspace Identification. A reason for it to be less successful in this research may lie in the fact that Subspace Identification can only obtain a state space system up to a similarity transformation. This implies that the states in the system do not represent physical states. In the above mentioned research a method is proposed that transforms the generic states of the system to a physical state space system. More research on this will show the effectiveness of this method in combination with the Joint Input-State Estimator with Artificial Displacement Measurements.
- Incorporating displacement measurements would drastically improve the performance of the Joint Input-State Estimator. Research on the application of displacement measurements with the use of strain gauges, lasers or computer vision would show the effectiveness of this method.
- Successful implementation of the Joint Input-State Estimator with displacement measurements would allow for real-time force identification.

A

Parameters of the Newmark-Beta Method

The appendix shows the parameters given in the solution found by the Newmark-Beta Method.

$$\boldsymbol{A}_0 = \hat{\boldsymbol{K}}^{-1} \tag{A.1}$$

$$\boldsymbol{A}_d = \hat{\boldsymbol{K}}^{-1}\left(\frac{1}{\beta\Delta t^2}\boldsymbol{M} + \frac{\gamma}{\beta\Delta t}\boldsymbol{C}\right) \tag{A.2}$$

$$\boldsymbol{A}_v = \hat{\boldsymbol{K}}^{-1}\left(\frac{1}{\beta\Delta t}\boldsymbol{M} + \left(\frac{\gamma}{\beta} - 1\right)\boldsymbol{C}\right) \tag{A.3}$$

$$\boldsymbol{A}_a = \hat{\boldsymbol{K}}^{-1}\left(\left(\frac{1}{2\beta} - 1\right)\boldsymbol{M} + \frac{\Delta t}{2}\left(\frac{\gamma}{\beta} - 2\right)\boldsymbol{C}\right) \tag{A.4}$$

$$\boldsymbol{B}_0 = \frac{\gamma}{\beta\Delta t}\hat{\boldsymbol{K}}^{-1} \tag{A.5}$$

$$\boldsymbol{B}_d = \frac{-\gamma}{\beta\Delta t}\hat{\boldsymbol{K}}^{-1}\boldsymbol{K} \tag{A.6}$$

$$\boldsymbol{B}_v = \frac{\gamma}{\beta\Delta t}\hat{\boldsymbol{K}}^{-1}\left(\left(\frac{\beta\Delta t}{\gamma} - \Delta t\right)\boldsymbol{K} + \frac{1}{\gamma\Delta t}\boldsymbol{M}\right) \tag{A.7}$$

$$\boldsymbol{B}_a = \frac{\gamma}{\beta\Delta t}\hat{\boldsymbol{K}}^{-1}\left(\left(\frac{\beta\Delta t^2}{\gamma} - \frac{\Delta t^2}{2}\right)\boldsymbol{K} + \left(\frac{1}{\gamma} - 1\right)\boldsymbol{M}\right) \tag{A.8}$$

$$\boldsymbol{C}_0 = \frac{1}{\beta\Delta t^2}\hat{\boldsymbol{K}}^{-1} \tag{A.9}$$

$$\boldsymbol{C}_d = \frac{-1}{\beta\Delta t^2}\hat{\boldsymbol{K}}^{-1}\boldsymbol{K} \tag{A.10}$$

$$\boldsymbol{C}_v = \frac{-1}{\beta\Delta t^2}\hat{\boldsymbol{K}}(\boldsymbol{C} + \Delta t\boldsymbol{K}) \tag{A.11}$$

$$\boldsymbol{C}_a = \frac{1}{\beta\Delta t^2}\hat{\boldsymbol{K}}^{-1}\left(\left(\gamma - 1\right)\Delta t\boldsymbol{C} - \beta\Delta t^2\left(\frac{1}{2\beta} - 1\right)\boldsymbol{K}\right) \tag{A.12}$$

with:

$$\hat{\boldsymbol{K}} = \boldsymbol{K} + \frac{1}{\beta\Delta t^2}\boldsymbol{M} + \frac{\gamma}{\beta\Delta t}\boldsymbol{C} \tag{A.13}$$

B
Norms

This appendix shows the norms used by the different regularization methods.

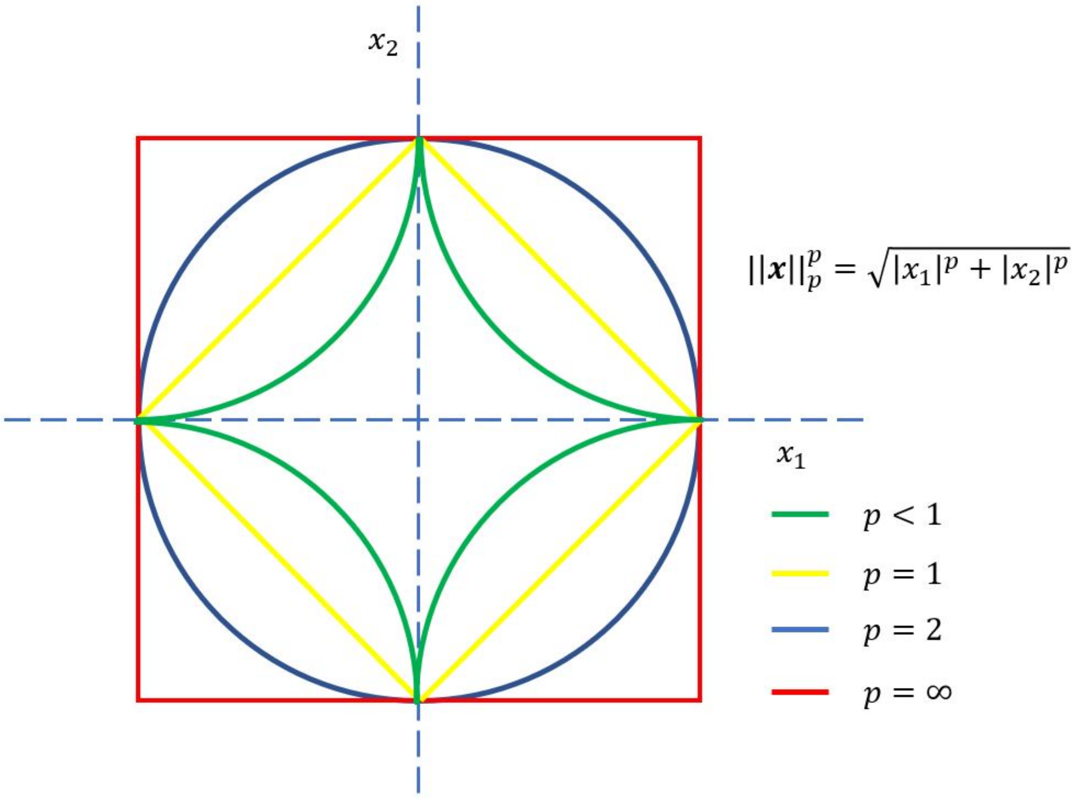


Figure B.1: Norms

C

Example of Tikhonov Regularization in 2D

This section illustrates the concept of Tikhonov regularization with an example. To visualize this example regularization has been applied on only two dimensions, however the expansion to higher dimensions is straightforward. The system of equations to be solved is:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{C.1}$$

where:

$$\boldsymbol{A} = \begin{bmatrix} 6 & 5 \\ 12 & 10 \end{bmatrix} \tag{C.2}$$

the to be found solution:

$$\boldsymbol{x} = \begin{bmatrix} 6 \\ 2 \end{bmatrix} \tag{C.3}$$

resulting in:

$$\boldsymbol{b} = \begin{bmatrix} 46 \\ 92 \end{bmatrix} \tag{C.4}$$

As clearly visible, the columns of matrix $\boldsymbol{A}$ are linearly dependent making the matrix rank deficient. A unique solution for this problem does not exist and the solution is dependent on either x_1 or x_2. This is shown by the blue line in figure C.1. The 2-norm (or Euclidian distance) of x_1 and x_2 is shown by the purple line. When the regularization parameter λ is chosen correctly the regularized solution lies on the intersection of the 2-norm of x_1 and x_2 and the line (or (hyper)plane in more dimensions) of all possible solutions.

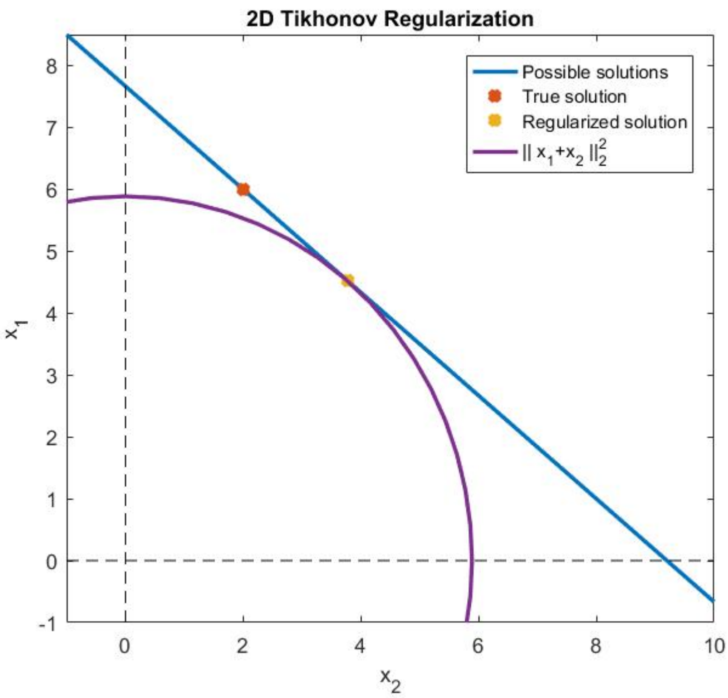


Figure C.1: Visualization of Tikhonov regularization in 2D

D

Derivation of the Kalman gain

The goal of the Kalman filter is to minimize the sum of the covariance between the true state and the estimated state. In this section a predicted value is shown with a bar ($\bar{\mathbf{x}}$), an updated value with a hat ($\hat{\mathbf{x}}$) and a true value without an addition ($\mathbf{x}$). The covariance is defined as:

$$\mathbf{P} = cov\left[\mathbf{x} - \hat{\mathbf{x}}\right] \tag{D.1}$$

From the measurement step of the Kalman Filter algorithm the estimated state (equation 5.6) can be plugged in:

$$\mathbf{P}(k) = cov\left[\mathbf{x}(k) - \left(\bar{\mathbf{x}}(k) + \mathbf{K}(k)\left(\mathbf{G}\mathbf{x}(k) + \mathbf{v}(k) - \mathbf{G}\bar{\mathbf{x}}(k)\right)\right)\right] \tag{D.2}$$

In this equation $\mathbf{y}$ is written in terms of the observation matrix $\boldsymbol{G}$, the true state $\mathbf{x}(k)$ and the noise vector $\mathbf{v}(k)$:

$$\mathbf{y}(k) = \mathbf{G}\mathbf{x}(k) + \mathbf{v}(k) \tag{D.3}$$

Written out, this becomes:

$$\mathbf{P}(k) = cov\left[\left(\mathbf{I} - \mathbf{K}(k)\mathbf{G}\right)\left(\mathbf{x}(k) - \bar{\mathbf{x}}(k)\right) - \mathbf{K}\mathbf{v}(k)\right] \tag{D.4}$$

$$\mathbf{P}(k) = cov\left[\left(\mathbf{I} - \mathbf{K}(k)\mathbf{G}\right)\left(\mathbf{x}(k) - \bar{\mathbf{x}}(k)\right)\right] - cov\left[\mathbf{K}(k)\mathbf{v}(k)\right] \tag{D.5}$$

Because:

$$cov\left[\mathbf{A}\mathbf{B}\right] = \mathbf{A}cov\left[\mathbf{B}\right]\mathbf{A}^T \tag{D.6}$$

equation D.5 can be written as

$$\mathbf{P}(k) = \left(\mathbf{I} - \mathbf{K}(k)\mathbf{G}\right)cov\left[\mathbf{x}(k) - \bar{\mathbf{x}}(k)\right]\left(\mathbf{I} - \mathbf{K}(k)\mathbf{G}\right)^T + \mathbf{K}(k)cov\left[\mathbf{v}(k)\right]\mathbf{K}(k)^T \tag{D.7}$$

The covariance of the noise is defined as $\mathbf{R}$

$$\mathbf{R} = cov\left[\mathbf{v}(k)\right] \tag{D.8}$$

and the covariance between the true state and the predicted state $\bar{\mathbf{P}}(k)$ as

$$\bar{\mathbf{P}}(k) = cov\left[\mathbf{x}(k) - \bar{\mathbf{x}}(k)\right] \tag{D.9}$$

Using this in equation D.7 gives:

$$\mathbf{P}(k) = \left(\mathbf{I} - \mathbf{K}(k)\mathbf{G}\right)\bar{\mathbf{P}}(k)\left(\mathbf{I} - \mathbf{K}(k)\mathbf{G}\right)^T + \mathbf{K}(k)\mathbf{R}\mathbf{K}(k)^T \tag{D.10}$$

This can be written out as:

$$\mathbf{P}(k) = \bar{\mathbf{P}}(k) - \bar{\mathbf{P}}(k)\mathbf{G}^T\mathbf{K}(k)^T - \mathbf{K}(k)\mathbf{G}\bar{\mathbf{P}}(k) + \mathbf{K}(k)\mathbf{S}(k)\mathbf{K}(k)^T \tag{D.11}$$

where $\mathbf{S}(k)$ is defined as:

$$\mathbf{S}(k) = \mathbf{G}\bar{\mathbf{P}}(k)\mathbf{G}^T + \mathbf{R} \tag{D.12}$$

$\mathbf{P}(k)$ turns out to be quadratic in $\mathbf{K}(k)$ for which the minimum can be found by setting the derivative with respect to $\mathbf{K}(k)$ to zero:

$$\frac{\delta \mathbf{P}(k)}{\delta \mathbf{K}(k)} = -2\Big(\mathbf{G}\bar{\mathbf{P}}(k)\Big)^T + 2\mathbf{K}(k)\mathbf{S}(k) = 0 \tag{D.13}$$

This results in the expression for the Kalman gain:

$$\mathbf{K}(k) = \bar{\mathbf{P}}(k)\mathbf{G}^T\mathbf{S}^{-1} \tag{D.14}$$

Additionally, from equation D.11 and the definition of the Kalman gain the update for the covariance $\mathbf{P}(k)$ can be determined. Writing out the last term of equation D.11 gives:

$$\mathbf{K}(k)\mathbf{S}(k)\mathbf{K}(k)^T = \bar{\mathbf{P}}(k)\mathbf{G}^T\mathbf{S}(k)^{-1}\Big(\mathbf{S}(k)\mathbf{K}(k)^T\Big) = \bar{\mathbf{P}}(k)\mathbf{G}^T\mathbf{K}(k)^T \tag{D.15}$$

Filling this in in equation D.11:

$$\mathbf{P}(k) = \bar{\mathbf{P}}(k) - \bar{\mathbf{P}}(k)\mathbf{G}^T\mathbf{K}(k)^T - \mathbf{K}(k)\mathbf{G}\bar{\mathbf{P}}(k) + \bar{\mathbf{P}}(k)\mathbf{C}^T\mathbf{K}(k)^T \tag{D.16}$$

Clearly, the second and fourth term cancel out and the expression for the updated covariance remains:

$$\mathbf{P}(k) = \Big(\mathbf{I} - \mathbf{K}(k)\mathbf{G}\Big)\bar{\mathbf{P}}(k) \tag{D.17}$$

E

Transformation Matrix for the Virtual Point Transformation

The section shows the derivation of the transformation matrix T used in the VPT. A detailed description of this method can be found in [64]. A matrix $\boldsymbol{R}$ translates the local displacements to six virtual point displacements and rotations plus a residual. The equation below shows the transformation of the three local displacements $\boldsymbol{u}$ to six virtual point displacements $\boldsymbol{q}$ for one sensor.

$$\begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} e_{x,X} & e_{x,Y} & e_{x,Z} \\ e_{y,X} & e_{y,Y} & e_{y,Z} \\ e_{z,X} & e_{z,Y} & e_{z,Z} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & r_Z & -r_Y \\ 0 & 1 & 0 & -r_Z & 0 & r_X \\ 0 & 0 & 1 & r_Y & -r_X & 0 \end{bmatrix} \begin{bmatrix} q_X \\ q_Y \\ q_Z \\ q_{\theta X} \\ q_{\theta Y} \\ q_{\theta Z} \end{bmatrix} + \begin{bmatrix} \mu_x \\ \mu_y \\ \mu_z \end{bmatrix} \tag{E.1}$$

where e represent unit vectors in the corresponding direction and r the distance between the sensor and the virtual point in the corresponding direction. μ is defined as the residual. Or shortly this equation can be written as:

$$\boldsymbol{u}^k = \boldsymbol{R}^{kv} \boldsymbol{q}^k + \boldsymbol{\mu}^k \tag{E.2}$$

where the index k is the sensor index and v the virtual point index. For a problem with three sensors per virtual point the $\boldsymbol{R}^{kv}$ matrices can be stacked to form the following equation:

$$\boldsymbol{u} = \boldsymbol{R}\boldsymbol{q} + \boldsymbol{\mu} \tag{E.3}$$

where:

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{R}^{1,1} & & & \\ \boldsymbol{R}^{2,1} & & & \\ \boldsymbol{R}^{3,1} & & & \\ & \boldsymbol{R}^{4,2} & & \\ & \boldsymbol{R}^{5,2} & & \\ & \boldsymbol{R}^{6,2} & & \\ & & \ddots & \\ & & & \boldsymbol{R}^{N_k,N_v} \end{bmatrix} \tag{E.4}$$

Matrix $\boldsymbol{T}$ is then defined as:

$$\boldsymbol{T} = (\boldsymbol{R}^T \boldsymbol{W} \boldsymbol{R})^{-1} \boldsymbol{R}^T \boldsymbol{W} \tag{E.5}$$

where $\boldsymbol{W}$ is a weighting matrix.

Bibliography

[1] R. Adams and J. F. Doyle. Multiple force identification for complex structures. *Experimental Mechanics*, 42(1):25–36, 2002. ISSN 00144851. doi: 10.1177/0018512002042001786.

[2] B. D. O. Anderson and J. B Moore. *Optimal Filtering*. Prentice Hall, Englewood Cliffs, New Jersey, 1979.

[3] W. Ang and B. Farhang-Boroujeny. A new class of gradient adaptive step-size LMS algorithms. *IEEE Transactions on Signal Processing*, 49(4):805–810, 2001.

[4] K.J. Astrom and B. Wittenmark. *Computer Controlled Systems*. Prentice Hall, 3rd edition, 1997.

[5] M. Aucejo. Structural source identification using a generalized Tikhonov regularization. *Journal of Sound and Vibration*, 333(22):5693–5707, 2014. ISSN 10958568. doi: 10.1016/j.jsv.2014.06.027.

[6] S. E. Azam, E. Chatzi, C. Papadimitriou, and A. Smyth. Experimental Validation of the Dual Kalman Filter for Online and Real-Time State and Input Estimation. *Model Validation and Uncertainty Quantification*, 3, 2015. ISSN 21915652. doi: 10.1007/978-3-319-15224-0. URL `http://link.springer.com/10.1007/978-3-319-15224-0`.

[7] S. E. Azam, E. Chatzi, C. Papadimitriou, and A. Smyth. Experimental validation of the Kalman-type filters for online and real-time state and input estimation. *Journal of Vibration and Control*, 23(15):2494–2519, 2017. ISSN 17412986. doi: 10.1177/1077546315617672.

[8] S.E. Azam, C. Papadimitriou, and E. Chatzi. Recursive Bayesian filtering for displacement estimation via output-only vibration measurements. In *Advances in Civil, Environmental, and Materials Research (ACEM14)*, number January 2015, Busan, Korea, 2014.

[9] J. Barzilai and J.M. Borwein. Two-Point Step Size Gradient Methods. *IMA Journal of Numerical Analysis*, 8:141–148, 1988.

[10] A. Beck and M. Teboulle. A Fast Iterative Shrinkage-Thresholding Algorithm. *Society for Industrial and Applied Mathematics Journal on Imaging Sciences*, 2(1):183–202, 2009. ISSN 1936-4954. doi: 10.1137/080716542.

[11] R. Bellman. The Theory of Dynamic Programming. In *American Mathematical Society*, Laramie, 1954.

[12] Dariusz Bismor, Krzysztof Czyz, and Zbigniew Ogonowski. Review and comparison of variable step-size LMS algorithms. *International Journal of Acoustics and Vibrations*, 21(1):24–39, 2016. ISSN 10275851. doi: 10.20855/ijav.2016.21.1392.

[13] T.G. Carne, R.L. Mayes, and V.I. Bateman. Force reconstruction using the sum of weighted accelerations technique. In *Proceedings of the 10th International Modal Analysis Conference*, pages 291–298, 1992.

[14] Jean Michel Franco, Byron Mauricio Mayag, Johannio Marulanda, and Peter Thomson. Static and dynamic displacement measurements of structural elements using low cost RGB-D cameras. *Engineering Structures*, 153(October):97–105, 2017. ISSN 0141-0296. doi: 10.1016/j.engstruct.2017.10.018. URL `https://doi.org/10.1016/j.engstruct.2017.10.018`.

[15] S. Gillijns and B. De Moor. Unbiased minimum-variance input and state estimation for linear discrete-time systems with direct feedthrough. *Automatica*, 43(5):934–937, 2007. ISSN 00051098. doi: 10.1016/j.automatica.2006.11.016.

[16] D. Ginsberg and C. P. Fritzen. Impact identification and localization using a sample-force-dictionary – General Theory and its applications to beam structures. *Structural Monitoring and Maintenance*, 3(3): 195–214, 2016.

[17] T. Goldstein and S. Osher. The Split Bregman Method for L1-Regularized Problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009. ISSN 1936-4954. doi: 10.1137/080725891. URL `http://epubs.siam.org/doi/10.1137/080725891`.

[18] G. H. Golub, P. C. Hansen, and D. P. O'Leary. Tikhonov Regularization and Total Least Squares. *SIAM Journal on Matrix Analysis and Applications*, 21(1):185–194, 1999. ISSN 0895-4798. doi: 10.1137/S0895479897326432. URL `http://epubs.siam.org/doi/10.1137/S0895479897326432`.

[19] F. E. Gunawan. Levenberg-Marquardt iterative regularization for the pulse-type impact-force reconstruction. *Journal of Sound and Vibration*, 331(25):5424–5434, 2012. ISSN 0022460X. doi: 10.1016/j.jsv.2012.07.025.

[20] U. Güntürkün. Sequential reconstruction of driving-forces from nonlinear nonstationary dynamics. *Physica D: Nonlinear Phenomena*, 239(13):1095–1107, 2010. ISSN 01672789. doi: 10.1016/j.physd.2010.02.014.

[21] P. C. Hansen. Analysis of Discrete Ill-Posed Problems by Means of the L-Curve. *SIAM Review*, 34(4):561–580, 1992. ISSN 0036-1445. doi: 10.1137/1034115. URL `http://epubs.siam.org/doi/10.1137/1034115`.

[22] P. E. Hollandsworth and H. R. Busby. Impact force identification using the general inverse technique. *International Journal of Impact Engineering*, 8(4):315–322, 1989. ISSN 0734743X. doi: 10.1016/0734-743X(89)90020-1.

[23] E. Jacquelin and P. Hamelin. Force recovered from three recorded strains. *International Journal of Solids and Structures*, 40(1):73–88, 2003. ISSN 00207683. doi: 10.1016/S0020-7683(02)00544-9.

[24] M. H. A. Janssens and J. W. Verheij. A pseudo-forces methodology to be used in characterization of structure-borne sound sources. *Applied Acoustics*, 31(3):285 – 308, 2000.

[25] R.E. Kalman. On the General Theory of Control Systems. In *Proceedings of 1st Int. Cong. of IFAC, Moscow*, page 1481, Butterworth, London, 1960.

[26] D. C. Kammer. Input Force Reconstruction Using a Time Domain Technique. *Journal of Vibration and Acoustics*, 120(4):868, 1998. ISSN 07393717. doi: 10.1115/1.2893913. URL `http://vibrationacoustics.asmedigitalcollection.asme.org/article.aspx?articleid=1470090`.

[27] H. Katkhuda. A time domain approach for identifying dynamic forces applied on structures. *Jordan Journal of Civil Engineering*, 7(3):259–269, 2013. ISSN 19930461.

[28] D. De Klerk, D. J. Rixen, and S. N. Voormeeren. General Framework for Dynamic Substructuring: History, Review and Classification of Techniques. *AIAA Journal*, 46(5):1169–1181, 2008. ISSN 0001-1452. doi: 10.2514/1.33274. URL `http://arc.aiaa.org/doi/10.2514/1.33274`.

[29] E. B. Kosmatopoulos, A. W. Smyth, S. F. Masri, and A. G. Chassiakos. Robust Adaptive Neural Estimation of Restoring Forces in Nonlinear Structures. *Journal of Applied Mechanics*, 68(6):880, 2001. ISSN 00218936. doi: 10.1115/1.1408614. URL `http://appliedmechanics.asmedigitalcollection.asme.org/article.aspx?articleid=1555339`.

[30] R. Kwong and E.W. Johnston. Variable step size LMS algorithm. *IEEE Transactions on Signal Processing*, 40(7):631–639, 1992.

[31] T. Lai, T. Yi, and H. Li. Parametric study on sequential deconvolution for force identi fi cation. *Journal of Sound and Vibration*, 377:76–89, 2016. ISSN 0022-460X. doi: 10.1016/j.jsv.2016.05.013. URL `http://dx.doi.org/10.1016/j.jsv.2016.05.013`.

[32] P. Lancaster and K. Salkauskas. Surfaces generated by moving least squares methods. *Mathematics of Computation*, 37(155):141–141, 1981. ISSN 0025-5718. doi: 10.1090/S0025-5718-1981-0616367-1. URL `http://www.jstor.org/stable/2007507{%}0Ahttp://www.ams.org/jourcgi/jour-getitem?pii=S0025-5718-1981-0616367-1`.

[33] S. S. Law, T. H. T. Chan, and Q. H. Zeng. Moving force identification: A time domain method. *Journal of Sound and Vibration*, 201(1):1–22, 1997. ISSN 0022460X. doi: 10.1006/jsvi.1996.0774.

[34] Junhwa Lee, Kyoung-chan Lee, Soojin Cho, and Sung-han Sim. Computer Vision-Based Structural Displacement Measurement Robust to Light-Induced Image Degradation for In-Service Bridges. 2017. doi: 10.3390/s17102317.

[35] K. Liu, S. S. Law, X. Q. Zhu, and Y. Xia. Explicit form of an implicit method for inverse force identification. *Journal of Sound and Vibration*, 333(3):730–744, 2014. ISSN 0022460X. doi: 10.1016/j.jsv.2013.09.040. URL `http://dx.doi.org/10.1016/j.jsv.2013.09.040`.

[36] E. Lourens. *Force Identification in Structural Dynamics*. Phd thesis, KU Leiven, 2012.

[37] E. Lourens, G. Lombaert, G. De Roeck, and G. Degrande. Identification of dynamic axle loads from bridge responses by means of an extended dynamic programming algorithm. *WIW (Weight in Motion), Load Capacity and Bridge Performance*, (September):59–69, 2008.

[38] E. Lourens, E. Reynders, G. De Roeck, G. Degrande, and G. Lombaert. An augmented Kalman filter for force identification in structural dynamics. *Mechanical Systems and Signal Processing*, 27(1):446–460, 2012. ISSN 08883270. doi: 10.1016/j.ymssp.2011.09.025.

[39] Z. R. Lu and S. S. Law. Force identification based on sensitivity in time domain. *Journal of Engineering Mechanics*, 132(10):1050–1056, 2006.

[40] Z. R. Lu and S. S. Law. Identification of system parameters and input force from output only. *Mechanical Systems and Signal Processing*, 21(5):2099–2111, 2007. ISSN 08883270. doi: 10.1016/j.ymssp.2006.11.004.

[41] V. Verdult M. Verhaegen. *Filtering and System Identification: A Least Squares Approach*. Cambridge University Press, Cambridge, 2007.

[42] C. K. Ma and C. C. Ho. An inverse method for the estimation of input forces acting on non-linear structural systems. *Journal of Sound and Vibration*, 275(3-5):953–971, 2004. ISSN 0022460X. doi: 10.1016/S0022-460X(03)00797-1.

[43] K. Maes, E. Lourens, K. Van Nimmen, E. Reynders, G. De Roeck, and G. Lombaert. Design of sensor networks for instantaneous inversion of modally reduced order models in structural dynamics. *Mechanical Systems and Signal Processing*, 52-53(1):628–644, 2015. ISSN 10961216. doi: 10.1016/j.ymssp.2014.07.018.

[44] K. Maes, G. De Roeck, A. Iliopoulos, W. Weijtjens, C. Devriendt, and G. Lombaert. Online Input and State Estimation in Structural Dynamics. *Special Topics in Structural Dynamics*, 6:1–10, 2016. doi: 10.1007/978-3-319-29910-5. URL `http://link.springer.com/10.1007/978-3-319-29910-5`.

[45] V.J. Mathews and Z. Xie. A stochastic gradient adaptive filter with gradient adaptive step size. *IEEE Transactions on Signal Processing*, 41(6):2075–2087, 1993.

[46] F. Naets, J. Cuadrado, and W. Desmet. Stable force identification in structural dynamics using Kalman filtering and dummy-measurements. *Mechanical Systems and Signal Processing*, 50-51:235–248, 2015. ISSN 08883270. doi: 10.1016/j.ymssp.2014.05.042. URL `http://linkinghub.elsevier.com/retrieve/pii/S0888327014002180`.

[47] J. Nagumo and A. Noda. A Learning Method for System Identification. *Transactions on Automatic Control*, 4(3):282–287, 1967.

[48] N. M. Newmark. A method of computation for structural dynamics. *Journal of Engineering Mechanics*, 85:67–94, 1959.

[49] T. S. Nord, E. Lourens, O. Øiseth, and A. Metrikine. Model-based force and state estimation in experimental ice-induced vibrations by means of Kalman filtering. *Cold Regions Science and Technology*, 111:13–26, 2015. ISSN 0165232X. doi: 10.1016/j.coldregions.2014.12.003.

[50] J Prawin and A R. Mohan. An online input force time history reconstruction algorithm using dynamic principal component analysis. *Mechanical Systems and Signal Processing*, 99:516–533, 2018. ISSN 0888-3270. doi: 10.1016/j.ymssp.2017.06.031. URL `http://dx.doi.org/10.1016/j.ymssp.2017.06.031`.

[51] B. Radhika. Force identification using correlated noise models. *Procedia Engineering*, 199:888–893, 2017. ISSN 18777058. doi: 10.1016/j.proeng.2017.09.223.

[52] A. Rezayat, V. Nassiri, S. Vanlanduit, and P. Guillaume. Force identification using mixed and penalized optimization techniques. *Proceedings of International Conference on Noise and Vibration Engineering (Isma2014) and International Conference on Uncertainty in Structural Dynamics (Usd2014)*, pages 2087–2099, 2014.

[53] E. Risaliti, B. Cornelis, T. Tamarozzi, and W. Desmet. Model Validation and Uncertainty Quantification, Volume 3. *Model Validation and Uncertainty Quantificatio*, 3:359–369, 2016. doi: 10.1007/978-3-319-29754-5. URL `http://link.springer.com/10.1007/978-3-319-29754-5`.

[54] G. T. Rocklin, J. Crowley, and H. Vold. A Comparison of H1, H2 and Hv Frequency Response Functions. *Structural Dynamics Research Corporation*.

[55] S. Samagassi, A. Khamlichi, A. Driouach, and E. Jacquelin. Reconstruction of multiple impact forces by wavelet relevance vector machine approach. *Journal of Sound and Vibration*, 359:56–67, 2015. ISSN 10958568 0022460X. doi: 10.1016/j.jsv.2015.08.014.

[56] T.J. Shan and T. Kailath. Adaptive algorithms with an automatic gain control feature. *IEEE Transactions on Circuits and Systems*, 35(1):122–127, 1988.

[57] A. D. Steltzner and D. C. Kammer. Input Force Estimation Using an Inverse Structural Filter. *Journal of Vibration and Acoustics*, 123(4):524–532, 2001.

[58] M. Sturm, A. Moorhouse, W. Kropp, and T. Alber. Robust force identification for complex technical structures with single degree of freedom excitation using an adaptive algorithm in time domain. 1(May 2014): 4–7, 2013.

[59] M. Sturm, A. Moorhouse, W. Kropp, and T. Alber. Robust calculation of simultaneous multi-channel blocked force signatures from measurements made in-situ using an adaptive algorithm in time domain. *20th International Congress on Sound and Vibration 2013, ICSV 2013*, 2(July):1610–1617, 2013.

[60] A. N. Thite and D. J. Thompson. The quantification of structure-borne transmission paths by inverse methods. Part 1: Improved singular value rejection methods. *Journal of Sound and Vibration*, 264(2): 411–431, 2003. ISSN 0022460X. doi: 10.1016/S0022-460X(02)01202-6.

[61] A.N Tikhonov. On the Solution of Ill-Posed Problems and the Method of Regularization. *Soviet Math*, 4: 1035–1038, 1963.

[62] M. Á. Torres-arredondo and C. P. Fritzen. Impact Monitoring in Smart Structures Based on Gaussian Processes. In *4th International Symposium on NDT in Aerospace*, pages 1–8, 2012.

[63] D. M. Trujillo. Application of Dynamic Programming to the General Inverse Problem. *International Journal for Numerical Methods in Engineering*, 12:613–624, 1978.

[64] M. van der Seijs, D. van den Bosch, D. Rixen, and D. de Klerk. an Improved Methodology for the Virtual Point Transformation of Measured Frequency Response Functions in Dynamic Substructuring. *Proceedings of the 4th International Conference on Computational Methods in Structural Dynamics and Earthquake Engineering (COMPDYN 2013)*, (April 2017):4334–4347, 2014. doi: 10.7712/120113.4816.C1539. URL `http://www.eccomasproceedia.org/conferences/thematic-conferences/compdyn-2013/4816`.

[65] M. V. Van Der Seijs, D. De Klerk, and D. J. Rixen. General framework for transfer path analysis: History, theory and classification of techniques. *Mechanical Systems and Signal Processing*, 68-69:217–244, 2016. ISSN 10961216. doi: 10.1016/j.ymssp.2015.08.004. URL `http://dx.doi.org/10.1016/j.ymssp.2015.08.004`.

[66] M. Verhaegen and V. Verdult. *Filtering and Identification: A Least Squares Approach*. Cambridge University Press, Cambridge, 2007.

[67] S N Voormeeren and D J Rixen. A Dual Approach to Substructure Decoupling Techniques. In *Proceedings of the IMAC-XXVIII*, number October 2011, 2011. ISBN 978-1-4419-9833-0. doi: 10.1007/978-1-4419-9834-7. URL http://link.springer.com/10.1007/978-1-4419-9834-7.

[68] B. Widrow and M.E. Hoff. Adaptive Switching Circuits. In *Proceedings of the Western Electronic Show and Convention*, Los Angeles, 1960.

[69] X. Xu and J. Ou. Force identification of dynamic systems using virtual work principle. *Journal of Sound and Vibration*, 337:1–24, 2014. ISSN 0022-460X. doi: 10.1016/j.jsv.2014.10.005. URL http://dx.doi.org/10.1016/j.jsv.2014.10.005.

[70] Y. W. Yang, C. Wang, and C. K. Soh. Force identification of dynamic systems using genetic programming. *International Journal for Numerical Methods in Engineering*, 63(9):1288–1312, 2005. ISSN 00295981. doi: 10.1002/nme.1323.

[71] P. Zhou, W. L. Li, W. Li, and Z. Shuai. Reconstruction of Input Excitation Acting on Vibration Isolation System. *Shock and Vibration*, 2016, 2016. ISSN 10709622. doi: 10.1155/2016/2784380.

[72] T. Zhu, S. N. Xiao, and G. W. Yang. Force identification in time domain based on dynamic programming. *Applied Mathematics and Computation*, 235:226–234, 2014. ISSN 00963003. doi: 10.1016/j.amc.2014.03.008. URL http://dx.doi.org/10.1016/j.amc.2014.03.008.