DLR-IB-AE-GO-2021-80

Development and Investigation of Algorithms for Modal Identification of Linear Systems with a Local Nonlinear Mechanism

Masterarbeit

Hauke Maathuis

DLR Deutsches Zentrum
für Luft- und Raumfahrt

Dokumenteigenschaften

Titel	Development and Investigation of Algorithms for Modal Identification of Linear Systems with a Local Nonlinear Mechanism
Betreff	Masterarbeit am Institut für Dynamik und Schwingungen der Leibniz Universität Hannover
Institut	Institut für Aeroelastik
Erstellt von	Hauke Maathuis
Beteiligte	Dr.-Ing. Marc Böswald, Dr. Keith Soal
Geprüft von	Prof. Dr.-Ing. Jörg Wallaschek, Dr.-Ing. Marc Böswald
Freigabe von	Prof. Dr.-Ing. Lorenz Tichy
Datum	02. Juni 2021
Version	1.0
Dateipfad	

Abstract

Vibration testing is a crucial step within the admission process of an aircraft. Before certification, these tests are essential to validate the existing numerical models and to ensure flutter clearance to guarantee the safe operation of the system. Currently, these tests are performed within the so called Ground Vibration Test, where the aircraft is under operational-like boundary conditions and excited in a hangar with shakers to identify the modal parameters of the system. However, these procedures are very time intensive and take place during a period of aircraft development that is very time critical. To circumvent this problem, the DLR has developed a new method that promises to save time, named Taxi Vibration Test. During this process, the aircraft merely taxis over the uneven runway. The occurring vibrations are recorded by sensors. Complex algorithms are then used to identify the modal parameters. In contrast to the Ground Vibration Test where the aircraft is hung up with cords to imitate the boundary conditions, the nonlinear landing gear is involved within the system which influence the modal parameters. Since the algorithms used for this purpose are formulated for linear systems, this work addresses the problem in which various methods are developed and investigated to identify the overall nonlinear system within the Taxi Vibration Test. In doing so, a promising algorithm is presented that can identify the nonlinearity for the most part.

Contents

List of Figures

Abbreviations

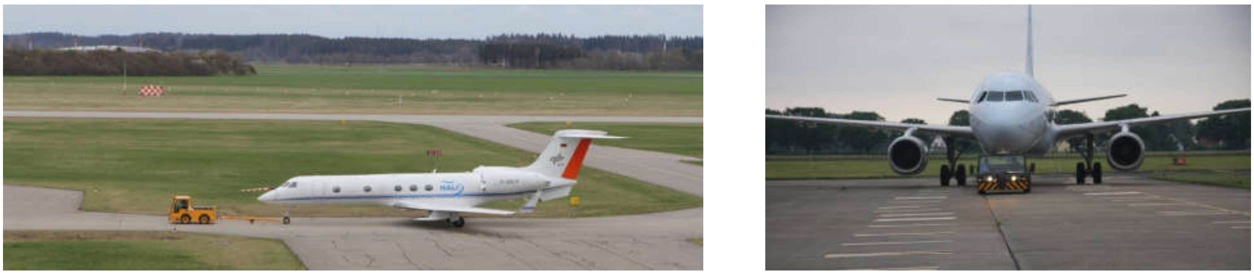
BC	Boundary Condition
DLR	German Aerospace Center
DoF	Degree of Freedom
EMA	Experimental Modal Analysis
FE	Finite Element
FEM	Finite Element Method
FRF	Frequency Response Function
GVT	Ground Vibration Test
IC	Initial Condition
i.e.	id est
LSCF	Least Squares Complex Frequency-domain
LTI	Linear Time Invariant
MAC	Mode Assurance Criterion
MIMO	Multiple Input Multiple Output
ODE	Ordinary Differential Equation
OEM	Operating Empty Mass
OMA	Operational Modal Analysis
PDF	Probability Density Function
RCA	Rainflow Counting Algorithm
RMS	Root Mean Square
SDof	Single Degree of Freedom
SSI	Stochastic Subspace Identification
SVD	Singular Value Decomposition
TVT	Taxi Vibration Test

1 Introduction

Within this introductory chapter, the motivation for this thesis is exposed. Afterwards the problem is defined and the research objectives arising in this work are presented and discussed. The chapter concludes with the structure of this work.

1.1 Motivation

The aim of aircraft manufacturer is to develop always lighter and more economic while being at the most cost-efficient. This makes the resulting structures very susceptible for vibrations. To ensure that these tremendously complex and lightweight systems are still safe and reliable, numerous tests have to be deployed before the first flight. During the development process, time plays a major role which is why experimental tests are suboptimal. With today's computation power, more and more of these tests or respectively analysis can be performed within simulations to save both time and costs. For structural analysis, for example fatigue testing or flutter analysis, the aircraft manufacturers often make use of the well established Finite Element Method (FEM). Anyway, it is not sufficient to just have these computer-based models without any verification. These Finite Element (FE) model verification have to be deployed before the first test flight and hence are in a very time-critical window from the manufacturer's point of view, see Soal et al. [2019]. Within these tests the absence of aeroelastic instability such as flutter, divergence or control reversal must be demonstrated experimentally. This can be achieved e.g. on the basis of identified damping ratios at different flight conditions within the flight envelope. Anyway, the modal parameters, the eigenfrequencies, damping ratios and eigenvectors of a system are of great importance which are afterwards used to validate the structural FE model. A common way to obtain these parameters is the so called Ground Vibration Test (GVT). In this process, the aircraft under investigation is inside a hangar and equipped with many sensors to measure the acceleration. To achieve this, the structure is excited through electro-dynamic shakers at different positions. Due to the complexity of such structures, the aircraft has to be excited using various combinations of excitation positions to ensure that all modal parameters are identified properly, making such a procedure very expensive in terms of time. As already mentioned this is suboptimal because this phase is very time sensitive for aircraft manufacturers. A procedure like the GVT, where the systems input and output are known can be classified within the Experimental Modal Analysis (EMA).
To improve such a procedure in terms of time and costs, the German Aerospace Center (DLR), Institute of Aeroelasticity, Department of Structural Dynamics and System Identification introduced a novel approach, see Böswald and Govers [2008]. This approach is, in contrast to the GVT, not part of the group of EMA techniques but can be classified into the so called Operational Modal Analysis (OMA) methods. This new testing method is called the Taxi Vibration Test (TVT). Throughout this test, the aircraft is pulled over the runway and is excited through the uneven ground. Here, not a determined excitation is used but the natural excitation during taxiing in combination with data-driven algorithms to obtain the desired modal parameter. Although the excitation can not be measured and is thus unknown, it can be described with the help of stochastics. There are a number of such algorithms to extract the modal parameters from the system response, like for example the Stochastic Subspace Identification (SSI) algorithm. Consequently, this test can be integrated in a multi-objective



(a) **(b)**

Figure 1 – Demonstration of the Taxi Vibration Test with the DLR HALO (a) and ATRA (b) aircraft

testing concept to decrease the expensiveness. Additionally, because the aircraft is equipped with sensors for the first flight test anyway, even more preparation time can be saved. Such a test is shown in figure 1, where the two DLR aircraft configurations HALO and ATRA are pulled over the runway.
It is shown in the papers from Soal et al. [2019] and Böswald et al. [2017] that with a sufficiently long time series, the modal parameters can be identified. Further it is demonstrated that even with shorter durations of testing, the data does not show trends of increasing uncertainties, but fewer modes are identified. This is why the authors state that the measurement duration has to be chosen in accordance with speed and objectives of the test campaign. Summarising, the motivation of this work is to further develop this novel method for the identification of the modal parameters of aircraft.

1.2 Description of the problem

During the first TVT, carried out by the DLR, a systematic deviation of the results was observed, see [Böswald and Govers, 2008]. Especially the modes where the landing gear is involved deviate. Since the damping ratios play an important role and need to be identified accurately, the GVT is until now the only way to obtain these modal parameters for model updating. After further investigation of this phenomena, it was concluded that the main problem is the highly nonlinear dynamic landing gear. The stick-slip friction is located within the shock absorbers of the landing gear and has an impact on the modal parameters such that they vary with changing excitation force, see Soal et al. [2019]. As it is mentioned in the motivation, the TVT is an output-only modal analysis and uses algorithms to extract the modal parameters. The problem is that these algorithms are written for linear systems and hence are without any improvements not applicable to nonlinear influenced systems.

1.3 Research objectives

The overall aim of this thesis to extend OMA to nonlinear systems and hence to further develop the aforementioned TVT. This shall be achieved by developing and investigating different algorithms. To do so, a numerical efficient substitute model shall be constructed of which all parameters are constantly accessible and known. This is to ensure that one is able to compute analytical solutions of the system if needed. In order to achieve the numerical efficiency an integration scheme shall be presented to be able to solve nonlinear systems under random excitation. Furthermore, this model shall represent the nonlinear behaviour of an aircraft, initiated through the nonlinear landing gear. For demonstration purposes, different excitation levels are used to simulate the system response to uncover the nonlinearity. Afterwards, it shall be shown that if the nonlinear subsystem is part of the excitation mechanism, the linear and nonlinear subparts can be decoupled to identify the analytical modal parameters of the linear structure. Following this, different methods for nonlinear OMA shall be developed. Since within the TVT different excitation levels can not be realised, only one simulation run shall be used to reveal the nonlinearity. With the aid of statistical analysis the obtained data shall be processed in advance of the modal analysis to be able to identify the overall nonlinear system properly.

1.4 Structure of this thesis

The aim of this work is to further develop the TVT procedure to make it applicable for linear systems with a local nonlinear mechanism. Therefore, theoretical investigations are carried out to develop and investigate different methods for OMA of nonlinear systems. To do so, an Airbus A320 is taken as an example aircraft and is modelled equivalently in a simplified theoretical way to investigate the influence of the landing gear. For this purpose, the needed theory of structural dynamics is explained in chapter 2. The nonlinearity of the landing gear plays a major role, since the modal parameter of modes can change under different excitation levels. Therefore, in chapter 3, the modelling of such nonlinearities and their effect on dynamic systems are presented and explained. After the fundamentals to describe the substitute model mathematically are submitted, an introduction into stochastic processes and signal processing is given in chapter 4. The intention is to describe and generate the excitation caused by the uneven road pavement as accurate as possible in order to realise a simulation of the substitute model. Additionally, this signal shall have optimal properties for modal analysis purposes. Basically, now the structure and the excitation can be united to compute the output, or differently said the response of the system caused by the excitation. At this point it can be taken in advance that the system output is supposed to stand for what the sensors on the aircraft actually record during the TVT. Anyway, to simulate the response, an integration of the equation of motion and its excitation is needed. Thereby the problem arises that for randomly excited nonlinear systems no analytic solution can be computed. Thus, a numerical integration scheme has to be employed. The combination of long simulation times and randomly distributed excitation leads to time-intensive computations which is why an improved Runge-Kutta integration scheme is presented in chapter 5 to fulfil this task faster. Subsequently in chapter 6, two modal analysis methods are presented on which the improvements are later based on. Thereby the focus is on the already quoted SSI algorithm. In addition, the Least Squares Complex Frequency-domain (LSCF) modal analysis is briefly discussed. After explaining the theory, the example problem, the Airbus A320 is mathematically modelled in chapter 7 to obtain a substitute model. Thereby, the before discussed fundamentals in structural dynamics are applied to create the model. The excitation is generated using chapter 4 and the output is computed by employing the developed improved Runge-Kutta scheme in chapter 5. Afterwards the simulation results are presented and the problem is illustrated again. Chapter 8 then discusses the different developed methods and shows whether they are applicable or not. The last chapter, chapter 9, concludes the thesis, discusses the results again and gives an outlook for further research.

2 Structural Dynamics Preliminaries

Structural dynamics in general deals with the vibration processes of structures. This research field has gained more and more attention during the last decades, see Freymann [2011]. A structural dynamic analysis contains a mathematical-mechanical description of the structure with the capability to oscillate, using a discretised and more or less simplified system which is easier to handle, see Gasch et al. [2012]. While at the start of this area the interest laid in the analytic models and the analytic and numeric solution, nowadays the focus is on the numerical analysis, corresponding for example with the well established FEM. The structures which appear for example in aerospace engineering are becoming more and more complex in the course of time, see Kerschen [2017]. For a better understanding of the theory, structural dynamics make use of simple, analytic solvable systems, which then can be investigated in detail. Afterwards, the gained knowledge can be extended to more complex geometries.

The regarded systems are mathematically described by the so called equations of motion, which are usually second order differential equations. If the system is simple enough, it can be possible to derive the equation of motion by hand, but with increasing complexity it becomes more and more difficult. Therefore, other methods have to be employed. An also well established way to do this analytically is to use the so called EULER-LAGRANGE equations, which is based on the energies contained in the system. For complex structures, as they occur in aerospace engineering, numerical solutions have been widely adopted, like the before mentioned FEM. As the name suggests, this method divides the complex geometries into finite elements, for which the governing partial differential equations are solved. Thereby, the displacements of the element-nodes are approximated with so called ansatz functions, which are polynomials. The approximation adjusts the coefficients of these polynomials such that the displacement of the whole structure is described by the displacements of subparts, the elements. This procedure can be done for several geometric objects such as squares and triangles in the two dimensional and cuboids for a three dimensional space. These simple geometries can then be put together to form the desired complex structure. Thereby, different assumptions are made, such as for example small deformations, which again leads to a linearised model [Wriggers, 2001]. An example is given in figure 2 where a meshed aircraft is presented, illustrating a FEM model.

Figure 2 – Artistic Finite Element model of an aircraft (from: progresstech.am)

It becomes clear that this complex geometry is divided into several elements with rather simple geometric shape and known material parameters for which the above described approximation procedure is applied. Each element has a certain amount of so called nodes, mostly located in the corner-points of each element. Depending on the choice of the element, each node represents a maximum of 6 Degree of Freedom (DoF), through which all of the three translatory and three rotatory DoF are covered. It is worth to mention that by increasing the number of DoF per element, the number of DoF of the whole system increases rapidly, too. By respecting the initial and boundary conditions, the system of equation within the FEM can be solved to obtain the so called system matrices, which can be used to construct the equation of motion and thus the mathematical description of the model. At this point it is worth to mention that in some cases these systems can consists of several million DoF and hence the same amount of equations, which shows the need for computational resources, see Henwood and Bonet [1996]. For further details on FEM, the reader is referred to Bathe [2014], Tenek and Argyris [1998] as well as the already cited literature.

Since the aim of this work is to investigate different methods to characterise and identify non-linear structural dynamic systems, a brief introduction into the fundamentals of structural dynamics in the linear range is given, where the mathematical equations are introduced, as well as different formulations to rewrite these systems in differential equations. Therefore, the fundamental equations of structural mechanics are shown which are then transformed to the so called state-space formulation.

This chapter shall provide the reader with the fundamental equations in structural dynamics. After already introducing the use of FEM in the aviation industry, at first a step is taken backwards to start at the beginning of formulating the equation of motion based on Lagrange's equation of the second kind as an example.

2.1 Lagrange's equation of the second kind

The equation of motion results from the physical laws, established by Newton and Euler and is a mathematical equation to describes how a mechanical system evolves in space and time. There are several methods to derive such an equation. For a fundamental derivation, consider the scalar functional L which is called the LagrangIan and describes the system energetically. Furthermore, the vector of generalised coordinates in a three dimensional system $\mathbf{q}_s \in \mathbb{R}^3$ is introduced, representing each point of mass of the system. For a rigid body motion, the Lagrange equation depends on the generalised coordinates and their time derivative, written as

$$L(\mathbf{q}, \dot{\mathbf{q}}) = T - U, \tag{2.1}$$

where T denotes the kinetic and U the potential energy of the system, see Pletser [2018]. For the sake of completeness, it should be mentioned that for non-rigid body motions an additional term appears, representing the strain energy functional. In the further course of this work, however, only rigid body motions will be considered, letting this term vanish.

Anyway, by applying Hamilton's principle, which is based on a variation of the functional L, it can be shown [Han and Benaroya, 2002] that the evolution of a conservative mechanical system can be described by

$$\left(\frac{d}{dt}\frac{\partial}{\partial \dot{\mathbf{q}}_s} - \frac{\partial}{\partial \mathbf{q}_s}\right) L = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\mathbf{q}}_s}\right) - \frac{\partial L}{\partial \mathbf{q}_s} = 0. \tag{2.2}$$

Equation (2.2) describes a conserving system. If the system does dissipate energy, the so called non-conservative forces have to be introduced, denoted by Q and extend equation (2.2) as

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\mathbf{q}}_s}\right) - \frac{\partial L}{\partial \mathbf{q}_s} - Q_s = 0. \tag{2.3}$$

By separating all variables which occur in the second time derivative on the left hand side, the equation of motion can be expressed as [Wallaschek, 2018]

$$\ddot{\mathbf{q}} = f(\dot{\mathbf{q}}, \mathbf{q}, t). \tag{2.4}$$

where f denotes a function.

2.2 The linearised equation of motion and the eigenvalue problem

The equation of motion, derived in the previous chapter is in most cases non-linear. In many engineering disciplines it is however sufficient to assume small displacements of the mechanical systems. In other words, it will remain close to its equilibrium. If this assumption is made, the system can be linearised. Therefore, the vector of the generalised coordinates $\mathbf{q}_s(t)$ is regarded again. Additionally, the system is considered to be in an equilibrium position $\mathbf{q}_s(t = 0) = \mathbf{q}_{eq}$. For the linearisation, equation (2.4) is evoked again. The non-conservative forces are neglected for the linearisation. The again rewritten equation of motion writes

$$f(\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q}, t) = 0. \tag{2.5}$$

For an equilibrium position, the system can be expressed as $f(\ddot{\mathbf{q}} = 0, \dot{\mathbf{q}} = 0, \mathbf{q} = \mathbf{q}_{eq}) = 0$. In the following let $\tilde{\mathbf{q}}$ and its time derivatives be a small deviation from the equilibrium

$$\mathbf{q} = \mathbf{q}_{eq} + \tilde{\mathbf{q}} \qquad \dot{\mathbf{q}} = \dot{\tilde{\mathbf{q}}} \qquad \ddot{\mathbf{q}} = \ddot{\tilde{\mathbf{q}}}. \tag{2.6}$$

If this expression is expanded around the equilibrium position by using Taylor series, this yields

$$\begin{aligned} f(\ddot{\mathbf{q}} = \ddot{\tilde{\mathbf{q}}}, \dot{\mathbf{q}} = \dot{\tilde{\mathbf{q}}}, \mathbf{q} = \mathbf{q}_{eq} + \tilde{\mathbf{q}}) &\approx f(\ddot{\mathbf{q}} = 0, \dot{\mathbf{q}} = 0, \mathbf{q} = \mathbf{q}_{eq} + \tilde{\mathbf{q}}) + \\ &\left.\frac{\partial f}{\partial \ddot{\mathbf{q}}}\right|_{\substack{\ddot{\mathbf{q}}=0\\ \dot{\mathbf{q}}=0\\ \mathbf{q}=\mathbf{q}_{eq}}} \ddot{\tilde{\mathbf{q}}} + \left.\frac{\partial f}{\partial \dot{\mathbf{q}}}\right|_{\substack{\ddot{\mathbf{q}}=0\\ \dot{\mathbf{q}}=0\\ \mathbf{q}=\mathbf{q}_{eq}}} \dot{\tilde{\mathbf{q}}} + \left.\frac{\partial f}{\partial \mathbf{q}}\right|_{\substack{\ddot{\mathbf{q}}=0\\ \dot{\mathbf{q}}=0\\ \mathbf{q}=\mathbf{q}_{eq}}} \tilde{\mathbf{q}} = 0. \end{aligned} \tag{2.7}$$

The first term vanishes due to the definition of the equilibrium position, resulting in the linearised equation of motion, as

$$\mathbf{M}\ddot{\tilde{\mathbf{q}}}(t) + \mathbf{C}\dot{\tilde{\mathbf{q}}}(t) + \mathbf{K}\tilde{\mathbf{q}}(t) = \mathbf{f}(t). \tag{2.8}$$

Since small displacements are always assumed in this thesis, the linearised and generalised coordinates and their time derivatives are defined as $\tilde{\mathbf{q}} = \mathbf{q}$. Let N be the number of DoF of the regarded system, the symbols denote:

$\mathbf{M} \in \mathbb{R}^{N \times N}$	mass matrix	$\mathbf{q}(t) \in \mathbb{R}^{N}$	displacement vector
$\mathbf{C} \in \mathbb{R}^{N \times N}$	damping matrix	$\mathbf{f}(t) \in \mathbb{R}^{N}$	force vector
$\mathbf{K} \in \mathbb{R}^{N \times N}$	stiffness matrix		

These symmetric matrices are called system matrices. In engineering disciplines, especially in aerospace engineering, very complex structures occur. In this case, the system matrices are approximated using for example the already mentioned FEM.

In the following, the solution for this Ordinary Differential Equation (ODE) (2.8) shall be briefly shown. Therefore, an ansatz of the complex-exponential type is used, like

$$\mathbf{q}(t) = \hat{\mathbf{q}} e^{\lambda t} \qquad \lambda \in \mathbb{C}, \tag{2.9}$$

where $\hat{\mathbf{q}}$ is an unknown amplitude and λ an unknown parameter, called eigenvalue. Inserting the ansatz into the equation of motion and rearranging it, the solution can be expressed as

$$(\lambda^2 \mathbf{M} + \lambda \mathbf{C} + \mathbf{K})\hat{\mathbf{q}} = \mathbf{0}. \tag{2.10}$$

In mathematics an equation of this type is called an eigenvalue problem. To solve this, the demand is made that the determinant of the expression must become zero, as in

$$\mathrm{Det}(\lambda_i^2 \mathbf{M} + \lambda_i \mathbf{C} + \mathbf{K}) = 0. \tag{2.11}$$

A system with N DoF delivers a $2N$-th order polynomial, also called the characteristic polynomial, which has to be solved. This procedure brings $2N$ complex-conjugated eigenvalues.[1] Ongoing, each complex eigenvalue has a corresponding complex eigenvector. The eigenvectors can be determined by inserting the eigenvalues into the eigenvalue problem, see equation (2.10). This can be written as

$$(\lambda_i^2 \mathbf{M} + \lambda_i \mathbf{C} + \mathbf{K})\boldsymbol{\phi}_i = \mathbf{0} \qquad \boldsymbol{\phi}_i \in \mathbb{C}^{N} \tag{2.12}$$

Since the eigenvalues exist in complex-conjugated pairs, the eigenvectors do as well. The ordering of the eigenvectors is given by the eigenvalues.

Subsequently, the orthogonality of the eigenvectors should be briefly introduced. This property results from the fact, that the eigenvalues λ_i and the corresponding eigenvectors $\boldsymbol{\phi}_i$ solve the eigenvalue problem. By arranging all eigenvectors into a matrix, like

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\phi}_1 & \boldsymbol{\phi}_2 & \cdots & \boldsymbol{\phi}_N \end{bmatrix} \qquad \boldsymbol{\Phi} \in \mathbb{C}^{N \times N}, \tag{2.13}$$

[1] Only for systems which are capable to vibrate and which are not overcritically damped

the so called modal matrix, a modal transformation can be performed, based on the relationship

$$\mathbf{q} = \boldsymbol{\Phi}\boldsymbol{\eta}, \tag{2.14}$$

where $\boldsymbol{\eta}$ represent the new introduced modal coordinates. Further it can be shown that by multiplying the equation of motion with the transpose of the modal matrix from the left and with the non-transposed one from the right, like

$$\boldsymbol{\Phi}^T\mathbf{M}\boldsymbol{\Phi}\ddot{\boldsymbol{\eta}} + \boldsymbol{\Phi}^T\mathbf{C}\boldsymbol{\Phi}\dot{\boldsymbol{\eta}} + \boldsymbol{\Phi}^T\mathbf{K}\boldsymbol{\Phi}\boldsymbol{\eta} = \mathbf{0}, \tag{2.15}$$

the equation of motion can be decoupled and yields

$$\mathbf{M}_\Phi\ddot{\boldsymbol{\eta}} + \mathbf{C}_\Phi\dot{\boldsymbol{\eta}} + \mathbf{K}_\Phi\boldsymbol{\eta} = \mathbf{0}. \tag{2.16}$$

After normalising the eigenvectors to one, the decoupled mass and stiffness matrices can be written as

$$\mathbf{M}_\Phi = \mathrm{diag}(m_i) \qquad\qquad \mathbf{K}_\Phi = \mathrm{diag}(\lambda_i^2). \tag{2.17}$$

In case of proportional damping, the damping matrix can be diagonalised as well by

$$\mathbf{C}_\Phi = \mathrm{diag}(2c_i\lambda_i). \tag{2.18}$$

where m_i is called the modal mass and c_i the modal damping. In this resulting equation of motion, only the main diagonals of the matrices are populated.

2.3 Modal damping for time-domain analysis

Since $\mathbf{M}$ and $\mathbf{K}$ can be approximated by, for instance, FEs, the question is how the damping matrix $\mathbf{C}$ can be constructed. It can be shown that an undamped system, once excited, oscillates indefinitely with a constant amplitude, see Paz and Leigh [2004]. In reality no completely undamped systems exist, see Markert [2016]. This is why it is crucial to consider how to model damped systems since within the numerical solutions it is the aim to predict the behaviour of the structure as accurate as possible.

There are several methods to build the damping matrix. Well established ones are for example the proportional or Rayleigh damping as well as the structural damping. Proportional damping uses the mass and stiffness matrices $\mathbf{M}$ and $\mathbf{K}$ in combination with two independent scalar factors. However, the problem arises that not the whole needed band-width of frequencies is covered and thus, overcritical damping can occur. Additionally, it is important to mention that this form of damping is not in every case applicable, especially when it comes to structures consisting of different materials [Böswald et al., 2016]. Structural damping in turn is indeed simple to define, but only within the frequency domain and therewith

inapplicable for the desired time domain analysis, see Böswald et al. [2016]. Hence, a third method is introduced, the so called modal damping.
Within modal damping the damping matrix is defined through the desired modal damping ratios. Each mode produces a contribution to the global matrix, see Link [2014]. This is done under the assumption that the eigenmodes are not coupled with each other. Assuming that modal damping values c_i can be obtained from experiments or empirical values, within this approach the damping matrix is constructed by using the orthogonality relations of the modal matrix $\mathbf{\Phi}$ like

$$\begin{aligned} \mathbf{\Phi}^T\mathbf{M}\mathbf{\Phi} &= \mathbf{M}_\Phi = \mathrm{diag}(m_i) \\ \mathbf{\Phi}^T\mathbf{K}\mathbf{\Phi} &= \mathbf{K}_\Phi = \mathrm{diag}(k_i) \\ \mathbf{\Phi}^T\mathbf{C}\mathbf{\Phi} &= \mathbf{C}_\Phi = \mathrm{diag}(c_i). \end{aligned} \tag{2.19}$$

Equation $(2.19)_1$ can be rewritten and shows the relationship

$$\begin{aligned} \mathbf{I} &= \mathbf{M}_\Phi^{-1}\mathbf{M}_\Phi \\ &= (\mathbf{M}_\Phi^{-1}\mathbf{\Phi}^T\mathbf{M})\mathbf{\Phi} = \mathbf{\Phi}^{-1}\mathbf{\Phi}. \end{aligned} \tag{2.20}$$

Rearranging equation $(2.19)_3$, the damping matrix $\mathbf{C}$ can be expressed by

$$\mathbf{C} = \mathbf{\Phi}^{-T}\mathbf{C}_\Phi\mathbf{\Phi}^{-1} = \mathbf{M}\mathbf{\Phi}\underbrace{\mathbf{M}_\Phi^{-1}\mathbf{C}_\Phi\mathbf{M}_\Phi^{-1}}_{\bar{c}_i=\frac{2c_i\omega_i}{m_i}}\mathbf{\Phi}^T\mathbf{M} \tag{2.21}$$

where $\bar{c}_i$ modal damping for decoupled differential equations.[2] Equation (2.21) can then be rewritten as

$$\mathbf{C} = \mathbf{M}\left[\sum_{i=1}^{N_c}\frac{2c_i\omega_i}{m_i}(\boldsymbol{\phi}_i \otimes \boldsymbol{\phi}_i\right]\mathbf{M} \qquad \mathbf{C} \in \mathbb{C}^{N\times N}. \tag{2.22}$$

The dyadic product of the desired mode shape vectors $\boldsymbol{\phi}_i$ forms a fully populated matrix. The rank of the damping matrix is defined by the number of modal contributions N_c, as $\mathrm{rank}(\mathbf{C}) = N_c$. Since the matrix is fully populated, there is no band-structure like in the mass and stiffness matrix which shows that the matrix does not have a proper physical meaning but can therefore be used in time domain. The ability to calculate the damping with respect to the mode shapes raises the question how to choose or define the latter. By computing the eigenvalues and eigenvectors of the system and incorporating the boundary conditions, the system is reduced by the constrained DoF. To obtain the correct size of the mode shapes, the eigenvectors have to be appended by zeros such that they replaces the constrained DoF. Several publications showed the performance of modal damping, see Bianchi et al. [2010] and Hasselman et al. [1993].
Anyway, now that the damping matrix $\mathbf{C}$ can also be constructed, FEM and the method presented in this section can be used to formulate the equation of motion as shown in equation (2.8).

[2] Only valid if the Modal Damping Assumption is applied.

2.4 Time-domain response analysis using state-space models

After defining the equation of motion in matrix formulation, the aim is now to transform this system into the so called state space formulation. The huge advantage of this kind of formulation is the reduction of the order. After re-formulating the equation of motion, the system is described by a first-order ODE instead of a second-order one, but the number of equations is doubled. Especially when it comes to integrating the equation of motion this formulation shows its strength since only one integration has to be carried out. To do so, another equation has to be introduced, which does not have to contain additional information about the system itself. By calling again the equation of motion (2.8), multiplying it with the inverse of the mass matrix $\mathbf{M}$ from left and introducing a second equation, like

$$\begin{aligned} \dot{\mathbf{q}}(t) - \dot{\mathbf{q}}(t) &= \mathbf{0} \\ \ddot{\mathbf{q}}(t) + \mathbf{M}^{-1}\mathbf{C}\dot{\mathbf{q}}(t) + \mathbf{M}^{-1}\mathbf{K}\mathbf{q}(t) &= \mathbf{M}^{-1}\mathbf{f}(t). \end{aligned} \tag{2.23}$$

the state space formulation of reduced order can be defined as

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \dot{\mathbf{x}}(t) + \begin{bmatrix} \mathbf{0} & -\mathbf{I} \\ \mathbf{M}^{-1}\mathbf{K} & \mathbf{M}^{-1}\mathbf{C} \end{bmatrix} \mathbf{x}(t) = \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1} \end{bmatrix} \mathbf{f}(t) \qquad \mathbf{x}(t) = \begin{bmatrix} \mathbf{q}(t) \\ \dot{\mathbf{q}}(t) \end{bmatrix}. \tag{2.24}$$

By rearranging the equation and introducing the matrices $\mathbf{A}$ and $\mathbf{B}$, the system can be written as

$$\dot{\mathbf{x}}(t) = \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{C} \end{bmatrix}}_{\mathbf{A}} \mathbf{x}(t) + \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1} \end{bmatrix}}_{\mathbf{B}} \mathbf{u}(t). \tag{2.25}$$

In summary, the state-space representation of time continuous systems can be defined as

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t), \end{aligned} \tag{2.26}$$

where the symbols denote

$\mathbf{A} \in \mathbb{C}^{2N \times 2N}$	state matrix	$\mathbf{x}(t) \in \mathbb{C}^{2N}$	state vector
$\mathbf{B} \in \mathbb{C}^{2N \times N}$	input matrix	$\mathbf{u}(t) \in \mathbb{C}^{N}$	input vector
$\mathbf{C} \in \mathbb{C}^{2N \times 2N}$	output matrix	$\mathbf{y}(t) \in \mathbb{C}^{N}$	output vector
$\mathbf{D} \in \mathbb{C}^{2N}$	feedthrough matrix		

After finding a solution for the vector $\mathbf{x}(t)$, the output matrix and the feedthrough matrix can be used to compute the freely selectable output vector, for example to compute afterwards the acceleration. The eigenvalues and eigenvectors can be computed in the same manner as before, by

$$\mathrm{Det}(\lambda\mathbf{A} + \mathbf{B}) = \mathbf{0}. \tag{2.27}$$

Ongoing, it can be shown that the analytical response of a system, described by equation (2.26), is given by [Juang, 1993]

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}(0) + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau. \tag{2.28}$$

Up to this point only a time-continuous formulation is regarded. The transformation from the continuous to the discretised model can be found for example in Murray-Smith [1995]. Here, the data is available in a discrete manner where **u** changes only at discrete time points. For $k \in \mathbb{N}$, this can be written with the so called hold function, as

$$\mathbf{u}(t) = \mathbf{u}(k) \qquad\qquad kT \leq t \leq (k+1)T \tag{2.29}$$

Therewith, equation (2.28) can be modified in terms of a discrete expression by

$$\mathbf{x}(k+1) = e^{\mathbf{A}T}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k)\left(\int_0^T e^{\mathbf{A}\alpha}d\alpha\right), \qquad \alpha = (k+1)T - \tau. \tag{2.30}$$

By rewriting the continuous form to the discrete form, the solution can be computed at the relevant sampling instants. Thus, the discrete state-space model is represented by

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}_d\mathbf{x}(k) + \mathbf{B}_d\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}_d\mathbf{x}(k) + \mathbf{D}_d\mathbf{u}(k) \end{aligned} \tag{2.31}$$

where

$$\begin{aligned} &\mathbf{A}_d = e^{\mathbf{A}T} \quad && \mathbf{B}_d = \mathbf{B}\left(\int_0^T e^{\mathbf{A}(kT+T-\tau)}d\tau\right) \\ &\mathbf{C}_d = \mathbf{C} && \mathbf{D}_d = \mathbf{D} \end{aligned} \tag{2.32}$$

The subscript d defines the discrete formulation. The huge advantage of the discrete form in equation (2.31) is that the next time step can be simply computed by inserting the current one, which can be evaluated much more efficient than for example equation (2.26).

3 Modelling Friction Nonlinearities for Time-domain Analysis

So far, only the linear theory of structural mechanics is regarded. For these linear system and consequently their governing linear differential equation, there are well defined analytical tools to obtain a solution, like for example LAPLACE or FOURIER. In contrast to that, if nonlinear systems occur, no systematic analytic tools are available. The question arises how linear and nonlinear systems can be distinguished from each other. Lakshmanan and Rajasekar [2003] state that linear systems admit the so called linear superposition principle. If u_1 and u_2 are two independent solution of a linear homogeneous differential equation, here denoted with the linear differential operator L, the solution is given by

$$Lu = 0. \tag{3.1}$$

If superposition is fulfilled, it can be shown that the linear combination of u_1 and u_2 written as

$$u = au_1 + bu_2 \tag{3.2}$$

where a and b are two arbitrary complex constants are also a solution of the homogeneous differential equation. If the operator L does not represent a linear differential equation, the system counts as nonlinear and superposition can not longer be applied. If these relationships are now transferred back to structural dynamics, it can be stated that the equations must be linear in nature. Accordingly, elasticity and viscoelasticity, for example, must also be described by a linear behaviour too. If this is not the case nonlinearities appear and have to be treated with attention. It is important to expand the analysis models to these non-linearities, otherwise false results are obtained. In aircraft structures the phenomena often occur by joints and friction. Lee and Tron [1989] state, that the often-exhibiting nonlinearities in aircraft structures affect not only the flutter speed but also the characteristics of the flutter motion. Regarding this statement, it is even more important to further investigate these nonlinearities to obtain more reliable results. But also the eigenfrequencies and eigenvectors behave differently under non-linear influence. A part of an aircraft which is treated in this thesis is the landing gear, which is very nonlinear because of friction.

In section 2.2, the linearisation of the equation of motion is discussed. This linearisation is performed with the assumption of small displacements of the structure, which is sufficient for many engineering tasks. This assumption shall also be assumed to be true in the further course of this work. By evoking the superposition principle again, the friction is the source of the nonlinearity since it can not longer be described by using the linear theory, which shall be briefly explained now.

For modelling this friction phenomena, the well-known mass-spring model is extended to a nonlinear model, using the so called MASING element, which consists of a parallel connection of two springs where one spring is in turn connected in series with a slider element, see figure 3. Starting from a characteristic threshold, this element dissipates energy.

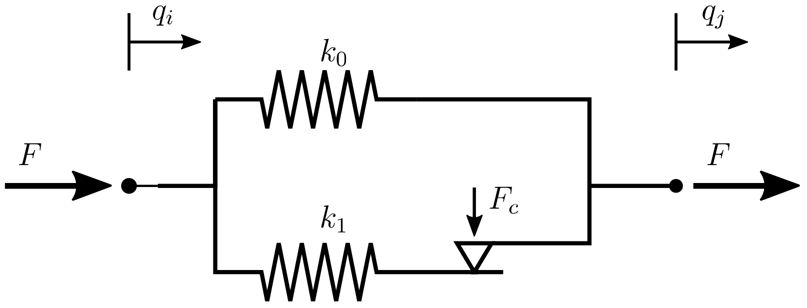


Figure 3 – MASING element

By thinking about the eigenfrequencies and damping ratios of a system containing such an element, it becomes clear that they have to change somehow and depend on the displacement amplitude. It is shown in Böswald [2006], that the two modal parameters, the eigenfrequency and the damping ratio, follow the course, illustrated in figure 4. Under harmonic excitations, the Harmonic-Balance-Method can be applied to compute the equivalent stiffness and damping ratios of such a system which are defined by

$$\begin{aligned} k_{eq}(\hat{u}) &= \begin{cases} k_0 + k_1 & \forall \hat{u} \leq u_c \\ k_0 + \frac{k_1}{\pi}\left(\alpha - \frac{1}{2}\sin(2\alpha)\right) & \forall \hat{u} > u_c \end{cases} & u_c &= \frac{h}{k} \\ c_{eq}(\hat{u}) &= \begin{cases} 0 & \forall \hat{u} \leq u_c \\ \frac{k_1}{2\pi\Omega}(1-\cos(2\alpha)) & \forall \hat{u} > u_c \end{cases} & \cos(\alpha) &= 1 - 2\frac{h}{k_1 \hat{u}} \end{aligned} \tag{3.3}$$

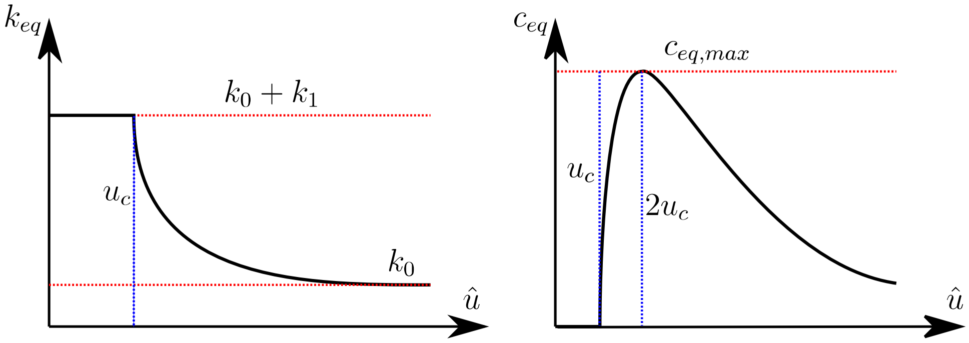


Figure 4 – Dependency of eigenfrequencies and damping ratios from displacement amplitude within the MASING element [Böswald, 2006]

It can be seen that the stiffness is constant until a specific threshold and then decreases with increasing amplitude. Equally the damping ratio stays constant up to an threshold but then increases with increasing amplitude, in contrast to the stiffness. The increasing damping is caused by the supplementary friction of the system. Additionally it is shown that the damping has a maximal value when the displacement amplitude is twice as high as the threshold. The area enclosed by the resulting hysteresis, as for example in figure 6, then measures the energy dissipation within one cycle and is thus a measure for damping. Of course these changes in the modal parameter of the system have an impact on the frequency response function which then show a shift of the eigenfrequency to the left.

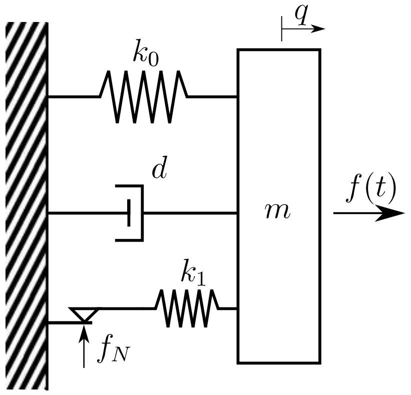


Figure 5 – Single degree of freedom oscillator with a friction nonlinearity

Anyway, the non-linear equation of motion for the Single Degree of Freedom (SDof) system, shown in figure 5 then yields

$$m\ddot{q} + d\dot{q} + kq + f_{nl} = f(t) \tag{3.4}$$

where f_{nl} represents the non-linear force, produced by the friction element. Different models exist to describe f_{nl} and probably the most intuitive one is based on static friction, like the Coulomb friction model, represented by

$$f_{nl} = f_c \text{sign}(v) = \mu F_N \text{sign}(v), \tag{3.5}$$

where the modelled force occurs tangential between the two friction surfaces. f_c is the Coulomb friction force, f_N the normal force and μ a friction coefficient. It should be remarked that the so defined force just addresses the friction force in case of sliding while the sticking phase has to be treated separately, see Gaul and Nitsche [2001]. The most obvious way to solve this model is to perform a case differentiation to define either a stick or a slip phase. Besides the fact, that this method can become numerically very inefficient, the transition from one to another of these stages is however very abrupt, and hence for a dynamic process not ideal. Additionally, neither the so called microslip nor the Stribeck effect can be simulated with such a model which is why more sophisticated ones have to be employed.
For dynamic friction models, the quite well known LuGre model, introduced in Canudas de Wit et al. [1995], is defined as

$$\begin{aligned} f_{nl} &= \sigma_0 w + \sigma_1 \dot{w} + \sigma_2 \dot{q} \\ \dot{w} &= \dot{q} - \sigma_0 \frac{|\dot{q}|}{g(\dot{q})} w \qquad\qquad w(t=0) = w_0 \\ g(v) &= f_c + (f_s - f_c) e^{-\frac{\dot{q}}{v_s}^2} \end{aligned} \tag{3.6}$$

It can be seen that the friction force depends on the relative velocity $\dot{q}$. The variable w is the friction state variable and σ_0, σ_1, σ_2 are material parameters. An important advantage of this model is the capability to model smooth transition from sticking to sliding friction (id est (i.e.) microslip effect) and the STRIBECK effect, see Gaul and Nitsche [2001], which occurs for example in self-exciting systems. Since the STRIBECK effect and possible self-exciting effects are not relevant for modelling the nonlinear damping of the landing gear, the focus is here on another model, the VALANIS model, known from endochromic plasticity [Valanis, 1971]. The advantage of this model over the LUGRE model in this application is that the transition from sticking to sliding friction can be better controlled through the later introduced parameter κ. The VALANIS model is represented through a single first order differential equation

$$f_{nl}'(z) + \lambda f(z) = E_0 q'(z) + \lambda E_t q(z). \tag{3.7}$$

where f denotes a generalised force, q is a generalised displacement and E_0, E_t and λ are material parameters. The variable z is the so called state variable and is defined by

$$\dot{z}(t) = \left| \dot{q}(t) - \kappa \frac{f(t)}{E_0} \right|, \tag{3.8}$$

where κ is another dimensionless parameter. By setting $\dot{f} = f'\dot{z}$ and rearranging the equation (3.7) yields

$$\dot{f}_{nl} = \frac{E_0 \dot{q}_r \left[1 + \frac{\lambda}{E_0}\operatorname{sgn}(\dot{q}_r)(E_t q_r - f_{nl})\right]}{1 + \kappa \frac{\lambda}{E_0}\operatorname{sgn}(\dot{q}_r)(E_t q_r - f_{nl})} = \dot{f}_{nl}(q_r, \dot{q}_r, f_{nl}), \tag{3.9}$$

where q_r is the relative displacement between the two sliding surfaces. The parameter κ defines the microslip and thus the shape of the hysteresis and can take values as $0 \leq \kappa < 1$. The closer this value reaches the value 1, the sharper is the transition phase between the stick and slip phase. Additionally, a parameter f_c is introduced, which can be interpreted as a threshold force which describes the point where the slip phase starts. This parameter is contained in λ and can be computed by

$$\lambda = \frac{E_0}{f_c \left(1 - \kappa \frac{E_t}{E_0}\right)}, \tag{3.10}$$

with

$$E_0 = k_0 + k_1, \qquad E_t = k_0. \tag{3.11}$$

Thereby is E_0 the overall stiffness and E_t the stiffness in case of slipping. The resulting hysteresis and the different model parameters appearing in it are illustrated in figure 6.

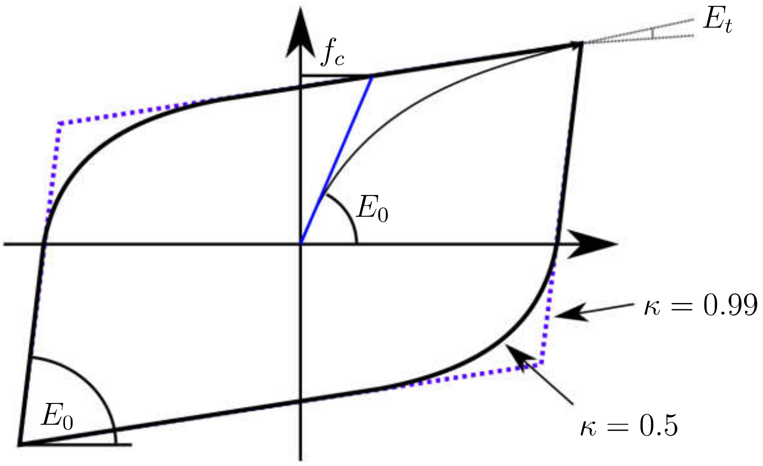


Figure 6 – Hysteresis curve

To conclude this section and to bring this theory into the context of an multiple DoF system, the state-space model can be extended by the non-linear part, as indicated in equation (3.4), described by the Valanis model in equation (3.9) as

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{B}\mathbf{u}_{nl}(t) \qquad \mathbf{u}_{nl}(t) = \begin{bmatrix} \mathbf{0} \\ \mathbf{f}_{nl}(t) \end{bmatrix}. \tag{3.12}$$

where $\mathbf{f}_{nl}$ is the integrated friction force similar to equation (3.9).

4 Stochastic Excitation Modelling and Signal Processing

This section shall now introduce the topic of stochastic excitations, how they are modelled and their application. Aircraft structures are often excited by random processes, for instance while taxiing, caused by the uneven runway, or an excitation due to turbulences during operational flight. The occurring mechanisms cannot be predicted and recreated exactly. Nevertheless, these processes follow certain regularities which can be described by using statistical analysis and the probability theory.
Referring to the central limit theorem [Bendat and Piersol, 2010], [Papoulis, 2002], a random signal $u(t)$ which is compounded by n different contributions u_i, can be defined as normally distributed, independent of the distribution of each sample. This distribution is also called the GAUSSian distribution theorem. With it, many random sources in nature can be described. Ongoing, this statistical behaviour shall be fundamentally defined. Following this, the required theory of signal processing is briefly explained.

4.1 Statistical definition for random processes

A random process is defined as stationary and ergodic. Stationary means that all statistical properties do not change in time, whereas ergodicity refers to the equality of different measured signals. The mean of a signal $u(t)$ is calculated by

$$\mu_u = E[u(t)] = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} u(t)dt. \tag{4.1}$$

For arbitrary t_i the mean value is equal like

$$E[u_1] = E[u_2] = E[u_i]. \tag{4.2}$$

The mean or also called the expectation operator can be described by an average value of the function and is thus computable by

$$E[u(n)] = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} u. \tag{4.3}$$

Furthermore, the probability distribution $P(u)$ and its derivative, the Probability Density Function (PDF) $P_u(u)$, are defined as

$$P(u) = \text{Prob}[u(t) \le u] \qquad\qquad P_u(u) = \frac{d}{du}[P(u)] \tag{4.4}$$

The probability density function defines the probability of the occurrence of the different possible results of an "experiment". A very prominent example for such a function is the well known GAUSSian distribution. As it is shown in equation $(4.4)_2$ the PDF is the derivative with respect to the random variable u. The function then describes the probability of the random variable falling into a specific range. This is represented through the integral under the PDE, visualised in figure 7. The blue area indicates the probability that a value falls into this range. The total area is of course always one.

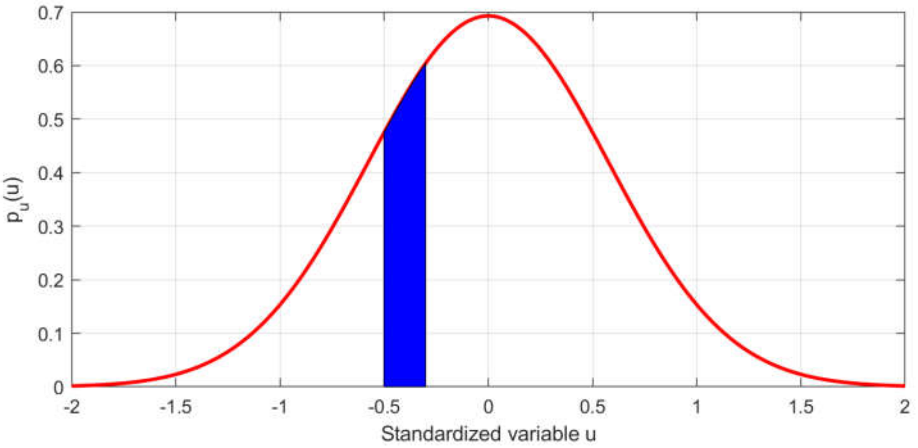


Figure 7 – Probability density function of a normally distributed signal

For a random signal $u(t)$, the expected value μ_u can be computed by

$$\mu_u = E[u] = \int_{-\infty}^{\infty} u P_u(u) du \tag{4.5}$$

and the estimator of the mean value

$$\tilde{u} = \hat{\mu}_u = \frac{1}{N} \sum_{n=0}^{N-1} u(n). \tag{4.6}$$

The variance σ_u^2 can be defined as

$$\sigma_u^2 = E[(u - \tilde{u})^2] = E[u^2] - \tilde{u}^2. \tag{4.7}$$

The root of this expression gives then the standard deviation which indicates the amount of variation. This can be described in other words by saying that the smaller the value for the standard deviation is, the closer are the samples to the mean value.
Another important expression is the correlation function which describes a measure how similar the signal is with a time-shifted version of itself. The cross correlation function R_{wu} hence describes the similarity with the input function $w(t)$, as

$$R_{uu}(\tau) = E[u(t)u(t-\tau)] \qquad R_{wu}(\tau) = E[w(t)u(t-\tau)]. \tag{4.8}$$

With these formulas, the GAUSSian probability distribution can be formulated as

$$p_u(u) = \frac{1}{\sigma_u\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{u-\mu_u}{\sigma_u})^2} \tag{4.9}$$

which can be again seen in figure 7. The last formula in this section is designated to the so called CREST factor which indicates how "peaky" the signal is. Usually it is defined by the peak amplitude divided by the Root Mean Square (RMS) value of the signal. Anyway, in this thesis another definition is used, which can be also found in Brandt [2011] and is computed by the expression

$$c_u = \frac{\max(|u(n)|)}{\sigma_u}. \tag{4.10}$$

For a GAUSSian randomly distributed signal this factor normally yields values between 4-5. In contrast to that, a sine wave has a value of 1.414, and a square wave of 1.

4.2 Signal processing in modelling random processes

To evoke the aim of this section again, the theory of stochastic analysis shall be applied to model a random excitation signal. This theory can now be used to define a so called pseudo-random signal. In order to be able to identify all desired modes clearly, the corresponding frequencies or eigenfrequencies, respectively, have to be excited by the signal. To make sure of this, not randomly distributed numbers shall be created but instead a more detailed one. The sampling frequency f_s is defined through the NYQUIST theorem as

$$f_s = 2\max\{f_n\}k_0 \qquad\qquad \Delta t = \frac{1}{f_s} \tag{4.11}$$

where f_n is the n-th eigenfrequency of the system, k_0 the so called oversampling factor [Murray-Smith, 1995] and Δt the time step size. To choose the sample step size such that each mode can be reproduced by a sufficient amount of samples, the bandwidth B is defined as

$$B := \frac{2\min\{\delta_n\}}{2\pi} \qquad\qquad \delta_n := f_n d_n \tag{4.12}$$

Equation $(4.12)_2$ is the so called decay constant. By choosing B such as it is described by the lowest decay constant, the peakiest mode is regarded. By saying that even this mode is sampled by n samples, Δf is defined by

$$\Delta f := \frac{B}{n} \tag{4.13}$$

This ensures that the resolution of the frequency is fine enough to illustrate all modes properly, see Brandt [2011], where n should be chosen $n \geq 4$. With these definitions, the needed simulation time and the number of samples can be computed by

$$T = N_T \frac{1}{\Delta f} \qquad\qquad N = \frac{T}{\Delta t} \tag{4.14}$$

while N_T defines the number of time series. Multiple time series are required since the excitation is a random process. The multiple averages of independent time series confirm the result statistically. When applying the WELCH method this averaging is automatically involved in the analysis of cross- and auto-power spectra, see Brandt [2011]. Anyway, another reason why the length of the signal is so important is the later introduced modal analysis. The more data available in the end, the better are the modal analysis results, see Peeters and De Roeck [1999]. The objective now is to define an appropriate random excitation signal representative for the random excitation when taxiing with an aircraft over the uneven runway. The general form of a signal within the frequency domain is defined as

$$\hat{u}(f) = p(f) \cdot a \cdot e^{j\varphi(f)} \tag{4.15}$$

where $\varphi(f)$ defines the phase and shall take a random distribution and the variable a a constant amplitude. The function $p(f)$ depends on the frequency and determines the excitations frequency band, like

$$p(f) = \begin{cases} 1 & \text{if } f \in f^* \\ 0 & \text{if } f \notin f^* \end{cases} \tag{4.16}$$

where f^* defines an intervall of the regarded frequencies $f_{min} \leq f^* < f_{max}$. The frequencies f_{min} defines the smallest and f_{max} the largest eigenfrequency to be excited from the signal. In the end the vector $p(f)$ is a purely logical vector and controls whether a frequency is excited $p = 1$ or not $p = 0$. In this way, the frequency band is defined. The signal is then transformed to the time domain with the help of the FOURIER transformation like for $u : \mathbb{C} \to \mathbb{R}$

$$u(t) = \int_{-\infty}^{\infty} \hat{u}(f) e^{2\pi j t f} \tag{4.17}$$

which results in a random excitation function in time domain. The corresponding code to generate a pseudo-random signal as it is introduced in this chapter can be found in appendix B.

5 Improved Runge-Kutta Integration to Solve Nonlinear Systems

In operational modal analysis, where systems are excited by stochastic processes, long data sets can be required for the modal identification process, especially when lightly damped systems with low eigenfrequencies are considered, see equation (4.12). This makes an integration of the equation of motion demanding and expensive in terms of computing time. For this reason, the present chapter is dedicated to an improved integration scheme for nonlinear randomly excited systems. It is clear that due to nonlinearities within the system, no analytic solution can be found and hence a numerical integration is unavoidable. The initial problem is the state space formulation introduced before, which reduces the second order differential equation to first order, while the system of equations doubles its size, as in equation $(2.26)_1$, written as

$$\begin{aligned} \underbrace{\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{f}_{nl} \end{bmatrix}}_{\dot{\mathbf{z}}} &= \begin{bmatrix} \mathbf{Ax}(t) + \mathbf{Bu}(t) + \mathbf{Bu}_{nl}(t) \\ \dot{f}_{nl}(q_r(x(t)), \dot{q}_r(x(t)), f_{nl}) \end{bmatrix} \\ \mathbf{z}(t_{n+1}) &= \mathbf{z}(t_n) + \int_{t_n}^{t_{n+1}} \dot{\mathbf{z}}(t) dt. \end{aligned} \tag{5.1}$$

For finding approximative values for $\mathbf{z}_i$ at time t_i, the initial values are given by $\mathbf{z}(t_0) = \mathbf{z}_0$, $(z, t) \in \mathbb{B}$ and the difference between two time steps is called the time step size $\Delta t = t_{n+1} - t_n$. From this general form, different integration schemes can be derived. Within this thesis, the so called explicit Runge-Kutta method is of great importance, which is derived from equation $(5.1)_2$ and writes

$$\mathbf{z}_{n+1} = \mathbf{z}_n + \frac{\Delta t}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4), \qquad t_{n+1} = t_n + \Delta t, \qquad \mathbf{z}(t_0) = \mathbf{z}_0, \tag{5.2}$$

using for $n_s = 0, 1, 2, ...$ the values for the so called stages, computed by

$$\begin{aligned} \mathbf{k}_1 &= f(t_n, \mathbf{z}_n), \\ \mathbf{k}_2 &= f(t_n + \Delta t/2, \mathbf{z}_n + (\mathbf{k}_1 \Delta t)/2), \\ \mathbf{k}_3 &= f(t_n + \Delta t/2, \mathbf{z}_n + (\mathbf{k}_2 \Delta t)/2), \\ \mathbf{k}_4 &= f(t_n + \Delta t, \mathbf{z}_n + \Delta t \mathbf{k}_3), \end{aligned} \tag{5.3}$$

as example for a fourth order scheme. The order is defined through the number of stages. Increasing the stages at the same time increases the accuracy but simultaneously the computational effort, see Quarteroni et al. [2007]. The evaluation of the stages is visualised in figure 8. For more information concerning the derivation of the method, the reader is referred to the cited literature.

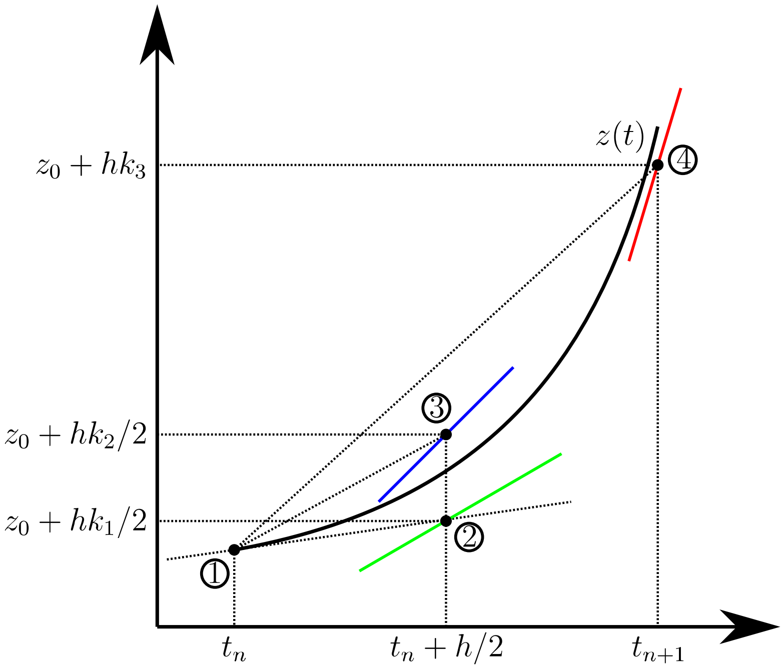


Figure 8 – Schematic representation of a 4th-order Runge-Kutta integration

In general, the weighting coefficients are summarised in the so called Butcher tableau and are denoted as a_{ij}, b_i and c_i for $i = 1 : n_s$. Therewith, the integration scheme can be rewritten in matrix notation and the computation can be carried out in matrix multiplication as well, see Atkinson et al. [2009]. Thus, the integration scheme for the discrete-time state space model as in equation $(2.31)_1$ can be formulated by

$$\begin{aligned} \mathbf{z}_i &= \mathbf{z}(n) + \Delta t \sum_{i-1}^{j=1} a_{ij}\mathbf{k}_{i-1}, \\ \mathbf{k}_i &= \dot{\mathbf{z}}(t = n\Delta t + c_i\Delta t), \\ \mathbf{z}(n+1) &= \mathbf{z}(n) + \Delta t \sum_{i=1}^{n_s} b_i\mathbf{k}_i. \end{aligned} \tag{5.4}$$

Since the aim of this thesis is to use the simulated data which is the system response due to the random excitation, it is also crucial to regard the interpolation of the latter. By starting with Matlab's ODE45 function, one rapidly realises that by interpolating the whole input signal at each step, long simulation times arise. Since the random signal is only available in form of a lookup table, the signal is exclusively accessible at discrete time points. Due to the fact that ODE45 uses adaptive step sizes to increases its stability and accuracy, the random signal has to be interpolated for each step. To overcome this problem, the following interpolation scheme is presented to faster evaluate the corresponding excitation at a certain time step within the integration scheme. The idea is to use a constant spacing between the time steps in combination with a very small sample size to be able to know at any time at which time step the integration scheme is.

As it is said, the random excitation signal is available in form of a so called lookup table.

Therewith, at each time point, randomly distributed samples exists. The next step is, to generate coefficients $\mathbf{c}$ of interpolating order p. These coefficients can be obtained, using the so called VANDERMONDE matrix $\mathbf{V}_{mn} = \alpha_m^{n-1}$ where m is the order of the interpolating polynomial, see Horn and Johnson [1991]. Let s be the number of samples, for $i = 1 : s-1$, the polynomial coefficients for the i-th row of $\mathbf{u}(t)$ can be written as

$$\underbrace{\begin{bmatrix} u_i(n+0) \\ u_i(n+1) \\ u_i(n+2) \\ \vdots \\ u_i(n+p) \end{bmatrix}}_{\mathbf{u}^T_{(n:n+p)}} = \underbrace{\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & \Delta t & \cdots & \Delta t^p \\ 1 & 2\Delta t & \cdots & (2\Delta t)^p \\ \vdots & \vdots & \ddots & \vdots \\ 1 & p\Delta t & \cdots & (p\Delta t)^p \end{bmatrix}}_{\mathbf{V}_p} \underbrace{\begin{bmatrix} c_{i,0}(n) \\ c_{i,1}(n) \\ c_{i,2}(n) \\ \vdots \\ c_{i,p}(n) \end{bmatrix}}_{\mathbf{c}_p}. \tag{5.5}$$

Since both $\mathbf{u}$ and $\mathbf{V}_p$ are known, the coefficients can be computed by rearranging the equation as

$$\mathbf{u}^T(n:n+p) = \mathbf{V}_p\mathbf{c}_p(n) \Rightarrow \mathbf{c}_p^T = \mathbf{u}(n:n+p)\mathbf{V}_p^{-T} \qquad \mathbf{c}_p(n) \in \mathbb{R}^{p\times 1} \tag{5.6}$$

By concatenating the s samples into a matrix, the interpolating polynomial coefficients can be computed within one single matrix $\mathbf{C}_p \in \mathbb{R}^{(s-1)\times p}$. Therewith, $\mathbf{C}_p^T(n) = \mathbf{c}(n)$ are the polynomial coefficients, which can be rewritten in terms of a vectorisation as [Jelicic et al., 2020]

$$\mathbf{c}(n) = \mathbf{u}(n:n+p)\mathbf{V}_p^{-T} = \left(\mathbf{V}_p^{-1} \otimes \mathbf{I}\right)\mathbf{u}(n:n+p) \qquad \mathbf{c}(n) \in \mathbb{R}^{p\times 1}. \tag{5.7}$$

Now, thanks to the equal spacing, for each Runge-Kutta step the corresponding polynomial can be evaluated to obtain the excitation at a specific time point.[3] In the following a benchmark test is shown, comparing the computation of a random signal with 10.000 samples using on the one hand the aforementioned ODE45 and Simulink/ODE4. These two integration algorithms are compared with the improved ones, described in this chapter, called FastRK4 for the RUNGE-KUTTA integration of order four and FastRK8 for the one of order eight.

[3] During the work, a problem occurred which was tried to solve with the help of an higher order integration scheme to obtain more accurate results. This is why also an RK scheme of order eight is introduced and compared, using the same equations as in equation 5.4, only with an extended Butcher tableau.

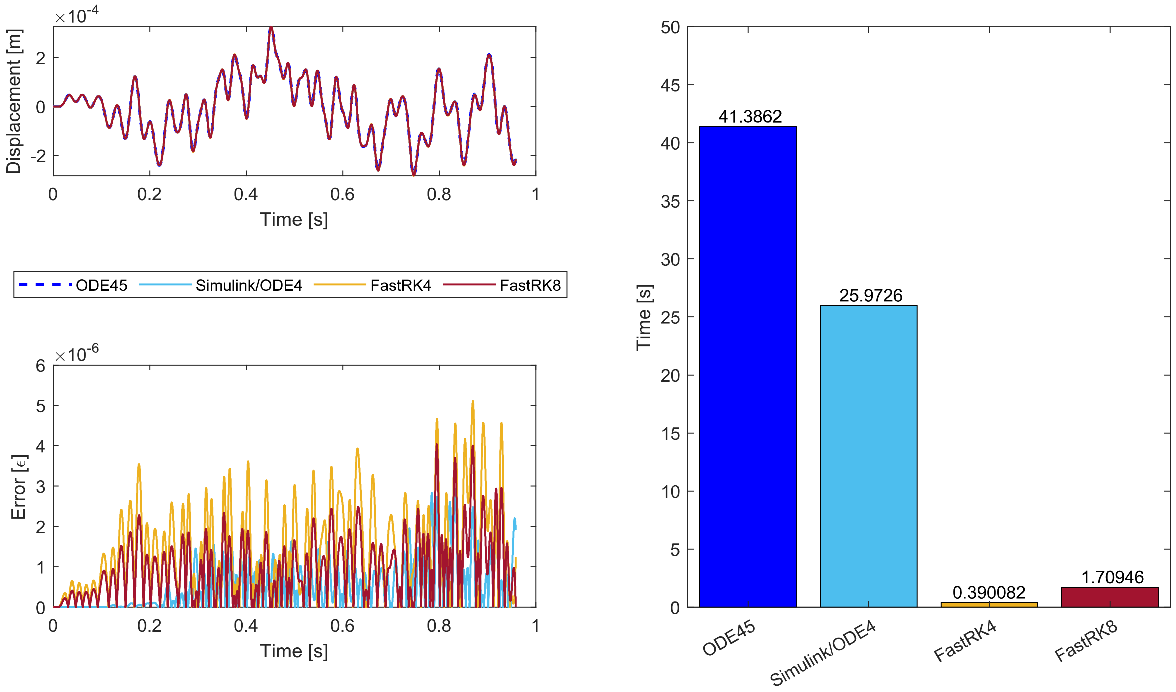


Figure 9 – Benchmark test of integration methods with 10.000 samples; Processor: Intel Core i7-6820HQ CPU @ 2.70GHz; RAM: 16GB

Looking at the results in figure 9 it can be seen that through using the new developed methods, a huge advantage in terms of computation time can be obtained. Additionally it is worth to mention that while using ODE45 in combination with an interpolation method (here interp1) from Matlab, the computation time increases exponentially with increasing number of samples because at each step the whole signal has to be interpolated to obtain the right time point within the random signal.
The presented scheme reduces the computing time significantly, allowing a faster simulation of the nonlinear randomly excited system.

6 Modal Analysis Methods

After the theoretical basis to construct and simulate the substitute model is given, two modal analysis methods shall now be briefly introduced. As the name of this chapter suggests, these methods aim to extract the modal parameters from the dynamic response. Of course, there are many different algorithms. Within this work the focus is on the so called Data-driven SSI and the LSCF analysis.

6.1 Data-driven Stochastic Subspace Identification

The SSI algorithm is a system identification method to analyse data structures with the aim to identify the systems modal parameters. Especially in output-only analysis cases, this method shows its strengths. The algorithm and how it works shall be briefly introduced in this section, with regard to the use case arising in this thesis where the system is excited by a random function and thus outputs a random system response. Following on chapter 4 where the stochastic excitation itself is introduced, this chapter is about how to extract the structural dynamic properties of the system, which is assumed to be linear, by analysing the outputted data, see Brincker and Andersen [2006].
Equation (2.8) represents the equation of motion, modelled through FE approximations. Further it is shown how to rewrite this formulation to the state-space model form. The discrete form of the state space model is hence written down in equation (2.31) and is intended to lay the foundation for further considerations. The vector $\mathbf{u}(k)$ contain the sampled excitations whereas $\mathbf{y}(k)$ represents the measured systems output. Especially in operational modal analysis, it is nearly impossible to measure the random input forces which is why the algorithm to identify the systems properties must be built on solely the measured outputs $\mathbf{y}(k)$. Londono et al. [2004] consider the input terms as random white, zero-mean, stochastic processes, independent of the state vector, noted as $\mathbf{w}$ and $\mathbf{v}$, defined as

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}_d\mathbf{x}(k) + \mathbf{w}(k), \\ \mathbf{y}(k) &= \mathbf{C}_d\mathbf{x}(k) + \mathbf{v}(k). \end{aligned} \tag{6.1}$$

By introducing the convention that $\bullet(k) = \bullet_k$ the discrete form can be rewritten as

$$\begin{aligned} \begin{bmatrix}\mathbf{x}_{k+1} & \mathbf{x}_{k+2} & \cdots & \mathbf{x}_{k+s}\end{bmatrix} &= \mathbf{A}_d \begin{bmatrix}\mathbf{x}_k & \mathbf{x}_{k+1} & \cdots & \mathbf{x}_{k+s-1}\end{bmatrix} + \mathbf{w}_k, \\ \begin{bmatrix}\mathbf{y}_k & \mathbf{y}_{k+1} & \cdots & \mathbf{y}_{k+p-1}\end{bmatrix} &= \mathbf{C}_d \begin{bmatrix}\mathbf{x}_k & \mathbf{x}_{k+1} & \cdots & \mathbf{x}_{k+s-1}\end{bmatrix} + \mathbf{v}_k. \end{aligned} \tag{6.2}$$

Ongoing, s denotes the number of samples. The output data matrix can be defined, which is simply a matrix where each column represents a time sample at the systems output (the column), as

$$\mathbf{Y} = \begin{bmatrix}\mathbf{y}_0 & \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_s\end{bmatrix} \qquad \mathbf{Y} \in \mathbb{R}^{q\times s}, \tag{6.3}$$

where q is the number of considered DOFs. Let i be the user defined number of block rows, the output data, stored in $\mathbf{Y}$ can then be rearranged to construct the so called BLOCK HANKEL matrix $\mathbf{Y}_H \in \mathbb{R}^{(r+l)i\times j}$, by

$$\mathbf{Y}_H = \frac{1}{\sqrt{j}} \begin{bmatrix} \mathbf{y}_0 & \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_{j-1} \\ \mathbf{y}_1 & \mathbf{y}_2 & \mathbf{y}_3 & \cdots & \mathbf{y}_j \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{y}_{i-1} & \mathbf{y}_i & \mathbf{y}_{i+1} & \cdots & \mathbf{y}_{i+j-2} \\ \mathbf{y}_i & \mathbf{y}_{i+1} & \mathbf{y}_{i+2} & \cdots & \mathbf{y}_{i+j-1} \\ \mathbf{y}_{i+1} & \mathbf{y}_{i+2} & \mathbf{y}_{i+3} & \cdots & \mathbf{y}_{i+j} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{y}_{2i-1} & \mathbf{y}_{2i} & \mathbf{y}_{2i+1} & \cdots & \mathbf{y}_{2i+j-2} \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_{(0:i-1)} \\ \mathbf{Y}_{(i:2i-1)} \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_H^- \\ \mathbf{Y}_H^+ \end{bmatrix}. \tag{6.4}$$

$\mathbf{Y}_H$ can be further divided into two sub-matrices, where $\mathbf{Y}_H^- \in \mathbb{R}^{ri \times j}$ represents the "past" and $\mathbf{Y}_H^+ \in \mathbb{R}^{li \times j}$ the "future". The total data shift is $2i$ and thus the number of columns of the matrix can be expressed as $j = s - 2i + 1$. Further, the $2i$ block rows can be distinguished into $i - p$ and $i + p$ rows for the past and the future configuration. For $p = 0$ both the past and the future configuration have the same size. Van Overschee and De Moor [1996] suggest to use $i = 2(maxorder)/q$ where *maxorder* is another user defined parameter.

The next step is to perform a so called projection. This is a crucial step within SSI. If more profound informations are sought, the reader is referred to Van Overschee and De Moor [1996]. In general, this operation is of a mathematically geometric nature. Here, since stochastic processes are regarded, the projection defines a conditional mean. Hence the idea is to retain informations from the past to predict the future. The projection can be written in two different manners, whereas the first one is the correct definition and the latter a series of matrix multiplication. Thus, $\mathbf{O}$ defines the stacked free decay of the system and can be computed with

$$\mathbf{O} = E\left(\mathbf{Y}_H^+ \setminus \mathbf{Y}_H^+\right) = \mathbf{Y}_H^+ (\mathbf{Y}_H^+)^T \left(\mathbf{Y}_H^- (\mathbf{Y}_H^-)^T\right)^\dagger \mathbf{Y}_H^- \quad \mathbf{O} \in \mathbb{R}^{(i+p)q \times j} \tag{6.5}$$

The symbol † designates the MOORE pseudo-inverse. The next step is to perform a QR-decomposition on matrix $\mathbf{Y}_H$ which writes

$$\mathbf{Y}_H = \begin{bmatrix} \mathbf{Y}_H^- \\ \mathbf{Y}_H^+ \end{bmatrix} = \mathbf{R}\mathbf{Q}^T \tag{6.6}$$

$$= \begin{bmatrix} \mathbf{R}_{11} & & & \\ \mathbf{R}_{21} & \mathbf{R}_{22} & & \\ \mathbf{R}_{31} & \mathbf{R}_{32} & \mathbf{R}_{33} & \\ \mathbf{R}_{41} & \mathbf{R}_{42} & \mathbf{R}_{43} & \mathbf{R}_{44} \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1^T \\ \mathbf{Q}_2^T \\ \mathbf{Q}_3^T \\ \mathbf{Q}_4^T \end{bmatrix} \tag{6.7}$$

with the properties

$$\mathbf{Q}^T\mathbf{Q} = \mathbf{Q}\mathbf{Q}^T = \mathbf{I} \qquad \mathbf{R} \in \mathbb{R}^{(i-q)i \times j} \qquad \mathbf{Q} \in \mathbb{R}^{j \times j} \tag{6.8}$$

The idea is now to retain all information from the past that is useful to predict the future. By inserting equation (6.6) into equation (6.5) the projection reduces the data to

$$\mathbf{O}_i = \begin{bmatrix} \mathbf{R}_{21} \\ \mathbf{R}_{31} \\ \mathbf{R}_{41} \end{bmatrix} \mathbf{Q}_1^T \qquad\qquad \mathbf{O}_i \in \mathbb{R}^{li \times j} \tag{6.9}$$

The matrix $\mathbf{O}_i$ in turn can be defined more precisely through the observability matrix $\mathbf{\Gamma}$ and the initial conditions, also called the KALMAN state matrix $\hat{\mathbf{X}}$. The KALMAN filter states are used to produce an optimal prediction for the state vector $\hat{\mathbf{X}}_{i+1}$. This is done by using the so called observations up to time i, the system matrices which are available up to then and the noise covariance, due to the assumption that the system is excited randomly (zero mean, white noise). When the initial state estimate $\hat{\mathbf{x}}_0 = \mathbf{0}$, the covariance $\mathbf{P} = E[\hat{\mathbf{x}}_0\hat{\mathbf{x}}_0^T]$ and the output measurements $[\mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2, ...]$ are given, the state vector can be computed for every time point [Peeters and De Roeck, 1999] using the following relationships

$$\begin{aligned} \hat{\mathbf{x}}_{i+1} &= \mathbf{A}\hat{\mathbf{x}}_i + \mathbf{K}_i(\mathbf{y}_i - \mathbf{C}\hat{\mathbf{x}}_i) \\ \mathbf{K}_i &= \left(\mathbf{G} - \mathbf{A}\mathbf{P}_i\mathbf{C}^T\right)\left(\mathbf{\Lambda} - \mathbf{C}\mathbf{P}_i\mathbf{C}^T\right)^{-1} \\ \mathbf{P}_{i+1} &= \mathbf{A}\mathbf{P}_i\mathbf{A}^T + \left(\mathbf{G} - \mathbf{A}\mathbf{P}_i\mathbf{C}^T\right)\left(\mathbf{\Lambda}_0 - \mathbf{C}\mathbf{P}_i\mathbf{C}^T\right)^{-1}\left(\mathbf{G} - \mathbf{A}\mathbf{P}_i\mathbf{C}^T\right)^T \end{aligned} \tag{6.10}$$

where $\mathbf{G}_0$ is the state output covariance matrix and $\mathbf{\Lambda}_0$ the output covariance matrix. The KALMAN filter states are further defined by

$$\hat{\mathbf{X}}_i = \begin{bmatrix} \hat{\mathbf{x}}_i & \hat{\mathbf{x}}_{i+1} & \cdots & \hat{\mathbf{x}}_{i+j-1} \end{bmatrix} \qquad\qquad \hat{\mathbf{X}}_i \in \mathbb{R}^{n \times j} \tag{6.11}$$

As it is mentioned before, the matrix $\mathbf{O}$ can also be described by the observability matrix $\mathbf{\Gamma}$ and the KALMAN filter states $\hat{\mathbf{X}}_{i+1}$. For $i = 1 : j$ this writes

$$\mathbf{O}_i = \mathbf{\Gamma}_i\hat{\mathbf{X}}_i. \tag{6.12}$$

It is implied that the observability matrix, as the name suggests, is observable, which means that the matrices $\mathbf{A}_d$ and $\mathbf{C}_d$ can be observed in the output with respect to equation (6.2) and is defined as

$$\mathbf{\Gamma}_s = \begin{bmatrix} \mathbf{C}\mathbf{A}_d^0 & \mathbf{C}\mathbf{A}_d^1 & \mathbf{C}\mathbf{A}_d^2 & \cdots & \mathbf{C}\mathbf{A}_d^{i-1} \end{bmatrix}^T. \tag{6.13}$$

Since neither the observability matrix, nor the initial conditions of each stage are known, Singular Value Decomposition (SVD) is performed on the matrix $\mathbf{O}$ to obtain estimates, as

$$\mathbf{O}_i = \mathbf{U}\mathbf{S}\mathbf{V}^T. \tag{6.14}$$

The SVD is then used to define the not unique estimates as

$$\hat{\mathbf{\Gamma}}_i = \mathbf{U}\mathbf{S}^{\frac{1}{2}} \qquad\qquad \hat{\mathbf{X}}_i = \mathbf{S}^{\frac{1}{2}}\mathbf{V}^T \tag{6.15}$$

One important step is the following. If one block row is removed from the top of $\mathbf{O}$ and one block row from the bottom of $\mathbf{\Gamma}_i$, the KALMAN state at the next time lag can be computed like

$$\mathbf{O}_{i+1} = \mathbf{\Gamma}_{i-1}\hat{\mathbf{X}}_{i+1}. \tag{6.16}$$

If this procedure is continued, all KALMAN states can be computed by

$$\hat{\mathbf{X}}_{i+1} = \mathbf{\Gamma}_{i-1}^{\dagger}\mathbf{O}_{i-1}. \tag{6.17}$$

The next step is to estimate the system matrices $\mathbf{A}_d$. By using equation (6.17) and evoking again equation $(6.2)_1$, the system matrices can be computed by

$$\hat{\mathbf{X}}_{i+1} = \hat{\mathbf{A}}_d\hat{\mathbf{X}}_i \qquad \Rightarrow \qquad \hat{\mathbf{A}}_d = \hat{\mathbf{X}}_{i+1}\hat{\mathbf{X}}_i^{\dagger} \tag{6.18}$$

From the system matrices the modal parameter can be extracted, so the eigenfrequencies, damping ratios and eigenvectors. Therefore, an numerical eigenvalue decomposition is applied to matrix $\mathbf{A}_d$, denoted by

$$\hat{\mathbf{A}}_d = \mathbf{\Phi\Lambda\Phi}^{-1} \Rightarrow \hat{\mathbf{A}}_d\boldsymbol{\phi}_i = \mu_i\boldsymbol{\phi}_i \qquad \mathbf{\Lambda} = \mathrm{diag}(\mu_i) \tag{6.19}$$

where $\mathbf{\Phi}$ is the modal matrix, containing all modes of the system and $\mathbf{\Lambda}$ a diagonal matrix with the discrete-time system poles μ_i which can be computed by solving the eigenvalue problem. The last step is to employ the following relationships, to compute the desired values

$$\lambda_i = \frac{\ln(\mu_i)}{\Delta t} \qquad f_i = \frac{|\lambda_i|}{2\pi} \qquad d_i = -\frac{\mathrm{Re}(\lambda_i)}{|\lambda_i|} \tag{6.20}$$

where λ_i are the continuous-time poles, f_i are the eigenfrequencies and d_i the damping ratios, see Van Overschee and De Moor [1996].

In the end, the selection of the subspaces n is important in terms of determining the modal parameter. The size of the subspace defines the size of the state matrix $\mathbf{A}_d$ and therefore the number of modes of the model. Since all modal parameter occur in complex conjugated pairs, n has to be chosen twice as big as the regarded frequency band, to identify all modes properly. Therefore, a stabilisation diagram is used together with the singular values to determine an optimal subspace number, see Soal [2018]. A basic implementation of data-driven SSI can be found in appendix C.

6.2 Least Squares Complex Frequency-domain modal analysis

Whereas SSI belongs to the group of OMA analysis techniques using time domain responses as input, the LSCF method works in frequency domain within EMA. It uses the frequency response functions as an input and thus uses both, output and input for the modal identification. This method can handle Multiple Input Multiple Output (MIMO) Linear Time Invariant (LTI) systems. Generally speaking, this algorithm estimates polynomial coefficients to extract the modal parameters from that. By regarding the poles and modal participation, it uses the denominator to reconstruct the transfer function. Consider a MIMO LTI system like

$$H(j\Omega) = \frac{\sum_{i=0}^{M} b_{kl}(j\Omega)^i}{\sum_{i=0}^{M} a_i(j\Omega)^i} \qquad H \in \mathbb{C}^{N_{input} \times N_{output}} \tag{6.21}$$

where $H(j\Omega)$ is the frequency response function, M is the user defined model order and a and b the polynomial coefficients. By bringing everything over to the left hand side of the equation, this equation yields

$$H(j\Omega)\sum_{i=0}^{M} a_i(j\Omega)^i - \sum_{i=0}^{M} b_{kl}(j\Omega)^i = \epsilon, \tag{6.22}$$

where ϵ denotes the value to be minimised. The coefficients of the polynomial are unknown and must be determined. This often leads to a non-linear optimisation problem that aims to minimise the mean value of ϵ. Therefore, all parameters are stored into the unknown vector $\mathbf{\Theta}$, like

$$\mathbf{\Theta} = \begin{bmatrix} \cdots & b_{kl,i} & \cdots & | & a_0 & \cdots & a_M \end{bmatrix}^T \tag{6.23}$$

leading to the following expression which needs to be minimised

$$\text{Error}(\mathbf{\Theta}) = \sum_{k,l,n} \epsilon^2 \rightarrow \text{min.} \tag{6.24}$$

This can be accomplished by following TAYLOR series linearisation with the JACOBIAN matrix like

$$\mathbf{J\Theta} = \mathbf{0} \rightarrow \frac{\partial \boldsymbol{\epsilon}}{\partial \mathbf{\Theta}}\mathbf{\Theta} = \mathbf{0}. \tag{6.25}$$

After finding the coefficients, the eigenvalues and damping ratios can be obtained from the roots of the dominator polynomial [Böswald, 2016]

$$\sum_{i=0}^{M} a_i(j\Omega)^i = 0 \rightarrow \lambda_i = \Re(\lambda_i) + j\Im(\lambda_i) = -D_i\omega_i + j\omega_i\sqrt{1-D_i^2} \quad i = 1, 2, ..., M \tag{6.26}$$

This section gives just a fundamental introduction into the wide field of LSCF estimators. For further information the reader is referred to Böswald et al. [2006] and Guillaume et al. [2003].

7 Example Problem: Vibration Analysis of an Aircraft During Taxiing

Within this chapter, a substitute model of an aircraft shall be constructed. This model shall retain the essential effect of coupled vibration if the linear airframe is excited through the nonlinear landing gear and shall be numerical efficient. The symmetric property shall be used to only consider one half of the aircraft in accordance with additional boundary conditions. This will then be used to run various simulations and the results can be subjected to the actual examinations. To do so, the present section is divided into three parts. Within the first part, the model itself is constructed and the parameters are defined. Afterwards, the random excitation is generated, using the stochastic fundamentals and signal processing introduced in chapter 4. The third section then defines and shows different excitation levels to uncover the nonlinearity of the system and to visualise the main motivation of this work.

7.1 Building the substitute model

As it is mentioned in the introductory chapter, this thesis aims to investigate the behaviour of the nonlinear landing gear, acting on linear structures. This is represented here through the wing in operation, i.e. under stochastic excitation. The structure discussed in this thesis is one half of the Airbus A320 or DLR D150 configuration, respectively. A representation of such an aircraft is shown in figure 10. This aircraft is then modelled, using the already introduced fundamentals of structural mechanics. To construct the system matrices, the FEM is used. The desired substitute model is visualised in figure 11. As can be seen, the symmetry of the aircraft has been exploited to increase the numerical effectiveness of the substitute model. This means that only one half is modelled with appropriate boundary conditions at the symmetry plane. Furthermore, the aircraft is represented through a cantilever beam, additional masses for the fuselage and the engine, the nonlinear friction model discretised with the Valanis model, two rods under and above the nonlinear branch (piston and fitting) as well as the tire with its mass, stiffness and damping values, see figure 11.

Figure 10 – Airbus A320neo (from: airbus.com)

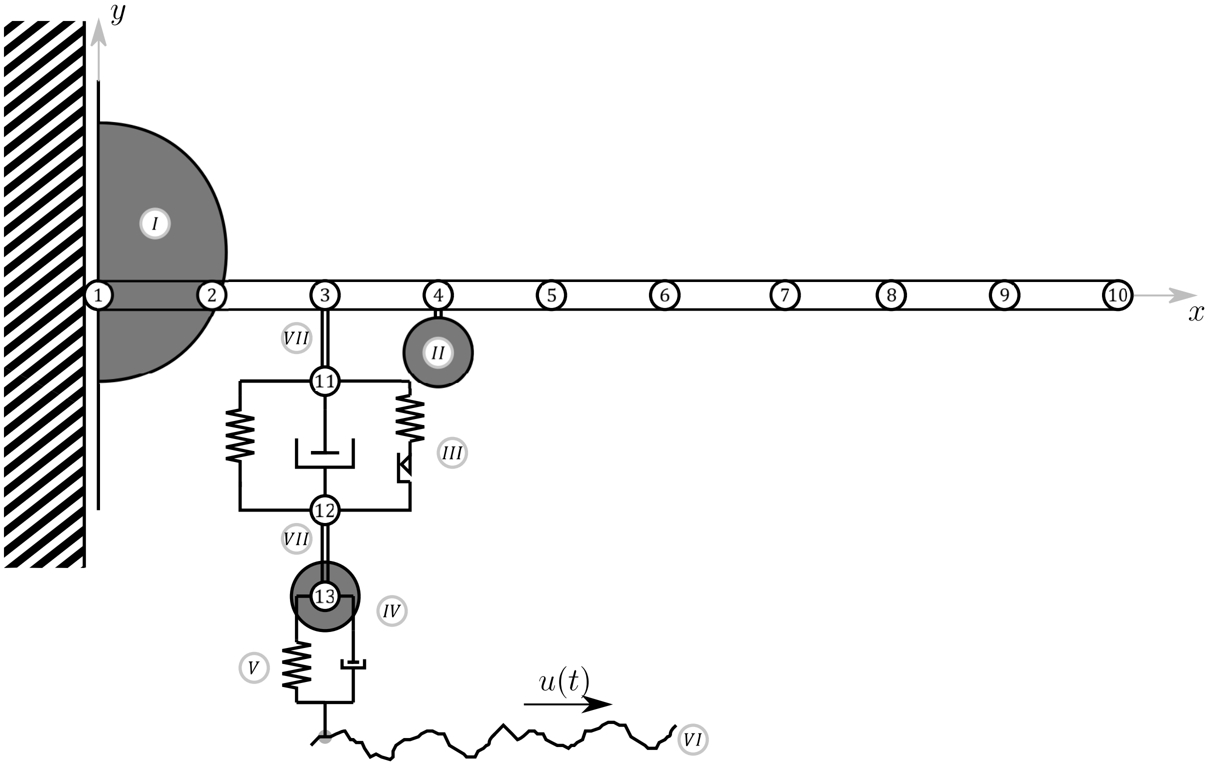


Figure 11 – Discretised aircraft, (*I*) fuselage, (*II*) engine, (*III*) non-linear stick-slip element described by the Valanis model, (*IV*) tire, (*V*) stiffness and damping of the tire, (*VI*) randomly distributed displacement, (*VII*) rods representing landing gear elements, (1-13) indicate the nodes

Ongoing, the particular parts shall be examined in more detail, starting with the structure of the wing. Rossow et al. [2014] state that there are different structural concepts of aircraft wings. Since in aviation, the weight and strength of parts are of great importance, hollow profiles are used to save weight without reducing the strength. This is illustrated in figure 12a. In order to approximate this within the model, the well known box profile is used. Since the wing is tapered, the depth decreases towards the tip. Therefore, a common parameter in aircraft design is used, the so called Mean Aerodynamic Chord (MAC). This parameter defines the mean value of the depth of the aircraft wing. By further investigating the wings profile, it becomes clear that the box profile of the beam has not the same depth as the wing itself. Approximately 50% of its depth is aerodynamic fairing. By saying that the wall thickness is 10% of the beams dimensions, the depth of the beam can be defined as

$$B = MAC \cdot 0.5 \qquad\qquad b = B \cdot 0.9 \tag{7.1}$$

Despite the fact that the height of the wing also decreases towards the tip, it is assumed to be constant within the model and is defined as H, $h = 0.9H$. The other needed material parameters can be taken from the documentary of the aircraft, see Airbus [2005].
The comparison between the actual wing design and the modelled one can be seen in figure 12.

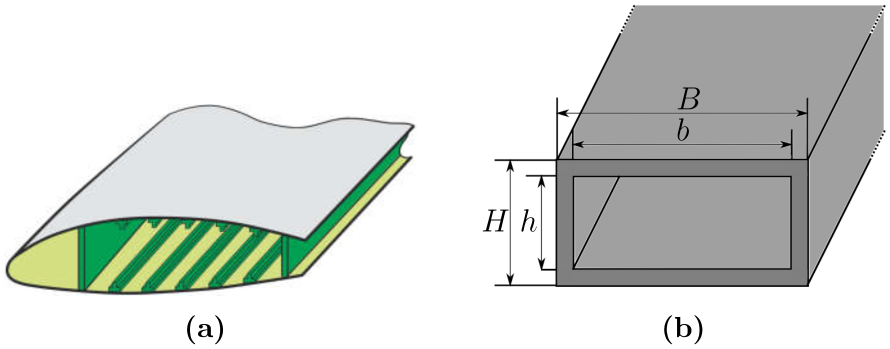


Figure 12 – Cross section of an aircraft wing (a)[Rossow et al., 2014] and cross section of the modelled hollow cantilever beam (b)

By looking at the cross section of an aircraft wing, see figure 12, it can be seen that it can be discretised by using a finite number of an hollow cantilever beam. As said, the mathematical description is based on the FEM. The system matrices for each element of such a cantilever beam are defined as

$$\mathbf{K}_c^e = \frac{E_c I_c}{l_c^3} \begin{bmatrix} 12 & 6l_c & -12 & 6l_c \\ & 4l_c^2 & -6l_c & 2l_c^2 \\ & & 12 & -6l_c \\ \text{sym} & & & 4l_c^2 \end{bmatrix} \qquad \mathbf{M}_c^e = \frac{\rho_c A_c l_c}{420} \begin{bmatrix} 156 & 22l_c & 54 & -13l_c \\ & 4l_c^2 & 13l_c & -3l_c^2 \\ & & 156 & -22l_c \\ \text{sym} & & & 4l_c^2 \end{bmatrix} \tag{7.2}$$

where E_c denotes the E-Module of the beam material, l_c represents the length of one element, I_c is the geometrical moment of inertia, ρ_c is the density and A_c the cross section area. The geometrical moment of area is computed by

$$I_c = \frac{1}{12}(BH^3 - bh^3) \tag{7.3}$$

where B and H are the outer and b and h the inner dimension of the profile. The derivation of these system matrices can be found in Link [2014]. Each node has two DoF, a translational and a rotational one. The displacement vector corresponding to the element matrices in equation (7.2) writes

$$\mathbf{q}_c^e = \begin{bmatrix} u_i & \varphi_i & u_j & \varphi_j \end{bmatrix}^T := \begin{bmatrix} q_{i,x} & q_{i,\varphi} & q_{j,x} & q_{j,\varphi} \end{bmatrix}^T . \tag{7.4}$$

The wing is discretised with a number of elements of $n_e = 10$. Therewith the length of the elements is computes by $l_c = L/n_e$. The needed measurements are again taken from the Airbus [2005] report. The used parameter are noted in the subsequent table

L	17	m	Overall length
B	1.55	m	Outer width
H	0.465	m	Outer height
b	1.345	m	Inner width
h	0.4185	m	Inner height
MAC	3.1	-	Mean Aerodynamic Chord Length
E	$7.3 \cdot 10^7$	N/m^2	E-Module
A	0.1369	m^2	For hollow profile
ρ	$1.8226 \cdot 10^3$	kg/m^3	Density
m	4243	kg	Overall mass wing
I	0.0045	m^4	Geometrical moment of inertia

It is assumed that the material for the wing will be DurAlu. Moreover, additional masses are added which represent the weight of the turbine attached to the wing and the fuselage of the aircraft. These masses can be simply added to the correct entry within the mass matrix. Since the center of gravity of the turbine differs from the node of the wing, the theorem of Huygens-Steiner must be taken into account, which writes

$$\mathbf{J}^{(A)} = \mathbf{J}^{(B)} + m[(\mathbf{r}_{AB} \cdot \mathbf{r}_{AB})\mathbf{I} - \mathbf{r}_{AB} \otimes \mathbf{r}_{AB}] \tag{7.5}$$

where $\mathbf{J}$ is the moment of inertia, $\mathbf{r}_{AB}$ is the vector describing the displacement of the axes crossing the center of gravity and its actual point of reference. $\mathbf{I}$ is the identity matrix. The additional masses in numbers are

Fuselage	1800	kg
Engine	3100	kg
Eccentricity	1	m

The damping of the beam is constructed with the help of equation (2.22) in section 2.3. Therefore, a separate eigenvalue and eigenvector analysis has to be executed, regarding solely the hollow beam. The modal damping of the system is described by an empirical value.

Consecutively, the landing gear shall be modelled. As illustrated in figure 11, the gear consists of two rods, one end is attached to the wing, the other one to the wheel. The two rods are connected with a spring, damper, slider branch, which holds the nonlinearity in form of a Valanis friction model, introduced in chapter 3. The element matrices of the rods are given by

$$\mathbf{M}_r^e = \rho_r A_r l_r \begin{bmatrix} 1/3 & 1/6 \\ 1/6 & 1/3 \end{bmatrix} \qquad \mathbf{K}_r^e = \frac{E_r A_r}{l_r} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \qquad \mathbf{q}_r^e = \begin{bmatrix} q_i \\ q_j \end{bmatrix} \tag{7.6}$$

and the parameters are

Parameter	Upper rod	Lower rod	Unit
L	1.1	1.1	m
E	$2.11 \cdot 10^{11}$	$2.11 \cdot 10^{11}$	N/m^2
A	0.15	0.15	m^2
m	610	287	kg

which are also chosen based on empirical landing gear parameters.
Subsequently, the tire takes the parameters

k_{tire}	$3.13 \cdot 10^6$	N/m
d_{tire}	4500	Nm/s
m_{tire}	202	kg

After constructing the model, a total mass of $m_{total} = 20529$kg for the half aircraft can be computed. The Operating Empty Mass (OEM) of $m_{OEM} = 42000$kg for the aircraft can be taken from the report which supports the correctness of the parameter estimation.
The last crucial part is the VALANIS element. As it is shown in equation (3.9), the VALANIS element is described by four independent parameters, where E_t, E_0 and κ can be estimated through empirical values. The parameter f_c denotes the threshold when the element changes its behaviour from stick to slip. This parameter was adjusted such that when rolling on typical taxiway with given roughness will lead to phases with sticking friction and phases with sliding friction. This is necessary to have the desired nonlinear features in the simulated response data. For the sake of completeness, the parameters used in the following simulations are

Parameter	Value	Unit
E_t	10^6	N/m
E_0	$5 \cdot 10^6$	N/m
f_c	$3 \cdot 10^4$	N
κ	0.9	-

Looking again at figure 11, it is easy to see that the model has 23 DoF, 20 wing and 3 landing gear DoF. With the aforementioned aspects, the mass, stiffness and damping matrix can be constructed and the state space formulation can be defined. As the just named VALANIS model has the same order as the state space model, these two equations can be merged as one system of equation which has to be solved. Therefore equation (5.1) is evoked again

$$\dot{\mathbf{z}} = \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{f}_{nl} \end{bmatrix} = \begin{bmatrix} \mathbf{Ax}(t) + \mathbf{Bu}(t) + \mathbf{Bu}_{nl}(t, f_{nl}) \\ \dot{f}_{nl}(q_r(x(t)), \dot{q}_r(x(t)), f_{nl}) \end{bmatrix} \tag{7.7}$$

After having defined the dynamic system and its mathematical description, the Initial Condition (IC) and Boundary Condition (BC) have to be set. The only BC is found at node number one. In other words, the fuselage can only move up and down but can not rotate, like $q_{1,\varphi}(t) = 0$ and the initial condition is defined as

$$\mathbf{z}_0 = \begin{bmatrix} \mathbf{x}_0 \\ F_{nl,0} \end{bmatrix} = \mathbf{0} \tag{7.8}$$

Thus, all conditions are fulfilled to perform a simulation of the system. Since the IC are zero, the system would not move without external force. Therefore, in the next chapter, the previously mentioned stochastic excitation is generated.

7.2 Modelling of the stochastic excitation

In chapter 4 the theory for stochastic processes and signal processing is briefly introduced. In this section this theory shall now be applied to construct a random excitation function in time domain to model operational loading which is the vertical displacement at the contact point of the tire with the runway. To be able to identify the systems modal parameters properly, the excitation signal must ensure that every eigenfrequency of interest is excited. Therefore the theoretical system, constructed in the previous chapter is used to better understand the systems eigenfrequency to be therewith able to define the random signal. After solving the eigenvalue problem for the linearised Valanis model, such that k_0 and k_1 are active, like

$$\mathrm{Det}(\lambda \mathbf{A}_{lin} + \mathbf{B}_{lin}) = 0. \tag{7.9}$$

a first estimate of the linear eigenfrequencies can be computed. The highest eigenfrequency of the linearised system is at $\max\{\lambda\} = 3318Hz$. Therewith, the sampling frequency can be defined as in equation $(4.11)_1$ and the time step size as in $(4.11)_2$. By also computing the damping ratios, the frequency resolution and consequently the minimum required simulation time can be computed as well. The defined spectrum is illustrated in figure 13. It can be seen that the sampling frequency is very high compared to the highest excited frequency.

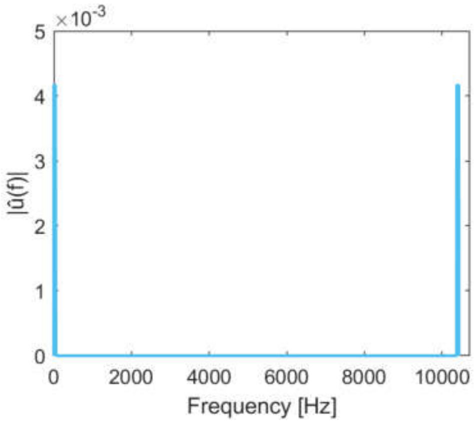


Figure 13 – Spectrum of the random signal

After transforming the signal from frequency to time domain, the whole randomly distributed, zero-mean signal $u(t)$ and its distribution is visualised in figure 14. Additionally, a segment of the time signal is illustrated, showing that the generated pseudo random signal is smooth.

The statistical properties of the signal are $\mu = 3.374 \cdot 10^{-20} \approx 0$, $\sigma = 0.0054$ and the crest factor $C = 4.6358$. These values can be used as a verification that the signal meets all the desired requirements, namely Gaussian, white, zero-mean noise. Besides the excitation as a displacement caused by the uneven pavement in time domain, the distribution of the same is also shown. Regarding the latter in combination with the before mentioned parameter of the standard deviation and mean value, it is proven that the excitation function fulfils the aforementioned requirements. Additionally the normal distribution can be clearly observed.

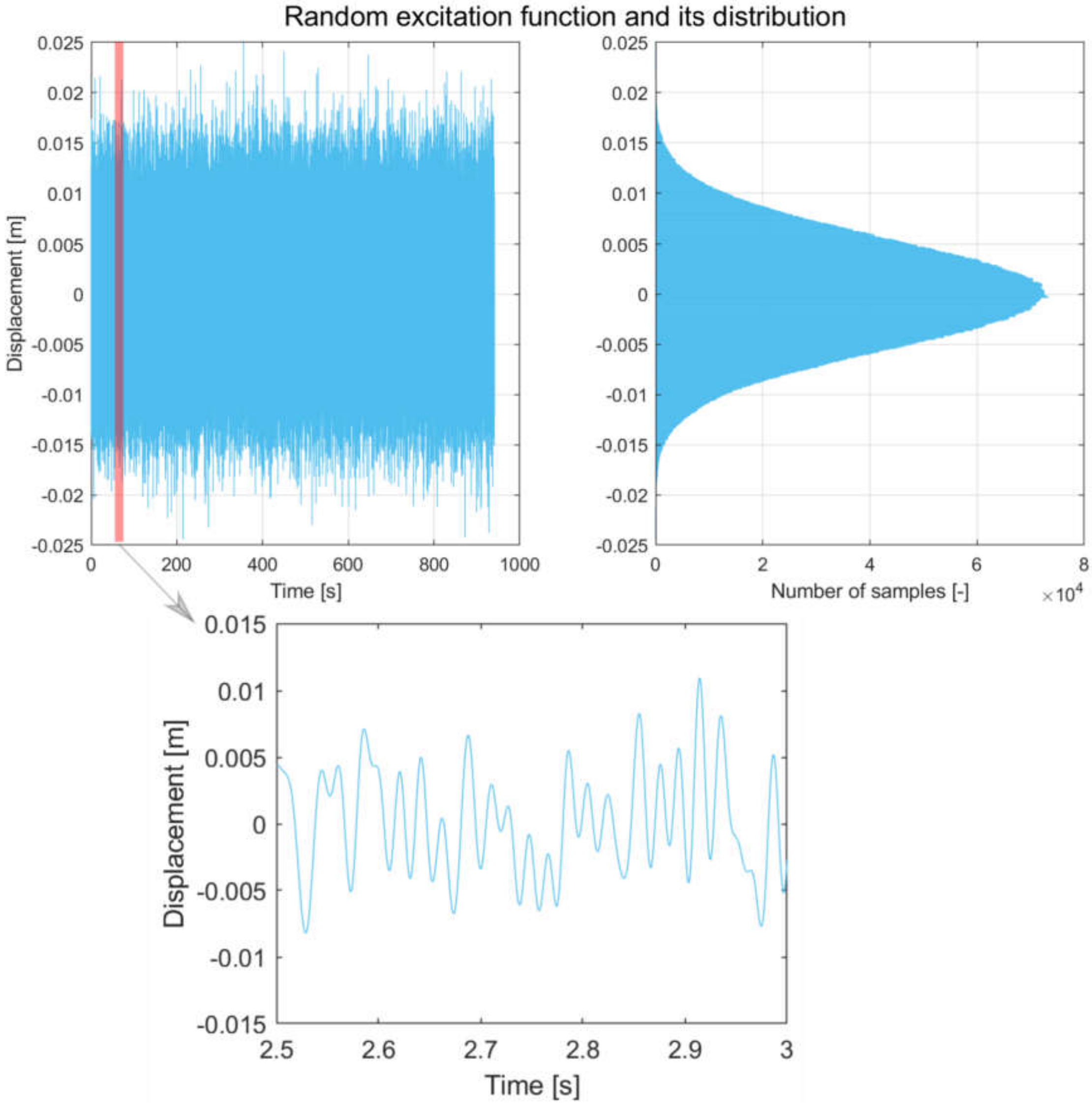


Figure 14 – The modelled random excitation and its normal distribution for $\max\{u(t)\} = 0.025m$

7.3 Simulation results and illustration of the problem

After constructing the model as well as the systems excitation, both can now be merged together to simulate the systems response caused by the excitation. Since the system is nonlinear because of the nonlinear friction element, a numerical integration is used to solve the nonlinear system of equation as equation (3.12). To do so, the improved Runge-Kutta integration scheme, presented in chapter 5, is employed.

Before showing the results, it should be fundamentally recapitulated what this thesis aims to show. Within these first simulations it should be demonstrated that the regarded system behaves nonlinear. In chapter 3 the theory for modelling friction nonlinearities is explained as well as the influence on the modal parameters. To recall the findings again, it is shown that with increasing displacement amplitude the eigenfrequency decreases while the damping ratios increase. This happens due to the change of phase of the Valanis model from stick to slip. In the further course of this chapter this development shall be presented. Therefore, the displacement function $u(t)$ is scaled to different excitation levels to show that the systems modal parameters are dependent on the excitation level. The levels are defined through the maximum value, like

Level	1	2	3	4	5
$\max\{u(t)\}$	0.005m	0.012m	0.025m	0.040m	0.060m

Since the third level reflects the behaviour of the uneven pavement very well, two higher and two lower excitations are chosen. Moreover, the first level is defined such that the excitation is small enough that no slipping phase within the Valanis model occurs and thus a linear system response is achieved.

After simulating the system, for demonstration purposes, figure 15 indicates the hysteresis curve, visualising the force within the nonlinear landing gear or respectively the Valanis model, depending on the relative displacement.

In contrast to that, figure 16 illustrates the hysteresis curves for the next lower and higher level. The change in shape of the curve, showing the different transitions from stick to slip, can be clearly observed.

After simulating the system for each level, applying SSI, computing the estimated Frequency Response Function (FRF) through SSI $H(j\omega)$ and averaging [Pintelon and Schoukens, 2012] the channels with

$$\tilde{H} = \frac{1}{N_{states}} \left(\sum_{i=1}^{N_{states}} |H(j\omega)| \right) + \frac{1}{N_{states}} \left(\sum_{i=1}^{N_{states}} \Im(H(j\omega)) \right), \tag{7.10}$$

the FRF for the levels can be visualised in figure 17. Additionally, the mode shapes for the six shown modes are illustrated in order to demonstrate the movement of each mode at the corresponding eigenfrequency.

The averaging reduces the impact of the disturbing noise and results in a smoother estimate, see Schoukens et al. [2014]. Within this figure, two phenomena can be observed. The first one is that not every mode is affected by the nonlinearity. This is because not in every mode the nonlinear element of the landing gear has an impact. A perfect example for that is mode

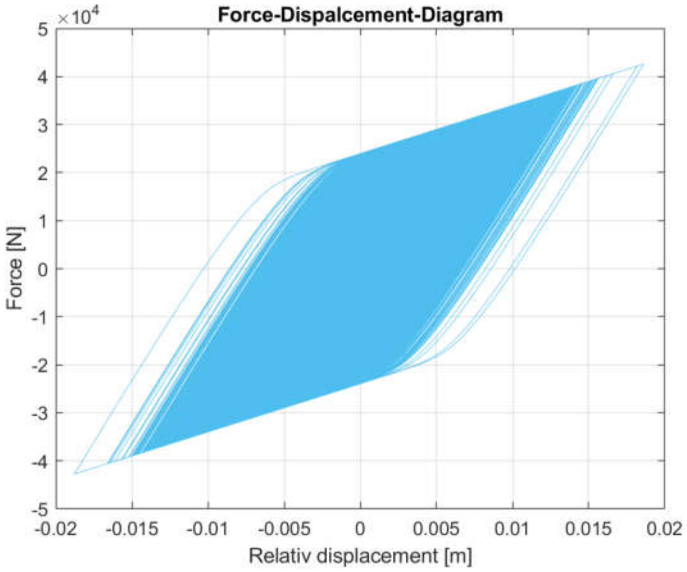
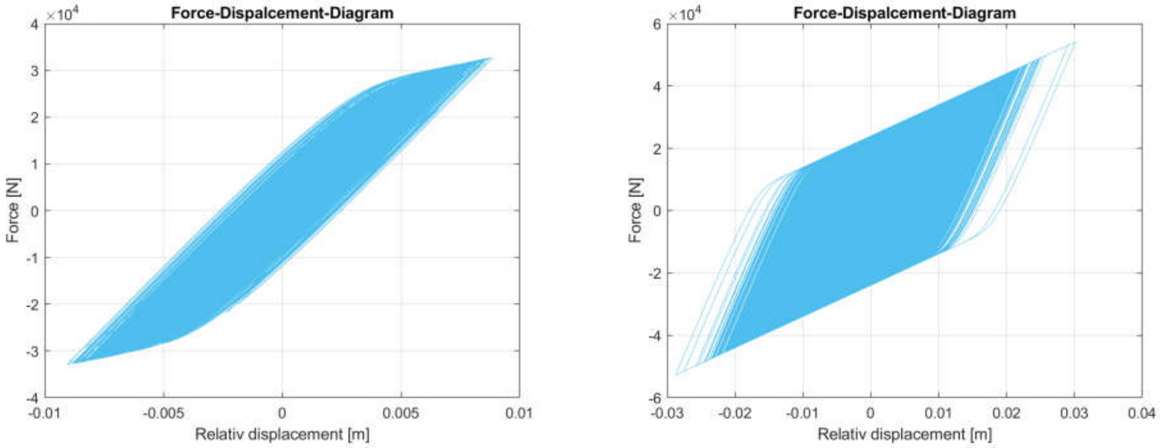


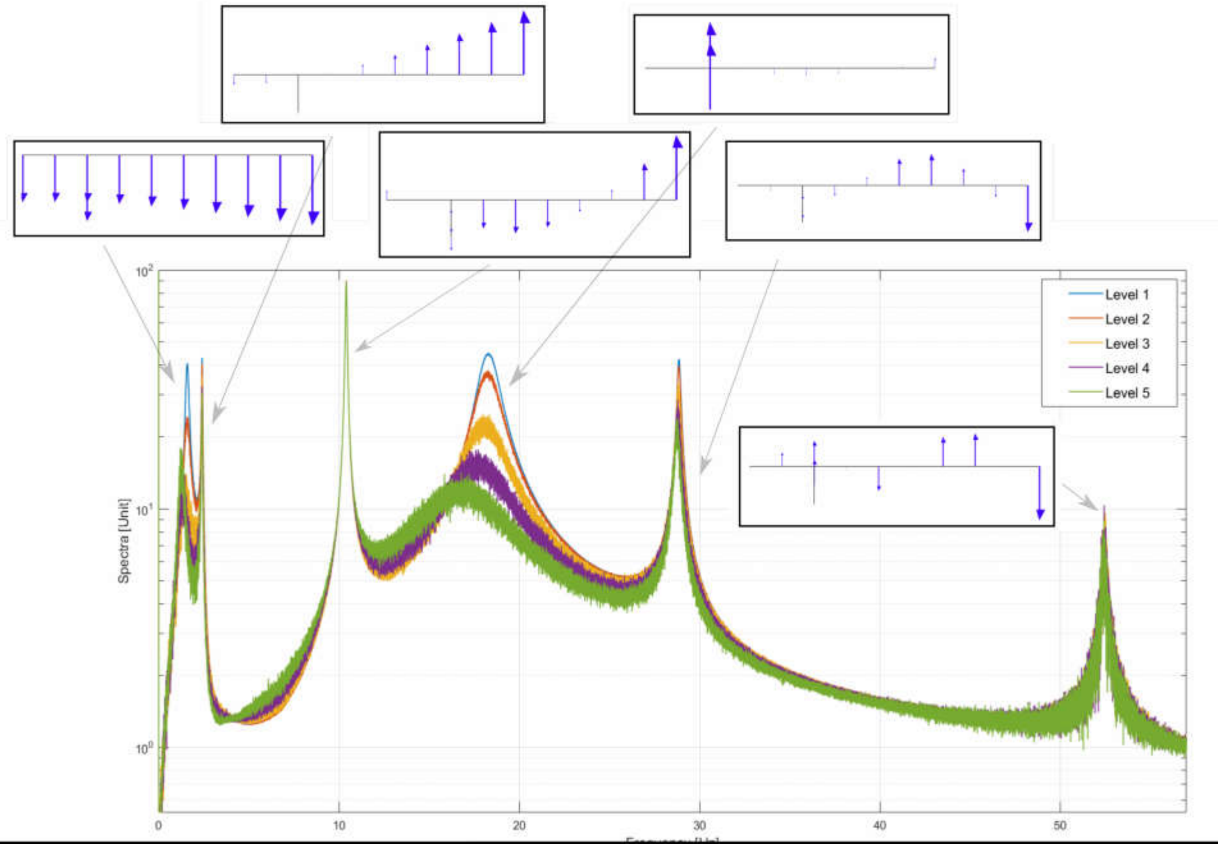
Figure 15 – Modelled hysteresis curve through the Valanis model at excitation level $\max\{u(t)\} = 0.025m$



(a) Modelled hysteresis curve through the Valanis model at excitation level $\max\{u(t)\} = 0.012m$

(b) Modelled hysteresis curve through the Valanis model at excitation level $\max\{u(t)\} = 0.040m$

Figure 16 – Hysteresis curves for different excitation levels



[redacted]

Figure 17 – Averaged Frequency Response Functions and corresponding mode shapes

three around 10Hz. The different levels match the others perfectly which is an indicator for the absence of the landing gear. Another important aspect is the increasing noise with increasing excitation level. This is caused by the increasing impact of the nonlinear subsystem. Schoukens et al. [2000] found the same results by saying that the eigenfrequency shifts with changing excitation level and the measurement becomes more noisy. These findings can also be explained by looking again at the mode shapes in figure 17 where it can be seen that especially in modes three and five the landing gear does not move.
To better visualise this behaviour, the so called nonlinearity plot is introduced, showing the eigenfrequency and damping ratio of a specific mode depending on the excitation level, see figure 18. This plot is essential for the further course of this work since it is a good indicator how well the nonlinearity can be reproduced by the developed methods.

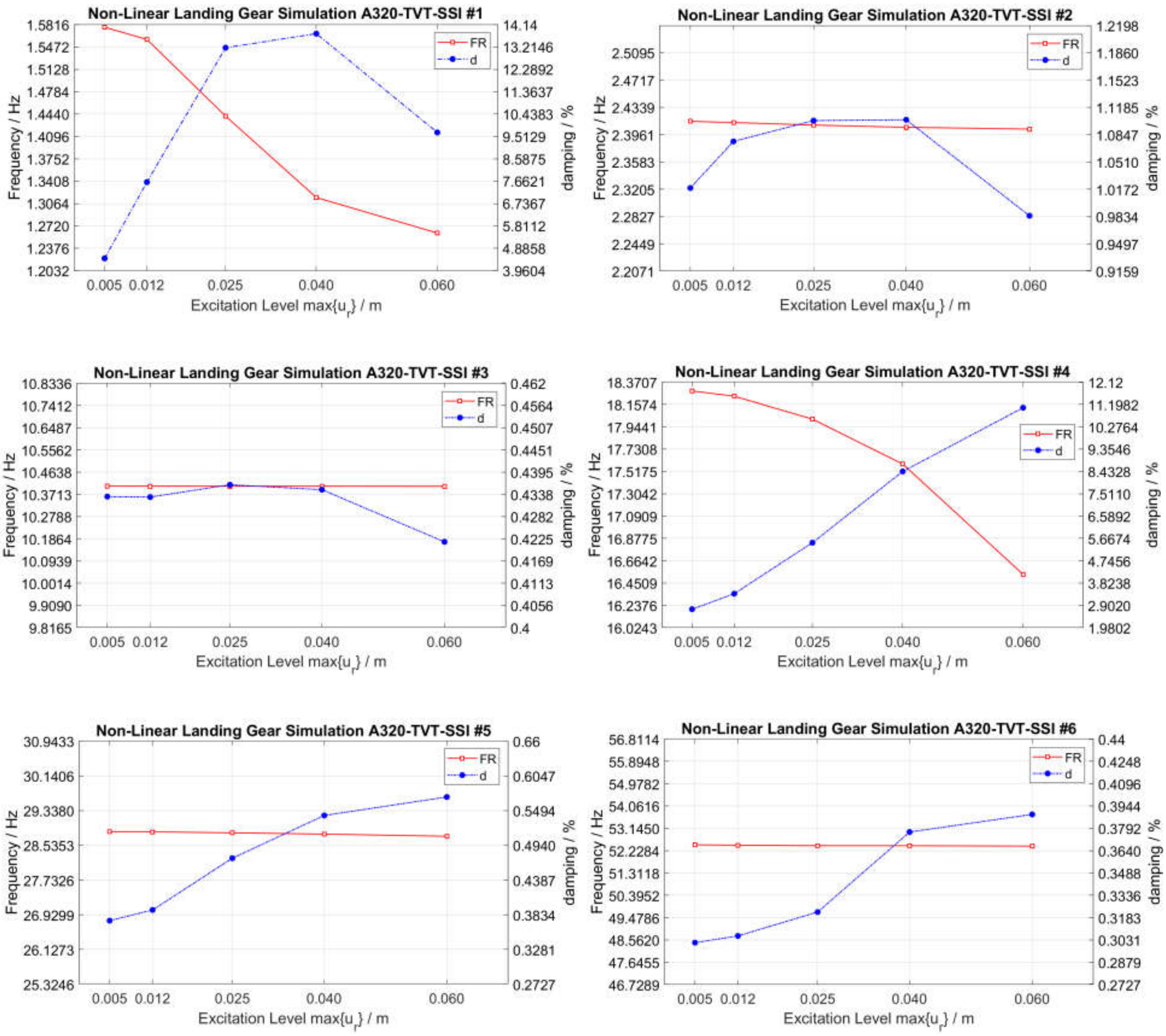


Figure 18 – Nonlinearity plots as problem description

Within this chapter the influence of nonlinearities are shown, which proves the prediction derived in the theory chapter. The problem that arises is, that within the introduced TVT, it is hard to obtain test data for different excitation levels, and thus a method for identifying this nonlinear behaviour of the system is lack. In the following, different approaches are developed and investigated to obtain the changing eigenfrequencies with changing excitation level. To identify the nonlinear behaviour here the five different simulations were used which would correspond to different TVT test under different excitation levels. This is however not possible which is why the goal is to use just one simulation dataset in combination with data analysis methods to extract the nonlinear modal parameters out of this single simulation run.

8 Nonlinear System Identification Methods

In this chapter, different methods will be developed and investigated to identify the previously described and demonstrated nonlinearity. As it is mentioned before, within the TVT it is hard to obtain different excitation levels to identify the changing modal parameters. Even though it is not possible to control the level of stochastic excitation, it can be seen that within these stochastic signals time segments exist with rather low and time segments with rather high excitation level. This shall be exploited for characterisation of the nonlinearity from a single excitation run. Thus, it is the aim of this chapter to use data analysis methods and stochastic analysis to further develop the already established OMA techniques for linear systems.

Therefore, a preliminary investigation is performed to validate the simulated results. Afterwards, three methods are regarded for nonlinear operational modal analysis.

8.1 Preliminary investigation: Decoupling the non-linear and linear subsystems

Within this preliminary investigation it is the aim to proof that by decoupling the linear and the nonlinear subsystems, the analytical modal parameters of the linear cantilever beam can be identified from the dynamic response. Therefore, the system is cut free between the nodes three and eleven, see again figure 11. With the help of the relative displacement between the nodes 3 and 11 and the stiffness of the upper landing gear rod, the element force can be computed that is transmitted into the wing. Figure 19 shows a schematic drawing of this test. Now, the linear part is separated from the nonlinear part. The forces generated by the nonlinear landing gear can be considered as excitation forces to the linear part and EMA can be applied to identify the linear subsystem.

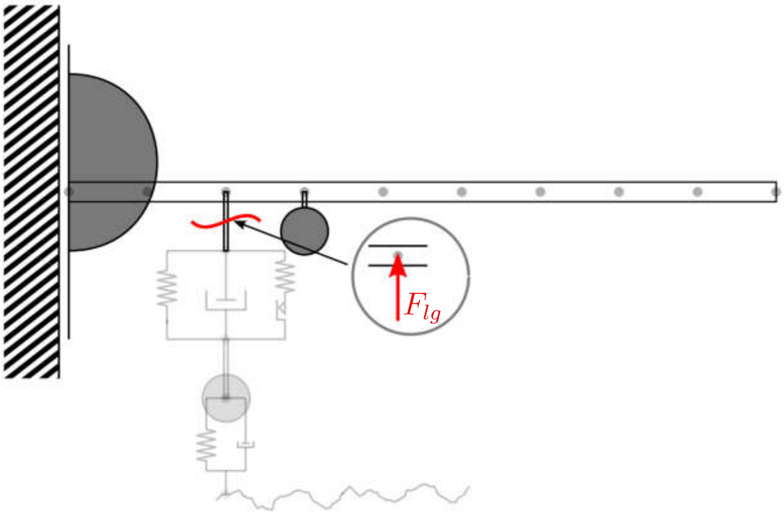


Figure 19 – Decoupled system

To do so, the following procedure is employed. First of all, the force F_{lg} is computed by using the response data of node 3 and 11 and the known stiffness parameter of the upper part of the landing gear. This force is then used as the excitation of the system. In combination with the simulated displacement of all nodes of the wing, LSCF is applied, resulting in the before mentioned stabilisation diagram as in figure 20. By considering figure 21 it is evident that the eigenfrequencies of the linear cantilever beam can be reproduced with high accuracy. Within the nonlinearity plots no dependence of eigenfrequencies or damping ratios on excitation levels is seen which means that the nonlinear part of the system no longer has any influence on the modal parameter of the linear subpart.

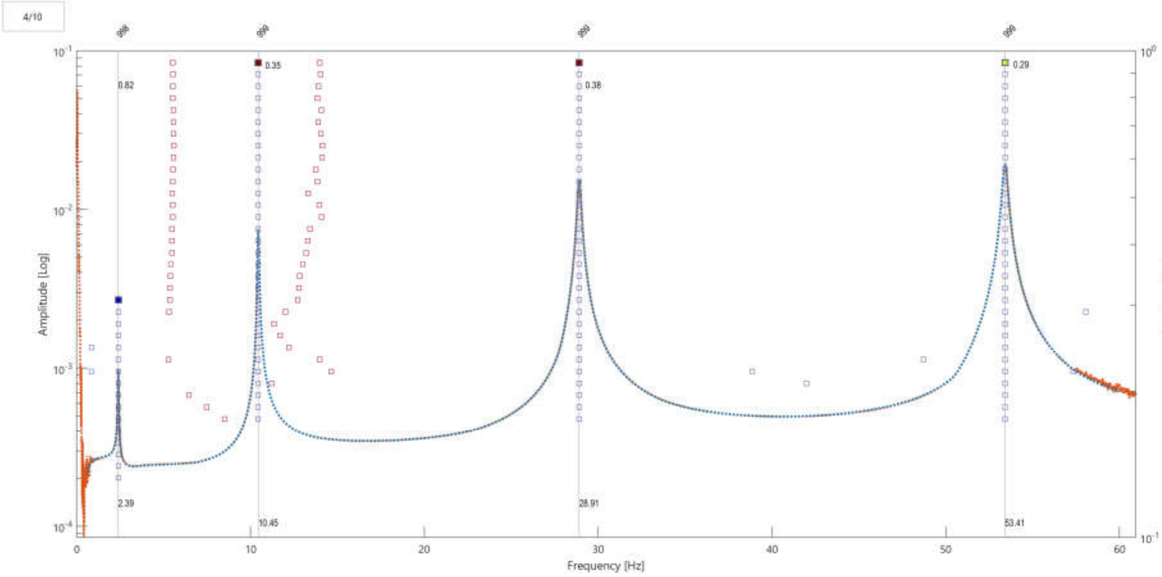


Figure 20 – Stabilisation diagram for the first level with Least Squares Complex Frequency-domain analysis

After the eigenfrequencies and damping ratios are already compared, it is additionally interesting to see how the identified eigenvectors behave compared to the analytical ones. Therefore the Mode Assurance Criterion (MAC) is used. Let $\boldsymbol{\phi}_a$ be an eigenvector identified through the LSCF algorithm and $\boldsymbol{\phi}_b$ an analytical one of the same system [Soal, 2018]. The MAC writes

$$\mathrm{MAC} = \frac{|\boldsymbol{\phi}_a^T \boldsymbol{\phi}_b|^2}{(\boldsymbol{\phi}_a^T \boldsymbol{\phi}_a)(\boldsymbol{\phi}_b^T \boldsymbol{\phi}_b)} \tag{8.1}$$

and gives a value near to one if the two compared modes are similar and tends to zero if the mode shapes are orthogonal.

By looking at figure 22 it can be clearly seen that the main diagonal has values very close to one which indicates that the LSCF identification can reproduce the analytical eigenvectors accurately. The same can be stated after looking at the modal parameters, the eigenfrequencies and damping ratios, see figure 21.

In contrast to the other investigations, it is worth to mention that within this subchapter, not OMA but EMA is performed, since after decoupling the system, the force which is at the

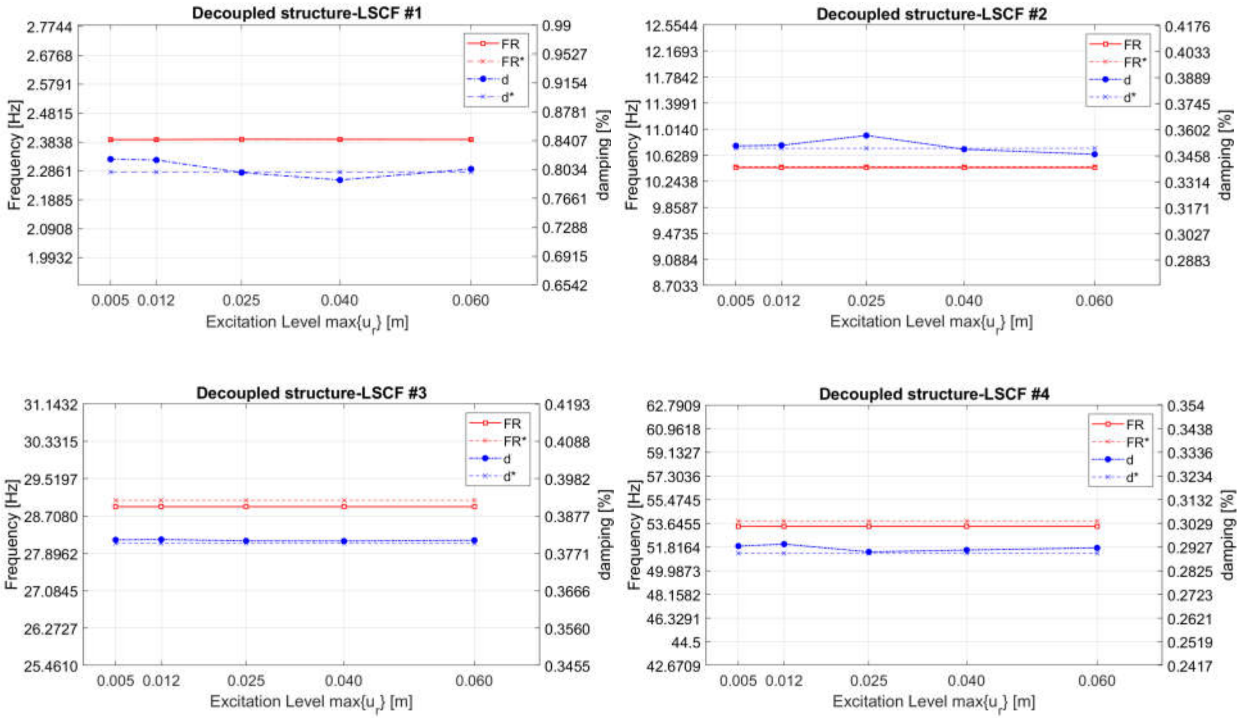


Figure 21 – Nonlinearity plots for different excitation levels of the preliminary investigation, FR^* and d^* show the analytical values

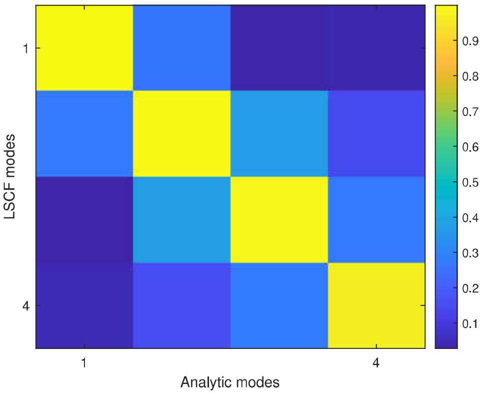


Figure 22 – MAC matrix to compare the analytical and identified modes

same time the excitation is known and used as reference. To conclude this section, it can be shown that if the nonlinear part is part of the excitation mechanism and the connection between these two is given through a unique, statically determined path, the two subsystems can be decoupled. Afterwards the modal parameters from the linear part can be determined and reflect the analytical solutions very well within a specific accuracy limit. It must be critically mentioned that not the exact values can be reproduced but only good approximations. For further information on how accurate the identification with LSCF can be performed, the reader is referred to Böswald [2016].

Anyway, the problem is that in reality the force which is transmitted into the wing can not be measured or computed easily. However, in an experiment, strain gauges can be installes at the upper landing gear part to measure the excitation force. Nevertheless, in this chapter only the feasibility of this approach shall be shown. Conversely, the decoupling of both subsystems could also be used to identify the nonlinear modal parameters of the landing gear with the help of the Restoring Force Surface method. However, this will not be discussed further in this thesis.

8.2 Main idea for nonlinear system identification

This section shall explain the basic idea of how to identify the nonlinear modal parameters within the OMA. This idea shall lay the foundation on which the following methods are built. In chapter 7 it is shown that with increasing the excitation signal in terms of its maximal displacement, the modal parameters change. As it is explained before, SSI is formulated for linear systems and thus not applicable for nonlinear influenced ones. Since the aim is to show the nonlinear trend of the system solely based on one signal, or respectively the responses obtained from one excitation level, the idea is to take the relative displacement of the landing gear and resort or manipulate it, such that the modal analysis can be performed for different levels which have however a fewer number of samples. Within the TVT it is possible to measure the relative displacement. This is considered as a measure of excitation of the aircraft, which is why the relative displacement of nodes 11 and 12, see figure 11, is used for sorting. Anyway, fundamental statistical equations are employed to realise this sorting or manipulation, respectively. The relative displacement can be computed using the translatory displacement of node eleven and twelve, see again 11, which is written as

$$y^{rel} = q_{11,x} - q_{12,x} \tag{8.2}$$

while assuming that this signal contains valuable information of the landing gears state, whether it is in stick or slip phase.

8.3 A standard deviation-based windowing technique

The first idea is to classify the data into specific levels and afterwards employ SSI to identify the nonlinear behaviour. The relative displacement signal, defined in equation (8.2) shall serve as a reference to define a number of excitation levels based on the displacement to then resort the data set of the system. Therefore, the run with the excitation level $\max\{u(t)\} = 25mm$ is chosen. This run is not only used for this method but also for all subsequent analysis. Anyway, the algorithmic procedure for this first method can be described as follows

```
Result: Modal parameters $\lambda_i$, $d_i$, $\phi_i$ for each level set $\mathbf{y}_L$
Using system output $\mathbf{y}$ and reference signal $y^{ref}$
Defining window size and overlapping factor
Compute number of windows $n_W$
for Number of defined windows $n_W$ in $y^{ref}$ do
    Compute standard deviation $\boldsymbol{\sigma}_{y^{ref}}$ of windows
end
Define levels based on standard deviation $\boldsymbol{\sigma}_{y^{ref}}$
for Number of levels $n_L$ do
    for Found values do
        Extract segment out of the systems output data $\mathbf{y}$
        Link the segments within the same level together $\mathbf{y}_L$
    end
end
for Number of levels $n_L$ do
    Perform SSI on the datasets $\mathbf{y}_L = \begin{bmatrix} \mathbf{y}_L^1 & \mathbf{y}_L^2 & \cdots & \mathbf{y}_L^{n_L} \end{bmatrix}$
end
```

Algorithm 1: Standard deviation-based windowing technique

In this case, the number of levels n_L is set to ten. The corresponding code can be found in the appendix, see D.1. To sum up, it is the aim to find a number of segments within the reference signal which share a specific statistical properties to be more precise a specific mean value and standard deviation. After classifying the signal and attach all segments one behind the other, it can be visualised that the different reference signals have different levels of displacement, see figure 23. In the end n_L data sets exist, consisting of a number of segments which are simply linked together. Nevertheless, it shall emphasised again that all levels are extracted from one simulation run. In the end, SSI is employed for each data set n_L.

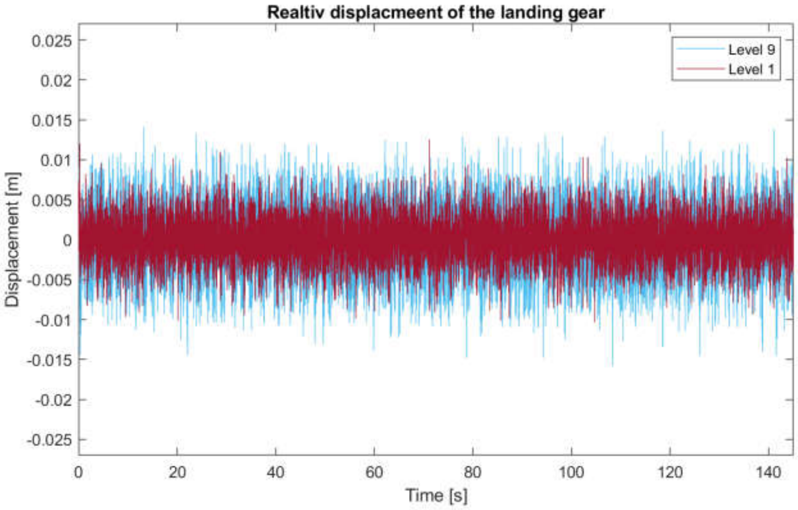


Figure 23 – Visualisation of sorted excitation levels for an arbitrary time segment of 140 seconds

After extending this sorting to the systems output $\mathbf{y}$, different data sets for the number of levels n_L are available which can be analysed with SSI. After employing SSI for each level, the already introduced nonlinearity plots are used again to show the performance of this method. The results are visualised in figure 24.

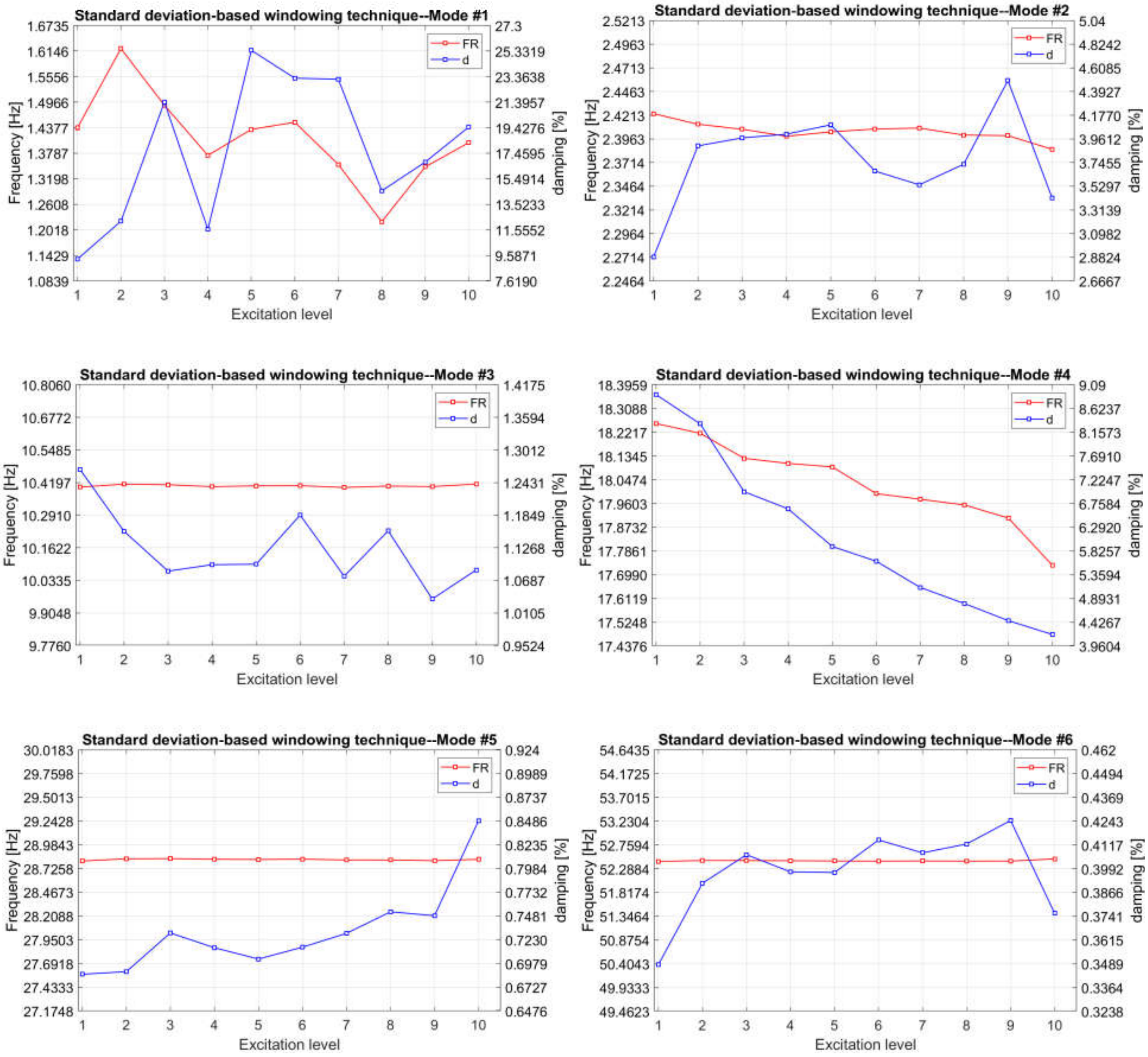


Figure 24 – Nonlinearity plots as results of the standard deviation-based windowing technique

By comparing the results with the ones shown in figure 18 it can be easily seen that especially the damping ratios can not be identified properly. In addition to that, also the change in frequency does not fit. By looking at mode number one and four which are earlier identified as clearly nonlinear modes, uncertainties can be seen.
Now some crucial points shall be discussed. By computing the standard deviation value of each window, n_w standard deviation values are obtained. By looking at the distribution of these values one rapidly realises that these are normally distributed, too, due to the randomness of the signal. That means that by resorting the signal with respect to the defined levels,

the length of each signal varies, depending on how extreme the level is. That is why a more extreme level, or differently said with a more significant distance (positive and negative) from the mean value, has less samples than a level in the middle of the level band. This of course has an impact on the identification through the algorithm, which is why the level band has to be chosen narrow enough such that an analysis through SSI is possible after all. Nevertheless, there are many other parameters which have an impact, too, such as the block size (as introduced in chapter 6 as a user defined parameter within SSI), the selected model order and the window size. Within the investigation this could be one aspect to enhance the results. If the window size is chosen too large, no proper levels can be identified, due to an averaging effect caused by the normal distribution of the signal. Additionally it is worth to mention that by setting the overlapping factor > 0 the same segments can be appear multiple times within the sorted data. Even after further investigations of these parameters no enhancement could be achieved.

Another critical point is that after resorting the signal, the single segments are simply linked together which creates discontinuities within the signal, which is then non-steady. By investigating a known linear system and by cutting out several segments, linking them together and employing SSI,it can be observed that this point, in turn, is not problematic for the results.

Since the results of this method were not satisfactory, another method is described and investigated in the following.

8.4 A Rainflow-counting algorithm-based identification

The second approach is based on the so called Rainflow Counting Algorithm (RCA). Again, the idea is to resort the data within a pre-processing step into specific levels on which then SSI can be employed. This time the classification is not carried out with the help of the simple windowing technique, but uses the aforementioned RCA. The algorithm itself is not discussed here in detail. For more information the reader is referred to the cited literature. Instead, it will shortly be explained how it works and which information it can provide.

The RCA, firstly presented by Endo et al. [1974], is mainly used within the fatigue analysis where it is aiming to count the load cycles. The method is based on the stress-strain response of a mechanical system, where the path is influenced by the history. A huge advantage is that even smaller hysteresis loops (or respectively load cycles) nested within larger hysteresis loops can be detected and counted. Anyway, one cycle is counted for one closed hysteresis loop. After counting, in fatigue analysis it is common to proceed with for example the MINERs rule, this, however, without going into the details. The complex RCA can be, albeit not straight forward, programmed and gives as an output the cycle count, the corresponding range which is the distance between the minimal and maximal displacement, the mean values of the considered samples, the starting and the ending sample.

The idea of the algorithm is the following. Imagine water which drops from a pagoda roof.[4] To obtain the cycle count from the variable amplitude loading, the peaks and valleys of the signal are needed. Therefore, the regarded signal is turned 90° clockwise. At each minimum and maximum water starts to flow and drop down the "roof" or peak, respectively. The

[4] A pagoda is a layered tower with multiple eaves that is common in China, Japan, Korea, Vietnam and other parts of Asia.

drop will stop when it passes an absolute value which is equal or bigger than the value where it started. Additionally it stops when it merges with a path of an early started drop or it reaches the end [Rychlik, 1987]. This is illustrated in figure 25.

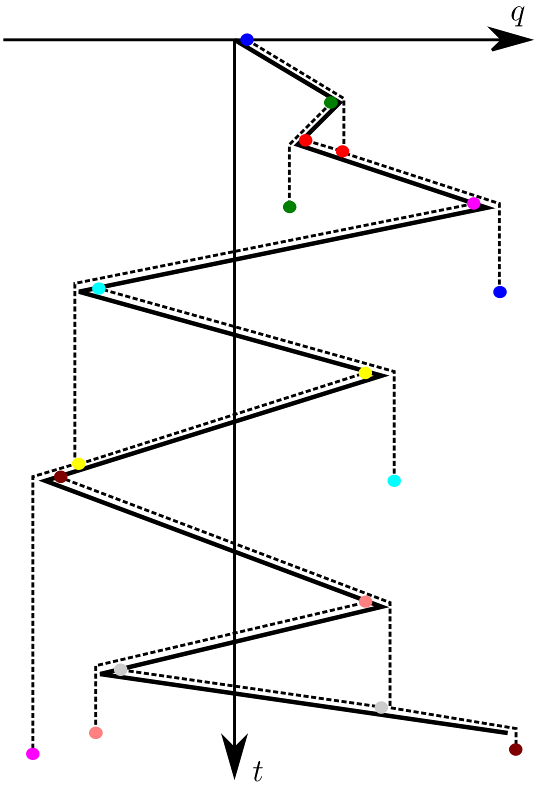


Figure 25 – Rainflow counting algorithm example

From this scheme, the hysteresis curves can now be derived, which, however, is not of interest here. The idea is to use the computed range from the RCA to define again specific levels to sort the signal and finally perform SSI. The range defines the absolute value between the start and the end point, for which the start and the end sample is additionally given. An important advantage of this method is, that no higher means of the peaks can appear within the sorted signals, thanks to the formulation of the RCA. The combination with SSI can be formulated as

```
Result: Modal parameters $\lambda_i$, $d_i$, $\phi_i$ for each level set $\mathbf{y}_L$
Using system output $\mathbf{y}$ and reference signal $y^{ref}$
Applying RCA on the reference signal $y^{ref}$
Delete cycles with to less samples
Using the range (output of RCA) to define $n_L$ levels
for Number of Levels $n_L$ do
    for Found values do
        Extract segment out of the systems output data $\mathbf{y}$
        Link the segments within the same level together to obtain $\mathbf{y}_L$
    end
end
for Number of levels $n_L$ do
    Perform SSI on the datasets $\mathbf{y}_L$
end
```

Algorithm 2: Rainflow-counting algorithm-based identification

Since through RCA also segments with a very low number of samples can appear, it is crucial to avoid that finally not enough data samples are available to achieve stable and confident results with SSI. After again performing the latter for each level, the results are illustrated in figure 26.

By looking at them, the same problems can be seen as in the chapter before. Since this approach is based on the same idea as the previous one, the same conclusions can be drawn. In the end the problem of this and the previous method could be, that despite the fact that the signal is sorted, still slices of higher levels can be appear since the sorting is solely based on statistical properties. These higher level shares can smear the results for the analysis.

8.5 An advanced stochastic subspace identification algorithm

After the results of the previous two methods were not satisfying, a third method is developed in this section. The idea is not only to treat the output data in form of a level dependent selection and initiate the data into the SSI algorithm but manipulate the latter actively by cancelling out columns in the Hankel matrix within the SSI algorithm. The main idea is to eliminate not needed columns to separate the different excitation levels to be afterwards able to identify the nonlinear trend within the modal parameters. The cancelling of columns is introduced by Juang [1993] to improve the numerical conditioning. This idea is taken up here with the previously formulated aim of identifying nonlinear systems. Here again, a reference signal for the level-dependent selection of the response data is used, introduced in equation (8.2).

The fundamental theory is explained in chapter 6. To evoke the algorithm again, the systems output is arranged in a Hankel matrix like

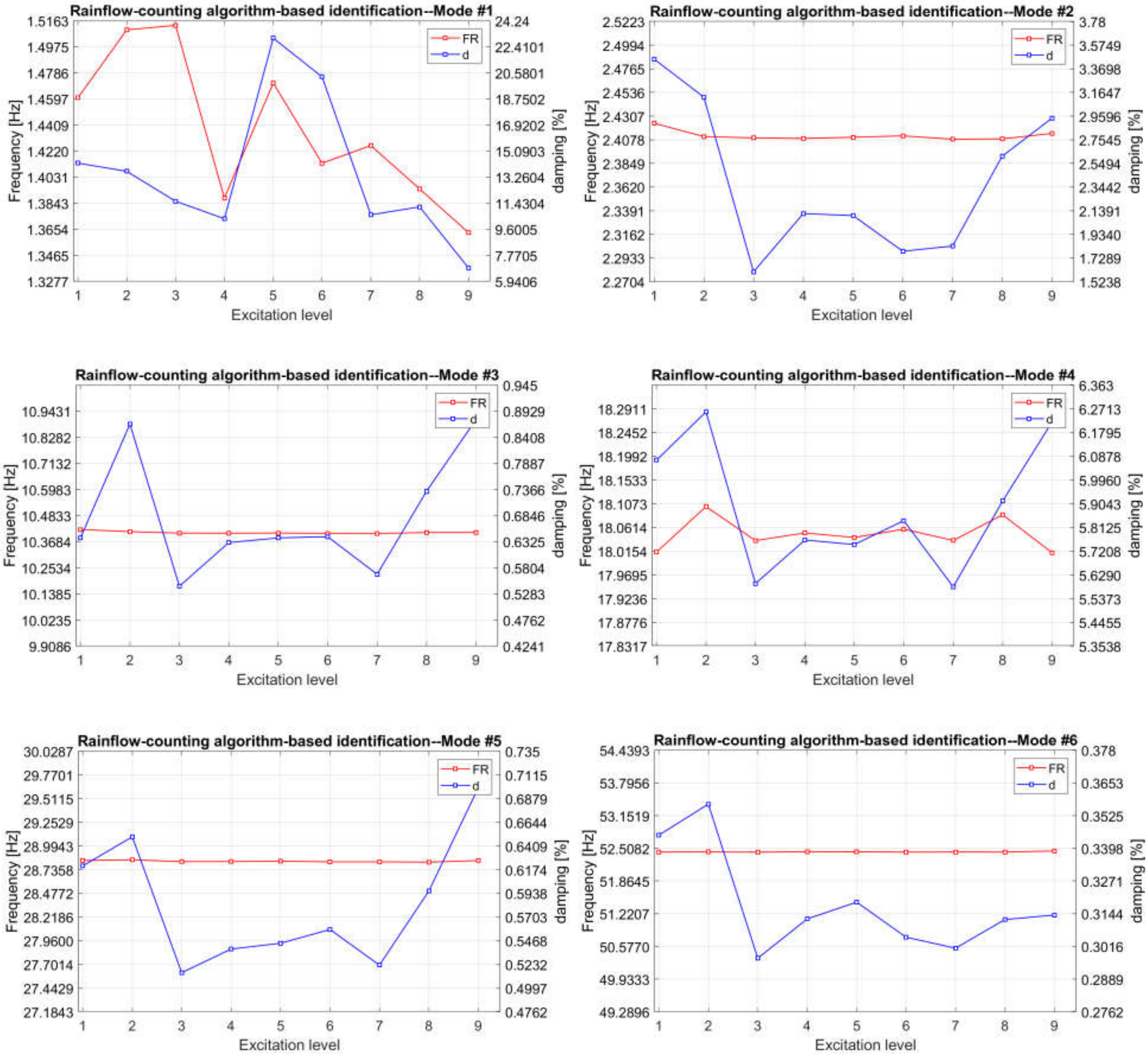


Figure 26 – Nonlinearity plots as result for the rainflow counting algorithm-based identification

$$\mathbf{Y}_H = \frac{1}{\sqrt{j}} \begin{bmatrix} \mathbf{y}_0 & \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_{j-1} \\ \mathbf{y}_1 & \mathbf{y}_2 & \mathbf{y}_3 & \cdots & \mathbf{y}_j \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{y}_{i-1} & \mathbf{y}_i & \mathbf{y}_{i+1} & \cdots & \mathbf{y}_{i+j-2} \\ \mathbf{y}_i & \mathbf{y}_{i+1} & \mathbf{y}_{i+2} & \cdots & \mathbf{y}_{i+j-1} \\ \mathbf{y}_{i+1} & \mathbf{y}_{i+2} & \mathbf{y}_{i+3} & \cdots & \mathbf{y}_{i+j} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{y}_{2i-1} & \mathbf{y}_{2i} & \mathbf{y}_{2i+1} & \cdots & \mathbf{y}_{2i+j-2} \end{bmatrix} \tag{8.3}$$

where $\mathbf{y}_0 \in \mathbb{R}^{N\times 1}$ denoting the current coordinates of each DoF at a specific time point. In the same manner the reference signal (the relative displacement of the landing gear) y^{ref}, defined in equation (8.2), is arranged as

$$\mathbf{Y}_H^{ref} = \frac{1}{\sqrt{j}} \begin{bmatrix} y_0^{ref} & y_1^{ref} & y_2^{ref} & \cdots & y_{j-1}^{ref} \\ y_1^{ref} & y_2^{ref} & y_3^{ref} & \cdots & y_j^{ref} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_{i-1}^{ref} & y_i^{ref} & y_{i+1}^{ref} & \cdots & y_{i+j-2}^{ref} \\ y_i^{ref} & y_{i+1}^{ref} & y_{i+2}^{ref} & \cdots & y_{i+j-1}^{ref} \\ y_{i+1}^{ref} & y_{i+2}^{ref} & y_{i+3}^{ref} & \cdots & y_{i+j}^{ref} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_{2i-1}^{ref} & y_{2i}^{ref} & y_{2i+1}^{ref} & \cdots & y_{2i+j-2}^{ref} \end{bmatrix} \qquad \mathbf{Y}_H^{ref} \in \mathbb{R}^{l\times k} \tag{8.4}$$

with the difference that $y_0^{ref} \in \mathbb{R}$ of the reference signal is just a scalar value. Let in the following l denote the number of rows and k the number of columns. The matrix can be rewritten as

$$\mathbf{Y}_H^{ref} = \frac{1}{\sqrt{j}} \begin{bmatrix} \mathbf{y}_{H,0}^{ref} & \mathbf{y}_{H,1}^{ref} & \mathbf{y}_{H,2}^{ref} & \cdots & \mathbf{y}_{H,k}^{ref} \end{bmatrix} \qquad \mathbf{y}_{H,i}^{ref} = \begin{bmatrix} y_i^{ref} \\ y_{i+1}^{ref} \\ y_{i+2}^{ref} \\ \vdots \\ y_{l+i}^{ref} \end{bmatrix} \in \mathbb{R}^l \tag{8.5}$$

In the following the standard deviation of each column of the matrix $\mathbf{Y}_H^{ref}$ is computed

$$\sigma_{Y_{H,i}^{ref}} = \sqrt{E\left[\left(\mathbf{y}_{H,i}^{ref} - \mu_{Y_{H,i}^{ref}}\right)^2\right]} \tag{8.6}$$

using the mean value

$$\mu_{Y_{H,i}^{ref}} = \frac{1}{l}\sum_{i=1}^{l} y_i^{ref} \qquad y_i^{ref} \in Y_{H,i}^{ref} \tag{8.7}$$

and leading to

$$\boldsymbol{\sigma}_H^{ref} = \begin{bmatrix} \sigma_{Y_{H,0}^{ref}} & \sigma_{Y_{H,1}^{ref}} & \sigma_{Y_{H,2}^{ref}} & \cdots & \sigma_{Y_{H,k}^{ref}} \end{bmatrix} \qquad \boldsymbol{\sigma}_H^{ref} \in \mathbb{R}^k. \tag{8.8}$$

After computing the standard deviation of each column, different levels can be defined. By again computing the standard deviation of the vector $\boldsymbol{\sigma}_H^{ref}$, ongoing denoted as $\sigma_L \in \mathbb{R}$ and the mean value μ_L, the levels L lie in the interval

$$-e\sigma_L + \mu_{\boldsymbol{\sigma}_H^{ref}} < L \leq e\sigma_L + \mu_{\boldsymbol{\sigma}_H^{ref}} \qquad e \in \mathbb{R} \tag{8.9}$$

where e is a constant, defining the width of the interval, which can then be divided into n_L sub-intervals. The value e typical takes values between 1 and 2.5 to ensure that enough values for a specific level are found, see Vesterholm et al. [2018]. Subsequently the subintervals represent the different levels. By looping over the number of levels, the task is now to find all columns of the reference signal whose standard deviation falls into a corresponding level band. The columns which fulfil this requirement are kept, the others are deleted to obtain a reduced HANKEL matrix. This pattern is then transferred to $\mathbf{Y}_H$ which brings the reduced matrix $\mathbf{Y}_H^{red}$, such as

$$\mathbf{Y}_H^{red} = \frac{1}{\sqrt{j-d}} \begin{bmatrix} \mathbf{y}_0 & \cancel{\mathbf{y}_1} & \cancel{\mathbf{y}_2} & \cdots & \mathbf{y}_{j-1} \\ \mathbf{y}_1 & \cancel{\mathbf{y}_2} & \cancel{\mathbf{y}_3} & \cdots & \mathbf{y}_j \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{y}_{i-1} & \cancel{\mathbf{y}_i} & \cancel{\mathbf{y}_{i+1}} & \cdots & \mathbf{y}_{i+j-2} \\ \mathbf{y}_i & \cancel{\mathbf{y}_{i+1}} & \cancel{\mathbf{y}_{i+2}} & \cdots & \mathbf{y}_{i+j-1} \\ \mathbf{y}_{i+1} & \cancel{\mathbf{y}_{i+2}} & \cancel{\mathbf{y}_{i+3}} & \cdots & \mathbf{y}_{i+j} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{y}_{2i-1} & \cancel{\mathbf{y}_{2i}} & \cancel{\mathbf{y}_{2i+1}} & \cdots & \mathbf{y}_{2i+j-2} \end{bmatrix}, \tag{8.10}$$

where d defines the number of deleted columns. Starting with equation (6.6), the standard SSI procedure is executed, albeit with the reduced HANKEL matrix. This identifies the modal parameters for each level, whereby it is aimed to generate the same change of the modal parameters as visualised in figure 27.

By looking at the results, the modal parameters of all modes can be reproduced with respect to figure 18. It is worth to mention again that in the just referenced figure the excitation levels result from the different simulation runs. In contrast to that, the levels in figure 27 are based on the standard deviation values, defined in equation (8.8), and thus can not be brought together on one axis. As it is explained before, the level band is chosen such that still enough values are found for each level to perform a proper modal analysis. Anyway, only for mode four the nonlinear characteristics as shown in figure 27 cannot be identified. Because this method looks promising after the first run, it is further investigated what can

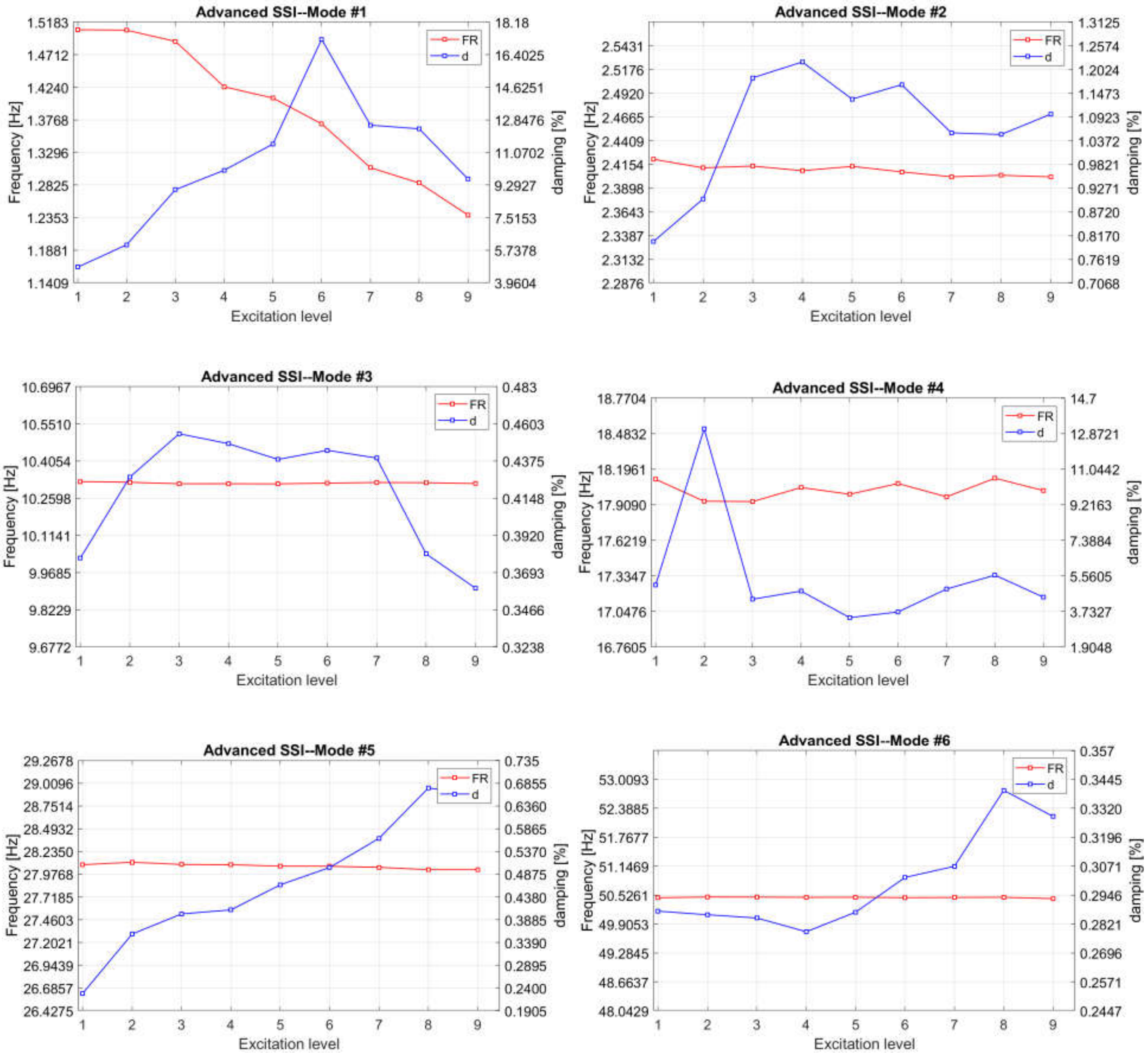


Figure 27 – Nonlinearity plots as result for the advanced Stochastic Subspace Identification algorithm

cause the inaccuracy at this specific fourth mode. For this, various plausible reasons are explored with regards to what could trigger the inability to identify the nonlinear trends in eigenfrequency and damping ratio.

The first reason one think of is that the deleting of columns in the Hankel matrix influence the data quality. It is believed that a minimum number of consecutive columns must be retained to perform modal analysis. By saying that columns occurring only in groups of n-columns, the identification procedure is employed again but no improvements can be observed. Further it is looked at the appearance of columns again. Additionally, no correlation between the appearance of columns with respect to the different levels is seen, which leads to the conclusion that even the selection of the columns happens randomly and no pattern can be identified.

Another point is the extension of the criteria for the selection of the columns. Until now, a reference signal which is the relative displacement of the nonlinear subsystem is regarded. When looking at equation (3.9), it becomes clear that the nonlinear friction force is not only dependent on the displacement but also on the velocity. In general, a good way to do the selection would probably be to use the energy within the friction element. By keeping in mind that this approach is supposed to lay the foundation for the TVT, this is hence not applicable because no real material parameter can be obtained from the aircraft structure. This is why another approach is chosen to not only use the displacement as a criterion for the selection but the velocity as well. The relative velocity can be easily computed. With the simulated data this can be done by simply using the velocity data since not only the displacement but also the velocity is the result of the state space formulation. Within the TVT this can be done by using the sensor data in combination with for example finite differences. Thereby, it is important to say that by expanding the selection criterions, less columns are picked for each level, since they have to meet an additional requirement. Although acceptable results were generally achieved by this attempt, the nonlinearity of mode four was not improved by it.

Another critical point is the choice of the parameter within the SSI algorithm. The user is supposed to define the so called block size and model order. By looking again at equation (8.4), the block size defines the parameter l. Because the standard deviation is computed for each column, the block size hence defines a kind of window size since the value among y_i^{ref} is y_{i+1}^{ref} and so on. The bigger the block size is chosen, the bigger is the regarded window size. Due to the random distribution, the bigger the window size the closer is the standard deviation value to the mean value, also described by the law of the large numbers. This is not beneficial for the definition of the individual level. However, the block size of course has an impact on the obtained results and has to be chosen with caution. In the end this can not improve the results of mode number four either.

By regarding the mode shape of this fourth mode in figure 17, it becomes clear that mode number four is effecting solely the landing gear DoF. Another idea was to use the so called referenced based SSI, see Peeters and De Roeck [1999], or use only the landing gear DoF for modal analysis but neither the first nor the latter is a successful attempt.

The last aspect which shall be discussed here also aims to identify mode number four more accurately. Thereby the attempt is made to filter the reference signal before introducing it into the advanced SSI algorithm. The idea behind that is to filter the signal such that solely the frequency band around mode number four is kept. It must be stated again that this attempt is not successful either.

Despite the imprecise identification of the aforementioned mode number four, it shall be investigated how the presented method works on higher and lower excitation signals. Therefore, the system response with the excitation levels two ($\max\{u(t)\} = 0.012m$) and four ($\max\{u(t)\} = 0.040m$) are taken to compute the system response which shall be further investigated with the advanced SSI algorithm. It becomes clear that the nonlinearities at level two are not detected because the energy of the excitation is not large enough to trigger the nonlinear friction. It should be noted that the force f_c was determined by a static analysis of the system and is only triggered at a displacement of approximately $0.01m$. Due to the normal distribution, it can be assumed that this signal is energetically not sufficient enough to trigger the nonlinearity. This result is in line with expectations. Thus, this level does not need further attention here.

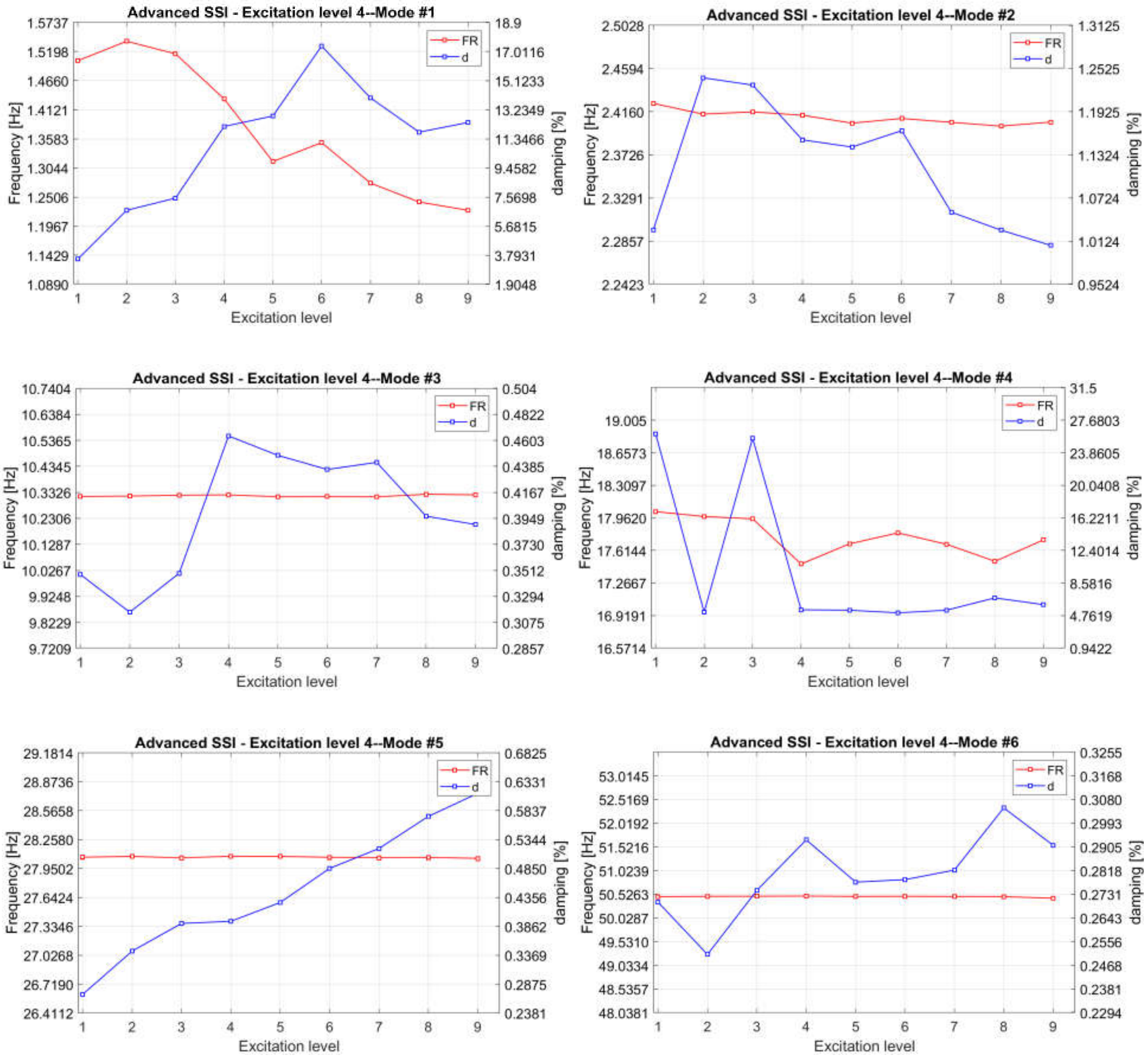


Figure 28 – Nonlinearity plots as results for the advanced Stochastic Subspace Identification algorithm at excitation level four

On the other hand, the signal in level four has a higher energy and does trigger the nonlinearity. In figure 28 the corresponding nonlinearity plots can be seen which show the correct trend as before. To be able to better compare especially mode number one, figure 29 shows this first mode identified through the simulations of level three and four.
Thereby it can be seen that the nonlinear trend is well captured, even though small deviations occur, resulting through the fact that the same levels were used for the fourth simulation run as for the third. This means that for the lower levels less corresponding columns appear which influence the result.

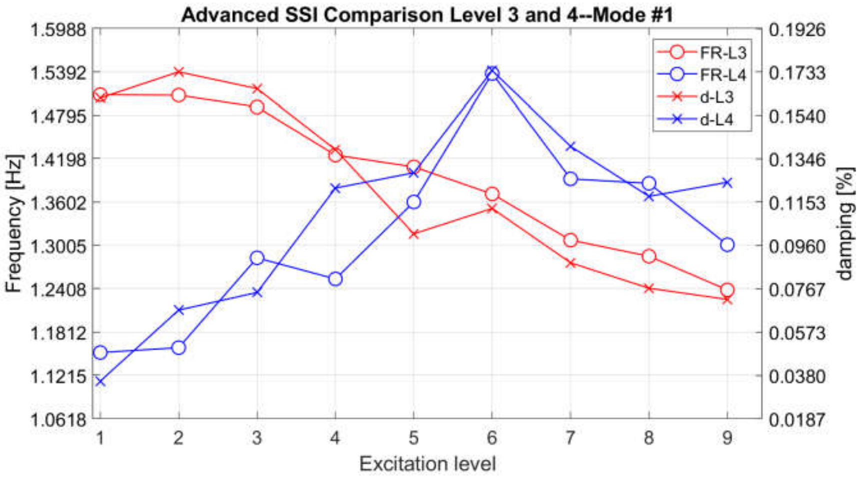


Figure 29 – Comparison of mode one identified through the third and fourth level excitation

9 Conclusion and Outlook

This last chapter is supposed to conclude this thesis. Therefore, a recap and a discussion of the obtained results is given. Afterwards, the work is critically reflected and an outlook is provided by issuing research questions based on this work.

9.1 Discussion of the results

Within this work a numerical efficient substitute model is built to be able to access all data and parameters at any time. Additionally with the help of stochastic analysis and signal processing a random excitation is modelled to reflect the excitation of the aircraft by the runway. By using a newly developed RUNGE-KUTTA integration scheme to be able to compute the solution of the ODE in an appropriate amount of time, the substitute model is simulated. The resulting data shows very well the nonlinear effect of the landing gear. The visual comparison with real aircraft data supports the obtained results since similarities of modeshapes and eigenfrequencies can be seen. Furthermore, within chapter 7 different hysteresis curves are shown which show the force-displacement behaviour of the nonlinear branch within the landing gear. At this point it is worth to mention that the parameter of the aforementioned nonlinear element are chosen by performing a static analysis of the system and by defining that the JENKINS element changes from stick to slip at an excitation displacement of approximately $10mm$.
Subsequently, different methods were developed and investigated. Within the TVT it is not possible to excite the structure with different levels, resulting in the fact that the identification of nonlinear modes is not possible. Therefore, the methods are supposed to use statistics to classify the data to identify the nonlinear behaviour. The first method represents a case that is not applicable in practice but shows very well, that if the linear system can be decoupled from the nonlinear system and the transmitted force between these two subsystems is computable, the linear structure can be properly identified, using LSCF as an EMA technique. This is confirmed by comparing the identified and analytical modal parameters, where even the eigenvectors can be well reproduced. As it is mentioned in chapter 8, the accuracy within such a model analysis method is investigated for instance in Böswald [2016].
After this preliminary investigation is carried out, the actual aim of this work is pursued. The first two methods, namely the standard deviation-based windowing technique and the rainflow counting algorithm-based identification are both based on the idea of sorting the system response into specific levels. This is done as a kind of pre-processing step before the SSI algorithm is employed on the sorted data. By looking at the results and comparing them with the nonlinearity plots obtained in chapter 7 it can be seen that besides the fact that the nonlinear modes can not be identified properly, especially the damping values deviate significantly. After further investigating different aspects which could cause these problems, no solution is found which leads to the dismissal of these two methods and the development of a third one.
The third method is based on the same idea but uses a different approach. Thereby, the data is not manipulated within a pre-processing step but the SSI algorithm is enhanced directly by implementing a scheme to reduce the BLOCK HANKEL matrix. By looking at the results, one immediately realises the potential of this method since five of the six modes were identified quite accurately. Even the damping ratios, which can become very critical are

identified well. Only the nonlinear trend of mode number four, a pure landing gear mode, can not be reconstructed correctly. However, it must be mentioned positively that the results are nevertheless realistic, only the non-linear trend is not visible. After investigating again several aspects, described within chapter 8, no solution is available yet. Up to this point, the methods were solely tested on the simulation run with $\max\{u(t)\} = 25mm$. To validate the algorithm, also the simulations with an excitation level of $\max\{u(t)\} = 12mm$ and $\max\{u(t)\} = 40mm$ were used. As expected, the $12mm$ case shows no nonlinearity since the displacement is energetically not sufficient to change the friction state from stick to slip. Nevertheless, the modes are identified properly and solely the nonlinear trend can not be seen. Remember that the parameter f_c was adjusted such that the system changes from stick to slip at approximately $10mm$. The other simulation in turn shows again the nonlinear trend, what strengthens the benefit and the validity of this method under the condition that the excitation triggers the nonlinearity sufficiently. In the end it can be stated that the advanced SSI algorithm is a promising approach and shows, apart from the fourth mode, good results. However, the question remains what triggers this inaccuracy in this one mode while the others can be well identified.

9.2 Critical reflection

This section is designated to reflect this thesis critically. First of all it has to be mentioned that the minimum amount of theory is reported here that is required to understand the technical problem and the solutions adopted in this work. For more information the reader is referred to the cited literature. Moreover, it is crucial to list that in addition to the blank algorithms the Matlab toolbox of the DLR is used to increase the functionality and to obtain better results.

Furthermore, the modelled system has to be reviewed. It must be emphasised again that only a simplified model is constructed and regarded in this work, to demonstrate the identification methods. Using a more detailed model can have an impact on the results. Thereby, for a three dimensional model the symmetry condition can no longer be applied and antisymmetric modes can appear in addition. Nevertheless, it must be stated that the results obtained from the simulation performed very well compared to real aircraft data during vibration test. Regarding the substitute model, the used parameter have to be reviewed critically as well. The example aircraft should be modelled as accurately as possible. Therefore, either the parameter were taken from empirical values of the DLR, out of the cited Airbus documentary or estimated as properly as possible. As it is mentioned, a check of the overall weight was carried out which supports the correctness of the model. In this course also the so called piston and the fitting of the landing gear have to be regarded. Within the work, these two parts are modelled without any damping. It is clear that there are no parts without damping. However, for the sake of simplicity, these parts are assumed to be so stiff that damping would not have an major impact on the results.

At next, the Valanis model is considered. Compared with the also introduced LuGre model, the results, especially the hysteresis curve could be reconstructed better. Additionally the transition from stick to slip can be seen more precisely since it is controllable via the parameter κ, which, in contrast, is not possible within the LuGre model. This is why the Valanis model is chosen here. However, it must be stated that there are of course

other friction models which can be employed. Regarding the nonlinearities, in chapter 3 the harmonic balance method is used to compute the equivalent stiffness and damping ratios of the JENKINS element. Of course, within this work only randomly distributed excitation are regarded while for the harmonic balance method a harmonic excitation is assumed. The use of a harmonic excitation is only intended to show how such a nonlinear element behaves with changing harmonic motion. It must be noted that the comparison of harmonic and random levels is difficult because the axis can only be related to each other in approximation.
A next important part is the developed time integration scheme. Of course there are schemes which are more accurate and stable. Within this work it is however the aim to be able to simulate the model in a reasonable time. As it is written in chapter 5, firstly the simulation was carried out with ODE45 in combination with the interpolation method INTERP1, due to the fact that the regarded excitation is non-deterministic and can not be evaluated based on an analytical function. Because of the very long simulation time and the huge number of samples, not even the DLR cluster was able to compute the system response within days. This is why another method was crucial to not further delay the course of this work. In the end it is possible to compute the system response for a simulation time of 9000 seconds within approximately 30 minutes, which is a huge advantage in contrast to the MATLAB functions, while a sufficiently small error is accepted.
Regarding the used modal analysis methods it has to be mentioned that only data-driven SSI and LSCF are employed within this work, while there are of course other variations of these algorithms.
The last major point is the review of the obtained results. It must be mentioned that it would of course be advantageous to compare the results from the various simulations in chapter 7 with those from chapter 8 in one plot. However, this is not possible due the unclear scaling of the response axis. The question arises how a purely harmonic and a purely random response can be compared. Another important point is that within the methods the number of levels is always set to ten to obtain a sufficient solution of the level band. Although, this parameter can also be changed. Similarly, the parameters of SSI, for instance the model order or the block size, play an important role and can significantly influence the results.

9.3 Outlook

In the end a short outlook shall be provided, to give impulses for further investigations. The results of the third method look very promising which is why it is worth to further pursue this approach. As it is already mentioned, several parameters can be adjusted which gives plenty of possibilities to further improve the results. In addition to that, it is also worth to mention that there are other modal analysis methods which perform maybe better for this application.
The last point to investigate could be to complement these methods with the so called Recurrence Quantification Analysis, which aims to uncover recurring phases of nonlinear systems. Ideally, the stick and slip phases could be better identified in order to achieve a more precise sorting of the signal.
Additionally, the method could be used to analyse actual acquired data from a TVT since within this work only the analysis of a model is regarded.

A Simulation Model

This section contains all Matlab files necessary to build and simulate the substitute model. To execute the file SimModell.m not only the functions rod2dof.m and beam4dof.m are needed but additionally the in the following listed random.m to generate the random excitation force and the improved Runge-Kutta methods.

A.1 SimModel.m

```
%% Substitute Model Half Aircraft Airbus A320
% Master Thesis Hauke Maathuis

%% Parameters

ner = 9; % number of elements beam
nad = 3; % number of additional nodes
nbe = ner + 1; % number of nodes beam
n = nbe + nad; % number of nodes (over all)
ndof_beam = nbe * 2; % number of DOFs beam
ndof = ndof_beam + nad; % number of DOFs (over all)
natt = 3; % node where the gear is attached

% Parameter Wing
pWing.E      = 0.73e11;
pWing.length = 17;
pWing.l      = pWing.length/ner; % Element length (discretization)
pWing.MAC    = 3.1; % Mean aerodynamic chord length
pWing.B      = pWing.MAC * 0.5;
pWing.H      = pWing.MAC * 0.15;
pWing.b      = pWing.MAC * 0.5 * 0.9;
pWing.h      = pWing.MAC * 0.15 * 0.9;
pWing.A      = pWing.B*pWing.H - pWing.b*pWing.h;
pWing.weight = 4243;
pWing.rho    = pWing.weight / (pWing.A * pWing.length);
pWing.I      = 1/12 * ( pWing.B*pWing.H^3 - pWing.b*pWing.h^3 );

% Parameter rods before and after the LuGre element within landing gear
% u -> upper rod
pRod.mass_u   = 610; % total mass of main fitting and piston
pRod.A_u      = 0.15;
pRod.l_u      = 1.1;
pRod.rho_u    = pRod.mass_u / (pRod.A_u * pRod.l_u);
pRod.E_u      = 2.1e11;
% l -> lower rod
pRod.mass_l   = 287;
pRod.A_l      = 0.15;
pRod.l_l      = 1.1;
pRod.rho_l    = pRod.mass_l / (pRod.A_l * pRod.l_l);
pRod.E_l      = 2.1e11;

% Parameter Tire
pTire.ktir = 3.13e6; % Stiffness of the tire (3.13e6)
pTire.dtir = 4500; % Damping of tire (but apparently not needed)
pTire.mtir = 202; % Mass of the tire

% additional masses
pXmasses.fuselage = 11800; % Additional mass fusalage
pXmasses.engine = 3100; % Additional mass engine
pXmasses.e = 1; % distance between node and center of gravity of the engine

% Parameter for the Valanis modell
val.Kges = 5e6;
val.Et = 0.2 * val.Kges;
val.E0 = 0.8 * lugr.Kges;
val.sigma0 = 3e4;
val.kappa = 0.90;
val.lambda = val.E0/(val.sigma0*(1-val.kappa*val.Et/val.E0));

HO = 3; % Hold-order for FastRK

%% Building the geometry model
% Node-Tabel of the system Tab = [Node-Number  X  Y  Z]

GeoTab = zeros(n,4);
for ii = 1:nbe
   GeoTab(ii,1) = ii;
   GeoTab(ii,2:4) = [pWing.l*(ii-1),0,0];
end
GeoTab(nbe+1,:) = [nbe+1,pWing.l*(3-1),-1.1,0];
GeoTab(nbe+2,:) = [nbe+2,pWing.l*(3-1),-1.1,0];
GeoTab(nbe+3,:) = [nbe+2,pWing.l*(3-1),-2.2,0];

% DOF Tabel DofTab = [Dof-Number  Node-Number  Dof-Direction(1:Translation,2:Rotoation)]

DofTab = zeros(ndof,3);
DofTab(:,1) = [1:ndof]';
DofTab(1:2*nbe,2) = sort(repmat([1:nbe]',[2,1]));
DofTab(1:2*nbe,3) = repmat([1;2],[nbe,1]);
DofTab(2*nbe+1:ndof,2) = nbe+[1;2;3];
DofTab(2*nbe+1:ndof,3) = [1;1;1];

%% Building element matrices
[me_2_u, ke_2_u] = SIMh.rod2dof(pRod.E_u,pRod.l_u,pRod.A_u,pRod.rho_u);
[me_2_l, ke_2_l] = SIMh.rod2dof(pRod.E_l,pRod.l_l,pRod.A_l,pRod.rho_l);
[me_4, ke_4] = SIMh.beam4dof(pWing.E,pWing.I,pWing.l,pWing.A,pWing.rho);

%% Initialize total values

S_total = zeros(3,1); % total center of gravity
M_total = 0; % total mass
I_total = 0; % total moment of inertia

%% Sorting element matrices into global

M = zeros(ndof,ndof); K = zeros(ndof,ndof); D = zeros(ndof,ndof);

```

```
% sort in the beam parameters
for i = 1:ner %loop over elements
    index = (i-1)*2 + [1:4];
    K(index,index) = K(index,index) + ke_4;
    M(index,index) = M(index,index) + me_4;

    tmp_Mass = 2*(me_4(1,1)+me_4(1,3));
    [M_total,S_total,I_total] = SIMh.RB_Inertia(GeoTab,DofTab,index,
   tmp_Mass,M_total,S_total,I_total);
end

Ms = M; Ks = K;

%% Additional Masses on Beam

idx = 1;
M(idx,idx) = M(idx,idx) + pXmasses.fuselage;
tmp_Mass = pXmasses.fuselage;
[M_total,S_total,I_total] = SIMh.RB_Inertia(GeoTab,DofTab,idx,tmp_Mass,
   M_total,S_total,I_total);

idx = 7;
M(idx,idx) = M(idx,idx) + pXmasses.engine;
tmp_Mass = pXmasses.engine;
[M_total,S_total,I_total] = SIMh.RB_Inertia(GeoTab,DofTab,idx,tmp_Mass,
   M_total,S_total,I_total);

idx = 8;
M(idx,idx) = M(idx,idx) + pXmasses.engine * pXmasses.e^2;
I_total = I_total + pXmasses.engine * pXmasses.e^2;

idx = ndof_beam+nad;
M(idx,idx) = M(idx,idx) + pTire.mtir;
tmp_Mass = pTire.mtir;
[M_total,S_total,I_total] = SIMh.RB_Inertia(GeoTab,DofTab,idx,tmp_Mass,
   M_total,S_total,I_total);


%% Defining static constrains

Dof_stat = 2; % rotation is fixed, translation is not
index_a1 = 1:ndof;
index_a1(find(index_a1==Dof_stat)) = [];
index_b1 = [Dof_stat];

%% Physical damping of the cantilever beam
modDamp = 0.005; % empirical value
index = [1 2];
Mn = Ms(1:ndof_beam,1:ndof_beam); Kn = Ks(1:ndof_beam,1:ndof_beam);
Mn(index,:)=[]; Mn(:,index)=[]; Kn(index,:)=[]; Kn(:,index)=[];

nnDOF = size(Kn,1);

[Psi,Lambda] = eig(Kn,Mn);

```

```
%Normieren der Eigenvektoren
for ii = 1:nnDOF
    [~,index_max]=max(abs(Psi(:,ii)));
    Psi(:,ii) = Psi(:,ii)/Psi(index_max,ii);
end

Lambda = diag(Lambda);
w_eig = abs(sqrt(Lambda));
Mn = Psi'*Mn*Psi; Kn = Psi'*Kn*Psi; % modal transformation

psi_zero = zeros(1,nnDOF);
Psi = [psi_zero; psi_zero; Psi];

k = zeros(2*(ner+1));

for i = 1:nnDOF
    kn = Psi(:,i)*Psi(:,i)'*(2*modDamp*w_eig(i)/Mn(i,i));
    k = k+kn;
end

D(1:ndof_beam,1:ndof_beam) = D(1:ndof_beam,1:ndof_beam)+ Ms(1:ndof_beam,1:ndof_beam) * k * Ms(1:ndof_beam,1:ndof_beam);

clearvars Kn Mn Ms Ks

%% Computing EF/EV of the damped beam for comparison in 8.1

tmp_id = [(natt*2)-1,ndof_beam+1];
M(tmp_id,tmp_id) = M(tmp_id,tmp_id) + me_2_u;

index = [Dof_stat];
Mb = M; Db = D; Kb = K;
Mb(index,:)=[]; Mb(:,index)=[];
Kb(index,:)=[]; Kb(:,index)=[];
Db(index,:)=[]; Db(:,index)=[];

dof_el = [20 21 22];
Mb(dof_el,:) = []; Mb(:,dof_el) = [];
Db(dof_el,:) = []; Db(:,dof_el) = [];
Kb(dof_el,:) = []; Kb(:,dof_el) = [];

Ab = [zeros(size(Mb)) eye(size(Mb)); -Mb\Kb -Mb\Db];
[Psi,Lambda] = eig(Ab); Lambda = diag(Lambda);
[~,id] = sort(Lambda); Lambda = Lambda(id); Psi = Psi(:,id);
id1 = find(imag(Lambda)>=0); id2 = find(imag(Lambda)<0);
Lambda = Lambda([id1;id2]); Psi = Psi(:,[id1;id2]);
id = find(imag(Lambda)>=0);
wn_beam = abs(Lambda(id)); fn_beam = wn_beam/(2*pi); Damp_beam = -1*real(Lambda(id))./wn_beam;
% Cut out lower part of psi matrix
Psi = Psi(1:19,id); Lambda = Lambda(id);
%Normalize eigenvectors
for ii = 1:numel(id)
    [~,index_max]=max(abs(Psi(:,ii)));
    Psi(:,ii) = Psi(:,ii)/Psi(index_max,ii);
```

```
end

%% Sort in the gear parameters

% Rod above friction element
tmp_id = [(natt*2)-1,ndof_beam+1];
K(tmp_id,tmp_id) = K(tmp_id,tmp_id) + ke_2_u;
M(tmp_id,tmp_id) = M(tmp_id,tmp_id) + me_2_u;
D(tmp_id,tmp_id) = D(tmp_id,tmp_id) + de_2_u;

tmp_Mass = 2*(me_2_u(1,1)+me_2_u(1,2));
[M_total,S_total,I_total] = SIMh.RB_Inertia(GeoTab,DofTab,idx,tmp_Mass,
    M_total,S_total,I_total);

% No damping in the bars under and above the LuGre element
% Stiffness and damping of the LuGre branch (parallel connected)
k1 = lugr.k1 * [1 -1; -1 1];
K(ndof_beam+1:ndof_beam+2,ndof_beam+1:ndof_beam+2) = K(ndof_beam+1:
    ndof_beam+2,ndof_beam+1:ndof_beam+2) + k1;
d1 = lugr.d1 * [1 -1; -1 1];
D(ndof_beam+1:ndof_beam+2,ndof_beam+1:ndof_beam+2) = D(ndof_beam+1:
    ndof_beam+2,ndof_beam+1:ndof_beam+2) + d1;

% Rod under the friction element
K(ndof_beam+2:ndof_beam+3,ndof_beam+2:ndof_beam+3) = K(ndof_beam+2:
    ndof_beam+3,ndof_beam+2:ndof_beam+3) + ke_2_l;
M(ndof_beam+2:ndof_beam+3,ndof_beam+2:ndof_beam+3) = M(ndof_beam+2:
    ndof_beam+3,ndof_beam+2:ndof_beam+3) + me_2_l;
D(ndof_beam+2:ndof_beam+3,ndof_beam+2:ndof_beam+3) = D(ndof_beam+2:
    ndof_beam+3,ndof_beam+2:ndof_beam+3) + de_2_l;

tmp_Mass = 2*(me_2_l(1,1)+me_2_l(1,2));
[M_total,S_total,I_total] = SIMh.RB_Inertia(GeoTab,DofTab,idx,tmp_Mass,
    M_total,S_total,I_total);

% Stiffness of the tire
K(end,end) = K(end,end) + pTire.ktir;
D(end,end) = D(end,end) + pTire.dtir;
M(end,end) = M(end,end) + pTire.mtir;

tmp_id = length(K(:,1));
tmp_Masse = pTire.mtir;
[M_total,S_total,I_total] = SIMh.RB_Inertia(GeoTab,DofTab,idx,tmp_Mass,
    M_total,S_total,I_total);

%% Incorporate static constraints

index = [Dof_stat];
M(index,:)=[]; M(:,index)=[];
K(index,:)=[]; K(:,index)=[];
D(index,:)=[]; D(:,index)=[];
DofTab(index,:) = [];

%% Creating state space model and computing EF/EV of the damped SYSTEM

```

```
A = [zeros(size(M)) eye(size(M)); -M\K -M\D];
B = [zeros(size(M)); inv(M)];
u = zeros(size(M,1),1); u(end,1) = 1;
p = zeros(size(M,1),1); p(20) = 1; p(21) = -1;
Bu = B*u; Bp = B*p;

[Psi,Lambda] = eig(A);

Lambda = diag(Lambda); [~,idx] = sort(Lambda); Lambda = Lambda(idx);
Psi = Psi(:,idx);

idx1 = find(imag(Lambda)>0); idx2 = find(imag(Lambda)<0);
Lambda = Lambda([idx1;idx2]);
Psi = Psi(:,[idx1;idx2]);
clear idx1 idx2

idx = find(imag(Lambda)>0);
wn = abs(Lambda(idx)); fn = wn/(2*pi); Damp = -1*real(Lambda(idx))./wn;

%% Computing Eigenvalus for the damped system with "retracted" gear

idx = 19;
M_op =  M(1:idx,1:idx); D_op =  D(1:idx,1:idx); K_op =  K(1:idx,1:idx);
% Add gear mass to the beam at node 3
mg = me_2_u(1,1) + me_2_l(1) + pTire.mtir;
idx = 3;
M_op(idx,idx) = M_op(idx,idx) + mg;

A_op = [zeros(size(M_op)) eye(size(M_op)); -M_op\K_op -M_op\D_op];
[Psi_op,Lambda_op] = eig(A_op);

Lambda_op = diag(Lambda_op); [~,idx] = sort(Lambda_op); Lambda_op = Lambda_op(idx);
Psi_op = Psi_op(:,idx);
idx1 = find(imag(Lambda_op)>0); idx2 = find(imag(Lambda_op)<0);
Lambda_op = Lambda_op([idx1;idx2]);
Psi_op = Psi_op(:,[idx1;idx2]);
idx = find(imag(Lambda_op)>0);
wn_op = abs(Lambda_op(idx));
fn_op = wn_op/(2*pi);
Damp_op = -1*real(Lambda_op(idx))./wn_op;
clear idx1 idx2 idx;

%% Initial Conditions

u0 = zeros(size(A,1),1);
z0 = 0; % initial condition of the friction modell
x0 = [u0;z0];

%% Static analysis to adjust parameter val.Fc

K_static = K((end-2):end,(end-2):end);
p_static = zeros(size(K_static,1),1); p_static(end) = 1;
ub = 0.01;
x_static = inv(K_static) * p_static * pTire.ktir * ub;
```

```

du = x_static(end-2) - x_static(end-1);
F_K0 = lugr.sigma0*du;
F_K1 = lugr.k1*du;

%% Create/Load random excitation vector

delta = fn.*Damp; nfreq = 6; % how many of the lowest eigenfrequencies shall be covered by the random excitation
Nt = 10; % adjust simulation time
[u_rand,dt,fs] = SIMh.randSignal(H0,delta,fn,nfreq,Nt);
maxDisp = 0.025;
scalef = maxDisp/max(u_rand); u_rand = u_rand' * scalef; % rescaling the random vector
N = length(u_rand); t_rand = [0:(N-1)]*dt; % Computing number of samples and time vector for random signal
v_rand = SIMh.centralDiffQuot(u_rand,t_rand,dt);
v_rand = v_rand';

%% Integration with FastRK
exc = 22;
tic;
[tout,xout,Fout] = SIMh.FastRK4(A,B,u_rand,dt,u0,z0,exc,H0,ner,val,pTire);
time_RK4 = toc;

%% PLOTTING %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

friction_state = Fout;
urel = xout(:,20)-xout(:,21); urel_max = max(urel);

subplot(2,1,1)
h = plot(tout,urel);
set(h,'LineWidth',0.002);
title('Relative Displacement Landing Gear')
xlabel('Time [s]')
ylabel('Displacement  [m]')

subplot(2,1,2)
h = plot(tout,friction_state);
set(h,'LineWidth',0.002);
XL = get(gca,'Xlim');
hl(1) = line(XL,lugr.Fc*[1 1]);
hl(2) = line(XL,-lugr.Fc*[1 1]);
set(hl,'Color','r','LineStyle','--','LineWidth',2)
title('Force within the friction element')
xlabel('Time [s]')
ylabel('Force [N]')

figure(2)
h = plot(urel,friction_state,'Color',[0.3010 0.7450 0.9330]);
set(h,'LineWidth',0.2);
title('Force-Dispalcement-Diagram')
xlabel('Relativ displacement [m]')
ylabel('Force [N]')
```

```
grid on

figure(3)
sgtitle('Random excitation function and its distribution')
subplot(1,2,1)
plot(t_rand,u_rand,'Color',[0.3010 0.7450 0.9330])
xlabel('Time [s]')
ylabel('Displacement [m]')
grid on
subplot(1,2,2)
h = histogram(u_rand,'FaceColor',[0.3010 0.7450 0.9330],'EdgeColor',[0.3010 0.7450 0.9330]);
ylabel('Number of samples [-]')
set(gca,'View',[90 270])
xlim([-0.025 0.025])
grid on

%% Statistics of random excitation

rand_stat.standard_dev = std(u_rand);
rand_stat.mean = mean(u_rand);
rand_stat.crest = max(abs(u_rand))/std(u_rand);

%% Compute acceleration through y in state space model

stiffness_rod = pRod.E_u * pRod.A_u / pRod.l_u;

Cs = [eye(size(M)) zeros(size(M)); zeros(size(M)) eye(size(M)); -M\K -M\D];
Ds = [zeros(size(M)); zeros(size(M)); inv(M)]; q = zeros(size(M,1),1); q(exc) = 1;

cf = zeros(1,2*size(M,1)); idx = [4 20];

y = Cs*xout' + Ds(:,exc)*(pTire.ktir*u_rand) + Ds*p*Fout';
y(end+1,:) = u_rand;
```

A.2 rod2dof.m

```
function [me,ke] = beam4dof(E,I,l,A,rho)

ke = E*A/l * [1 -1; -1 1];

me = rho*A*l*[1/3 1/6; 1/6 1/3];

end
```

A.3 beam4dof.m

```
function [me,ke] = beam4dof(E,I,l,A,rho)

ke = E*I/(l^3) * [12, 6*l,  -12, 6*l;
    6*l,4*l^2,-6*l,2*l^2;
    -12,-6*l, 12,  -6*l;
    6*l,2*l^2,-6*l,4*l^2];
```

```

me = rho*A*l/420*[156,  22*l,  54,  -13*l;
    22*l, 4*l^2, 13*l,-3*l^2;
    54,   13*l,  156, -22*l;
    -13*l,-3*l^2,-22*l,4*l^2];

end
```

B Pseudo-Random Excitation Generator

This section introduces a function to be able to generate pseudo-random excitation function.

```
function [u_rand,dt,fs] = randSignal(HO,delta,fn,nfreq,Nt)
% delta:  fn*damp (Abklingkonstante) to determine sample step size
% fn:     Vector containing all eigenfrequencies
% nfreq:  Number up to which EF the signal covers the excited frequencies
% Nt:     Number of Ts, for evaluating with windows

switch HO % oversampling factor
    case {0,'constant'}, k0 = 2.841;    % see D.J. Murray-Smith (1995)
    case {1,'linear'}, k0 = 4.016;
    case {3,'spline'}, k0 = 1.572;
    otherwise
        error('Unknown hold order "%s"',string)
end

fs = round(2*max(fn)*k0);

dt = 1/fs;

amplitude = 1/120;
offset = 0; % constant part (for random signal usually =0)

Bandwidth = 2*min(delta)/(2*pi);

df = Bandwidth / 6; % points per peak
T = Nt * 1/df; % T=10, for evaluating with windows
N = T/dt;
N = 2*floor(N/2);

df = fs/(N-1);
freq = df*[0:(N-1)]';

f_dach = zeros(N/2,1);
index = find(freq>=0.01 & freq<=(fn(nfreq)*1.1));
amp = 0.5*amplitude*ones(size(index));
phase = 2*pi*(rand(size(index))-0.5);
f_dach(index) = amp.*exp(1i*phase);
f_dach(1) = offset;

f_dach_conj = conj(flipud(f_dach(2:N/2)));
f_dach = [f_dach;0;f_dach_conj];

f   = N*ifft(f_dach); u_rand   = real(f);
end
```

C A basic Stochastic Subspace Identification algorithm

This section contains a simple SSI implementation, taken from Otto [2021]. These function shall solely serve as an example for how an implementation could look like. As it is mentioned in the critical reflection, within this work the DLR toolbox is used to increase the functionality. Anyway, file main.m computes the matrices **A**, **C** and **G**. These are used within modalparams.m to extract the modal parameters.

C.1 main.m

```
function [A,C,G,R0] = ssidata(Y,order,s)
% [A,C,G,R0] = SSIDATA(Y,order,s)
%   Data-based stochastic subspace identification (SSI-data).
%
%   INPUTS:
% Y       sensor data matrix
%   order desired maximum model order to identify (scalar)
%   s       number of block rows in the block hankel matrix, should be at
%           least ceil(order/ns) to obtain results up to the desired model
%           order, generally recommended that s > order
%
%   OUTPUTS:
%   A       cell array of state transition matrices for model orders {i}
%   C       cell array of output matrices for model orders {i}
%   G       cell array of next state output covariance matrices for model
%           orders {i}
%   R0      zero-lag output covariances
%
% REFERENCES:
% [1]   "Subspace Identification for Linear Systems" by Peter van Overschee
%       and Bart de Moor, doi:10.1007/978-1-4613-0465-4
% [2] "System Identification Methods for (Operational) Modal Analysis:
%       Review and Comparison" by Edwin Reynders in Archives of
%       Computational Methods in Engineering, Vol. 19, No. 1,
%       doi:10.1007/s11831-012-9069-x
% [3]   "Operational Modal Analysis of Civil Engineering Structures" by
%       Carlo Rainieri and Giovanni Fabbrocino,
%       doi:10.1007/978-1-4939-0767-0
%
% NOTES:
% (1)   this implementation can easily consume a lot of memory for large
%       sensor records or numerous time lags, iterative algorithms
%       operating on the sensor data to create the projection are
%       recommended and will be the focus of future updates

disp('SSI-data status:')

[r,c] = size(Y);
if r > c % make sure y is shaped correctly with samples going across rows
    Y = Y';
end

```

```
[ns,nt] = size(Y); % ns = # of sensors, nt = # of samples

% shifted data matrix
disp('  forming shifted data matrix...')
Yh = zeros(ns*2*s,nt-2*s+1);
for i = 1:2*s % go down block rows of the Hankel data matrix
    Yh((i-1)*ns+1:i*ns,:) = Y(:,i:nt-2*s+i); % fill out the entire row
end
Yh = Yh/sqrt(nt-2*s+1);

% QR decomposition and projection of raw data
disp('  projecting raw data...')
R = triu(qr(Yh'))';
R = R(1:2*s*ns,1:2*s*ns);
Proj = R(ns*s+1:2*ns*s,1:ns*s);

% SVD (no weighting = balanced PC)
disp('  performing singular value decomposition...')
[U,S,~] = svd(Proj);
S = diag(S);

% zero lag output covariance
R0 = R(ns*s+1:ns*(s+1),:)*R(ns*s+1:ns*(s+1),:)';

% output cell arrays
A = cell(1,order);
C = cell(1,order);
G = cell(1,order);

% loop over model orders and generate system matrices
disp(['  generating system matrices A,C,G for ' num2str(order) ' model orders...'])
for i = 1:order
    U1 = U(:,1:i);
    gam   = U1*diag(sqrt(S(1:i)));
    gamm = U1(1:ns*(s-1),:)*diag(sqrt(S(1:i)));
    gam_inv   = pinv(gam);
    gamm_inv = pinv(gamm);
    A{i} = gamm_inv*gam(ns+1:ns*s,:); % state transition matrix
    C{i} = gam(1:ns,:); % output matrix
    delta = gam_inv*(R(ns*s+1:2*ns*s,1:ns*s)*R(1:ns*s,1:ns*s)');
    G{i} = delta(:,ns*(s-1)+1:ns*s); % next state output covariance matrix
end

disp('SSI-data finished.')


end
```

C.2 modalparams.m

```
function [f,zeta,Phi] = modalparams(A,C,dt)
% [f,psi,Phi] = MODALPARAMS(A,C,dt)
% Modal decomposition of discrete state space system.
%
```

```
%   INPUTS:
%   A       cell array of system matrices for model order {i}
%   C       cell array of output matrices for model order {i}
%   dt      sampling period of the discrete system A,C
%
%   OUTPUTS:
%   f       cell array containing the system pole frequencies in Hz for
%           model orders {i}
%   psi     cell array containing the damping ratios of each pole for
   model
%           orders {i}
%   Phi     cell array containing the mode shape vectors for model orders
%           {i}

% NOTES:
% (1) modal scaling (normalization) is not performed
% (2)   complex conjugate pairs are eliminated and modes are sorted by
%       frequency

% check for cell array input
if ~iscell(A)
    A = {A};
end
if ~iscell(C)
    C = {C};
end

f = cell(size(A));
zeta = cell(size(A));
Phi = cell(size(A));

% loop over model orders
for i = 1:length(A)
    [v,d] = eig(A{i});
    lam = log(diag(d))/dt;
    f{i} = abs(lam)/2/pi; % modal frequencies (Hz)
    [f{i},I] = sort(f{i}); % sort using ascending frequencies
    zeta{i} = -real(lam)./abs(lam); % modal damping ratios
    zeta{i} = zeta{i}(I);
    Phi{i} = C{i}*v; % mode shapes
    Phi{i} = Phi{i}(:,I);
    % eliminate complex conjugate pairs
    [f{i},I] = unique(f{i});
    zeta{i} = zeta{i}(I);
    Phi{i} = Phi{i}(:,I);
end



end
```

D Algorithms for Nonlinear System Identification

This last section presents the codes corresponding to the developed and investigated algorithms for nonlinear modal analysis. All codes are formulated within the syntax of the DLR toolbox.

D.1 Standard deviation-based windowing technique

```
function sorted_data = sdbwt(td,ns,s,overlapping,nlvl,lb)

% td:            TimeD Object containing system output
% ns:            Numbers sample per window
% overlapping:   Overlapping factor
% nlvl:          Number of levels
% lb:            Level boundary (-lb*sigma:lb*sigma)

rel_var = td.DataY(:,20) - td.DataY(:,21);  % compute relative displacement
t = td.DataX;

numberwindows = floor((numel(rel_var)-ns)/(ns*(1-overlapping)));

s = 1;
for k = 1:numberwindows

    if k == 1
        s = 1;
    else
        s = (k-1)*(1-overlapping)*ns; s = round(s);
    end

    y = rel_var(s:s+ns);
    y_rms = std(y);

    if k == 1
        YY = y;
        RMS = y_rms;
    else
        RMS(:,end+1) = y_rms;
    end

end
RMS = round(RMS,4,'decimals');
histogram(RMS)

%% find RMS

mu = mean(RMS);
sigma = std(RMS);
level = linspace(-2.5*sigma,2.5*sigma,nlevel) + mu;
level = round(level,4,'decimals');

for k = 1:numel(level)
```

```

    idx = find(RMS == level(k));

    for q = 1:numel(idx)
        s = (idx(q)-1) * (1-overlapping)*ns;

        if s == 0; DataY = td.DataY(1:ns,:); end

        if idx(q) == 1; s = 1;
        else s = (idx(q)-1) * (1-overlapping)*ns; s = round(s);
        end

        DataY = td.DataY(s:s+ns,:);

        if q == 1; YYY = DataY;
        else %YYY = [YYY; DataY]; % TIME CONSUMING!!!
            YYY(end+1:end+size(DataY,1),:) = DataY;
        end
    end
    Data(k).DataY = YYY;
end

for n = 1:numel(level)
    Td(n).td = TimeD;
    tvec(n).tvec = [1:size(Data(n).DataY,1)]*dt;
    Td(n).td.DataY = Data(n).DataY;
    Td(n).td.DataX = tvec(n).tvec;
    Td(n).td.ChTable = td.ChTable;
    Td(n).td.Header = td.Header;
end
end
```

D.2 Rainflow counting algorithm-based identification

```
function sorted_data = rca(td,crit,lb)

% td:   TimeD Object containing system output
% crit: Threshold for min number of samples
% lb:   Level boundary (-lb*sigma:lb*sigma)

rel_var = td.DataY(:,20) - td.DataY(:,21); % compute relative displacement
t = td.DataX;

[R,rm] = rainflow(rel_var); % Matlabs Rainflow
%OUTPUT: R = [CycleCount Range Mean SampleStart SampleEnd]

%% Clean from cycles with to less samples
Rc = zeros(1,5);

for k = 1:size(R,1)
    samp_diff = R(k,5)-R(k,4);
    if samp_diff > crit
        Rc(end+1,:) = R(k,:);
    end
end
```

```

Rc(1,:) = [];

%% Defining levels and searching for them

sigma = std(Rc(:,3));
mu = mean(Rc(:,3));
nlevel = 10;
level = linspace(-2.5*sigma,2.5*sigma,nlevel) + mu;

% search for series which fall into the defined level and sort signal

for ii = 1:numel(level)-1
    Data(ii).DataY = [];
end

for k = 1:size(Rc,1)
    for i = 1:numel(level)-1
        if Rc(k,3) > level(i) && Rc(k,3) < level(i+1)
            %store ref signal and system outputs
            if isempty(Data(i).DataY) == 1
                Data(i).DataY = td.DataY(Rc(k,4):Rc(k,5),:);
            else
                %Data(i).DataY = [Data(i).DataY;
                %td.DataY(R(k,4):R(k,5),:)]; % TIME CONSUMING! Better next
    line
                b = td.DataY(Rc(k,4):Rc(k,5),:);
                s = size(td.DataY(Rc(k,4):Rc(k,5),:),1);
                Data(i).DataY(end+1:end+s,:) = b;
            end
        end
    end
end

%% Create for each trigger level band a timeD object

for n = 1:numel(level)-1
    Td(n).td = TimeD;
    dt = td.DataX(2);
    tvec(n).tvec = [1:size(Data(n).DataY,1)]*dt;
    Td(n).td.DataY = Data(n).DataY;
    Td(n).td.DataX = tvec(n).tvec;
    Td(n).td.ChTable = td.ChTable;
    Td(n).td.Header = td.Header;
end
end
```

D.3 Advanced Stochastic Subspace Identification

```
function [U,S,R,R0,ns] = x_ssi(Y,order,ReferenceSignal,SelectedLevel,nlvl,
    dt)

s = order(1); % Block Size

[ns,nt] = size(Y); % ns = # of sensors, nt = # of samples
```

```
if ns > nt % make sure y is shaped correctly
    Y = Y';
    [ns,nt] = size(Y);
end

% shifted Hankel matrix
Yh = zeros(ns*2*s,nt-2*s+1);
for i = 1:2*s % go down block rows of the Hankel data matrix
    Yh((i-1)*ns+1:i*ns,:) = Y(:,i:nt-2*s+i); % fill out the entire row
end
Yh = Yh/sqrt(nt-2*s+1);

% shifted Hankel matrix for reference signal
[rs,rt] = size(ReferenceSignal);
if rs > rt % make sure y is shaped correctly with samples going across rows
    ReferenceSignal = ReferenceSignal';
end
[rs,rt] = size(ReferenceSignal);

Yref = zeros(rs*2*s,rt-2*s+1);
for i = 1:2*s % go down block rows of the Hankel data matrix
    Yref((i-1)*rs+1:i*rs,:) = ReferenceSignal(:,i:rt-2*s+i); % fill out the entire row
end
Yref = Yref/sqrt(rt-2*s+1);
% compute statistical properties
Yref_std = std(Yref);
Yref_mu  = mean(Yref_std);
Yref_sig = std(Yref_std);

lvl = linspace(-2.5*Yref_sig,2*Yref_sig,10) + Yref_mu;

idx    = (Yref_std<lvl(SelectedLevel+1) & Yref_std>=lvl(SelectedLevel));

Yh_red = Yh(:,idx); % reduced Hankel matrix

% QR decomposition and projection of raw data
R = triu(qr(Yh_red'))';
R = R(1:2*s*ns,1:2*s*ns);
Proj = R(ns*s+1:2*ns*s,1:ns*s);

% SVD (no weighting = UPC)
[U,S,~] = svd(Proj);
S = diag(S);

% zero lag output covariance
R0 = R(ns*s+1:ns*(s+1),:)*R(ns*s+1:ns*(s+1),:)';

end

function [A,C,delta,G] = x_ssifun(N,order,ns,U,S,R)
% Observable and Controllable Subspaces
s = order(1); % Block Size

```

```
U1 = U(:,1:N);
gam   = U1*diag(sqrt(S(1:N)));
gamm = U1(1:ns*(s-1),:)*diag(sqrt(S(1:N)));
gam_inv   = pinv(gam);
gamm_inv = pinv(gamm);

A = gamm_inv*gam(ns+1:ns*s,:); % state transition matrix
C = gam(1:ns,:); % output matrix
delta = gam_inv*(R(ns*s+1:2*ns*s,1:ns*s)*R(1:ns*s,1:ns*s)');
G = delta(:,ns*(s-1)+1:ns*s); % next state output covariance matrix

end

function [x_dot] = centralDiffQuot(x,dt)

x_dot = zeros(length(x),1);
n = length(x);
warning('CD works only with constant time steps!')

for i = 1:n  % t is timevector

    if i == 1; xm2h = x(1); elseif i == 2; xm2h = x(i-1); else xm2h = x(i-2); end
    if i == 1; xm1h = x(1); else xm1h = x(i-1); end
    if i == n-1; xp2h = x(n); elseif i == n; xp2h = x(n); else xp2h = x(i+2); end
    if i == n; xp1h = x(n); else xp1h = x(i+1); end

    x_dot(i,1) = (-xp2h + 8*xp1h - 8*xm1h + xm2h)/(12*dt);
end
end
```

References

S.A.S Airbus. Airbus a320 aircraft characteristics airport and maintenance planning. 2005.

K. Atkinson, W. Han, and D. Stewart. *Numerical solution of ordinary differential equations*. Pure and applied mathematics. Wiley, 2009. ISBN 978-0-470-04294-6.

K.-J. Bathe. *Finite Element Procedures*. Pearson Education, 2014. ISBN 978-0-9790049-5-7.

J. Bendat and A Piersol. *Random Data - Analysis and Measurement Procedures*. 2010. ISBN 978-0-470-24877-5. doi: 10.1002/9781118032428.

J. Bianchi, E. Balmes, G. Vermot Des Roches, and A. Bobillot. Using modal damping for full model transient analysis. application to pantograph/catenary vibration. 2010.

A. Brandt. *Noise and vibration analysis: signal analysis and experimental procedures*. Wiley, 2011. ISBN 978-0-470-74644-8. doi: 10.1002/9780470978160.

R. Brincker and P. Andersen. Understanding stochastic subspace identification. 2006.

M. Böswald. *Updating of Local Non-Linear Stiffness- and Damping Parameters in Large Order Finite Element Models by Using Vibration Test Data*. 2006.

M. Böswald. Analysis of the bias in modal parameters obtained with frequency-domain rational fraction polynomial estimators. 2016. doi: https://elib.dlr.de/108128/.

M. Böswald and Y. Govers. Taxi vibration testing - an alternative method to ground vibration testing of large aircraft. 2008. doi: https://elib.dlr.de/56120/.

M. Böswald, D. Göge, U. Füllekrug, and Y. Govers. A review of experimental modal analysis methods with respect to their applicability to test data of large aircraft structures. 2006.

M. Böswald, M. Höser, and Y. Govers. Aufbau globaler dämpfungsmatrizen aus modalen dämpfungsmaßen von substrukturtests. 2016. doi: https://elib.dlr.de/107992/.

M. Böswald, J. Schwochow, G. Jelicic, and Y. Govers. Recent developments in operational modal analysis for ground and flight vibration testing. 2017.

C. Canudas de Wit, H. Olsson, K.J. Astrom, and P. Lischinsky. A new model for control of systems with friction. *IEEE Transactions on Automatic Control*, 40(3):419–425, 1995. ISSN 00189286. doi: 10.1109/9.376053. URL `http://ieeexplore.ieee.org/document/376053/`.

T. Endo, K. Mitsunaga, K. Takahashi, K. Kobayashi, and M. Matsuishi. Damage evaluation of metals for random or varying loading - three aspects of rain flow method. 1974.

R. Freymann. *Strukturdynamik: Ein anwendungsorientiertes Lehrbuch*. Springer, 2011. ISBN 978-3-642-19697-3 978-3-642-19698-0. doi: https://doi.org/10.1007/978-3-642-19698-0.

R. Gasch, K. Knothe, and R. Liebich. *Strukturdynamik: Diskrete Systeme und Kontinua.* Springer-Vieweg, 2., neu bearb. aufl., korrigierte neuaufl edition, 2012. ISBN 978-3-540-88976-2 978-3-540-88977-9. doi: 10.1007/978-3-540-88977-9.

L. Gaul and R. Nitsche. The role of friction in mechanical joints. *Applied Mechanics Reviews*, 54(2):93–106, 2001. ISSN 0003-6900, 2379-0407. doi: 10.1115/1.3097294.

P. Guillaume, P. Verboven, S. Vanlanduit, H. Van der Auweraer, and B. Peeters. A poly-reference implementation of the least-squares complex frequency-domain estimator. 2003.

S. Han and H. Benaroya. *Nonlinear and Stochastic Dynamics of Compliant Offshore Structures.* Springer Netherlands, 2002. doi: 10.1007/978-94-015-9912-2.

T.K. Hasselman, J.D. Chrostowski, and R. Pappa. Estimation of full modal damping matrices from complex test modes. 1993. doi: https://doi.org/10.2514/6.1993-1668.

D. Henwood and J. Bonet. *Finite Elements - A Gentle Introduction.* 1996.

R. Horn and C. Johnson. *Topics in Matrix Analysis.* Cambridge University Press, 1991. ISBN 9780511840371. doi: https://doi.org/10.1017/CBO9780511840371.

G. Jelicic, M. Böswald, and A. Brandt. Improved computation in terms of accuracy and speed of lti system response with arbitrary input. 150: 107252, 2020. ISSN 08883270. doi: 10.1016/j.ymssp.2020.107252. URL `https://linkinghub.elsevier.com/retrieve/pii/S0888327020306385`.

J.-N. Juang. *Applied System Identification.* 1993. ISBN 978-0130792112.

G. Kerschen. *Nonlinear Dynamics, Volume 1: Proceedings of the 35th IMAC, A Conference and Exposition on Structural Dynamics 2017.* Springer International Publishing, 2017. ISBN 978-3-319-54404-5. doi: 10.1007/978-3-319-54404-5.

M. Lakshmanan and S. Rajasekar. *Nonlinear Dynamics.* Springer, 2003. ISBN 978-3-642-62872-6. doi: 10.1007/978-3-642-55688-3.

B. H. K. Lee and A. Tron. Effects of structural nonlinearities on flutter characteristics of the cf-18 aircraft. *Journal of Aircraft*, 26(8):781–786, 1989. ISSN 0021-8669, 1533-3868. doi: https://doi.org/10.2514/3.45839. URL `https://arc.aiaa.org/doi/10.2514/3.45839`.

M. Link. *Finite Elemente in der Statik und Dynamik.* Springer Vieweg, 4., korrigierte aufl edition, 2014. ISBN 978-3-658-03556-3 978-3-658-03557-0. doi: 10.1007/978-3-658-03557-0.

N. Londono, S. Desjardins, and D. Lau. Use of stochastic subspace identification methods for post-disaster condition assessment of highway bridges. 2004.

R. Markert. *Strukturdynamik.* Shaker, 2016. ISBN 978-3-8440-2098-4.

D. J Murray-Smith. *Continuous System Simulation.* Springer-Science, 1995. ISBN 978-1-4615-2504-2. URL `https://doi.org/10.1007/978-1-4615-2504-2`.

A. Otto. Ooma toolbox (https://www.mathworks.com/matlabcentral/fileexchange/68657-ooma-toolbox). 2021.

A Papoulis. *Probability, Random Variables and Stochastic Processes.* 2002. ISBN 0-07-366011-6.

M. Paz and W. Leigh. *Structural Dynamics: Theory and Computation.* Springer US, Boston, MA, 2004. ISBN 978-1-4613-5098-9 978-1-4615-0481-8. doi: 10.1007/978-3-319-94743-3.

B. Peeters and G. De Roeck. Reference-based stochastic subspace identification for output-only modal analysis. 1999. doi: https://doi.org/10.1006/mssp.1999.1249.

R. Pintelon and J. Schoukens. *System Identification: A Frequency Domain Approach.* Wiley, 2012. ISBN 9781118287422.

V. Pletser. *Lagrangian and Hamiltonian analytical mechanics.* Springer Berlin Heidelberg, New York, NY, 2018. ISBN 9789811330254. doi: 10.1007/978-981-13-3026-1.

A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics.* 2007. ISBN 0939-2475. doi: 10.1007/b98885.

C. Rossow, K. Wolf, and P. Horst. *Handbuch der Luftfahrzeugtechnik.* Carl Hander Verlag, 2014. ISBN 978-3-446-43604-6. doi: https://doi.org/10.3139/9783446436046.fm.

I. Rychlik. A new definition of the rainflow cycle counting method. 1987. doi: https://doi.org/10.1016/0142-1123(87)90054-5.

J. Schoukens, R. Pintelon, Y. Rolain, and T. Dobrowiecki. Frequency response function measurements in the presence of nonlinear distortions. *Automatica*, 2000. doi: https://doi.org/10.1016/S0005-1098(01)00037-1.

J. Schoukens, A. Marconato, R. Pintelon, Y. Rolain, M. Schoukens, K. Tiels, G. Vanbeylen, G. Vandersteen, and A. Van Mulders. System identification in a real world. *Advanced Motion Control*, 2014. doi: 10.1109/AMC.2014.6823250.

K. Soal. System identification and modal tracking on ship structures. 2018. doi: https://scholar.sun.ac.za/handle/10019.1/103788.

K. Soal, Y. Govers, M. Böswald, and A. Vollmer. Taxi vibration testing: A new and time efficient procedure for the identification of modal parameters on aircrafts. 2019.

L.T. Tenek and J. Argyris. *Finite Element Analysis for Composite Structures.* Springer Netherlands, 1998. ISBN 978-90-481-4975-9. doi: 10.1007/978-94-015-9044-0.

K.C. Valanis. A theory of viscoplasticity without a yield surface. *Archive of Mechanics*, (3): 517–551, 1971.

P. Van Overschee and B. De Moor. *Subspace identification for linear systems theory, implementation, applications.* Springer, 1996. ISBN 978-1-4613-8061-0. doi: 10.1007/978-1-4613-0465-4.

K.K. Vesterholm, R. Brincker, and A. Brandt. Linearization of modal parameter in duffing oscillator using the random decrement technique. 2018.

J. Wallaschek. *Skript zur Vorlesung Maschinendynamik*. Number 1. Gottfried Wilhelm Leibniz Universität Hannover, 2018.

P. Wriggers. *Nichtlineare Finite Element Methoden*. Springer, Berlin, 2001. ISBN 978-3-540-67747-5. doi: 10.1007/978-3-642-56865-7.