# Linear Parameter-Varying Model Identification for Flutter Prediction

# M. Visser





Delft Center for Systems and Control

Cover photo  ${\ensuremath{\mathbb O}}$  Hans Hillewa<br/>ert

# Linear Parameter-Varying Model Identification for Flutter Prediction

MASTER OF SCIENCE THESIS

For the degree of Master of Science in Mechanical Engineering at Delft University of Technology

M. Visser

June 22, 2015

Faculty of Mechanical, Maritime and Materials Engineering (3mE)  $\cdot$  Delft University of Technology





Copyright © Delft Center for Systems and Control (DCSC) All rights reserved.

## Abstract

Flutter-induced vibrations can easily cause large structures like airplanes and bridges to fail, which even today makes flutter prediction an important topic of research. To predict flutter, recursive system identification can be used to capture the time-varying behavior using time-invariant techniques combined with the forgetting of old data. However, forgetting older data takes time, which introduces a delay in the predictions.

In order to circumvent this prediction delay the use of Linear Parameter-Varying (LPV) model identification from pre-flutter data is proposed, as this directly identifies a time-varying model. This thesis will compare recently published *global*, *local*, and *glocal* LPV identification algorithms and assess their flutter prediction capabilities both in simulation and using experiments.

The simulation results show that using LPV model identification the flutter speed could be predicted to within 10 % even in the presence of significant noise. Furthermore, guidelines are given on how these methods can be used and what their limitations are. Regarding the experiments, flutter-induced vibrations could not be reproduced due to fundamental problems in the experimental setup for which recommendations are made.

# **Table of Contents**

	Ackı	nowledgements	ix
1	Intro	oduction	1
	1-1	Motivation	1
	1-2	Problem statement	3
	1-3	Methodology	3
	1-4	Contributions	4
	1-5	Structure of MSc thesis	4
2	First	t principles flutter modeling	5
	2-1	Aeroelasticity	5
		2-1-1 Divergence	6
		2-1-2 Flutter	7
	2-2	Modeling	8
		2-2-1 Model description	9
		2-2-2 LPV model	10
	2-3	Influence of design parameters on flutter	11
3	LPV	/ identification methods	13
	3-1	Introduction	13
		3-1-1 Global	13
		3-1-2 Local	14
		3-1-3 Glocal	15
	3-2	Methods	15
		3-2-1 Global: PBSID <sub>opt</sub>	15
		3-2-2 Local: SMILE	16
		3-2-3 Glocal: $\mathcal{H}_2$ norm	16
		3-2-4 Glocal: $\mathcal{H}_{\infty}$ norm	17
	3-3	Simulation scenario	18
		3-3-1 LTI identification	18
	3-4	Results	19

		3-4-1 Glocal $\mathcal{H}_\infty$ -based	19
		3-4-2 Flutter prediction	20
		3-4-3 Model reduction	22
		3-4-4 Computational aspects	23
	3-5	Synopsis	24
4	Exp	erimental setup validation	27
	4-1	Morphing airfoil	27
		4-1-1 Wind tunnel	29
	4-2	Electronically controlled suspension	29
		4-2-1 Concept	29
		4-2-2 Control scheme	31
		4-2-3 Encountered problems	32
	4-3	System dynamics	34
		4-3-1 Airfoil bending	34
		4-3-2 Influence of the wind	36
	4-4	Recommendations	38
	4-5	Synopsis	39
5	Exp	erimental identification	41
-	5-1	Data processing	41
	5-2	LTI identification	42
	5-3	LPV identification	44
		5-3-1 Global: PBSID <sub>opt</sub>	44
		5-3-2 Local: SMILE	46
		5-3-3 Glocal: $\mathcal{H}_2$ norm	47
		5-3-4 Comparison	48
	5-4	Synopsis	50
6	Con	clusions and recommendations	51
-	6-1	Recommendations for future research	53
	Bibl	liography	55
	Clea		50
		of Acronyms	- 59 - 50
	LIJU	or / Gionymo	00

# **List of Figures**

1-1 1-2	Examples of aeroelastically coupled structures	$\frac{1}{2}$
2-1	Pole frequency, damping, and location for $V \in [0, 9]$ m/s in the case of divergence.	6
2-2	Pole frequency, damping, and location for $V \in [0, 16]$ m/s in the case of flutter.	$\overline{7}$
2-3	Schematic representation of a 'smart' airfoil (Wingerden, 2008)	9
3-1 3-2	$\mathcal{H}_{\infty}$ -based glocal LPV estimate fits the <i>local</i> continuous-time LTI models very well. The <i>global</i> method shows the best flutter prediction, for low noise the poles fit very well. Furthermore, the variance of the estimates increases when more noise is	20
3-3	At low noise, the <i>global</i> method shows highest accuracy, while at higher noise values <i>glocal</i> methods show the best bias/variance trade-off.	21 21
3-4	Flutter prediction deteriorates significantly when a degree of freedom is removed $(\hat{n}_x = 2)$	23
4-1	Morphing airfoil with a deformable trailing edge	27
4-2	Photographs of the morphing airfoil used in the experiments.	28
4-3	Small visualization wind tunnel inside the lab.	29
4-4	Adaptive springs concept based on torque feedback control	30
4-5	Electronically controlled suspension setup.	31
4-6	Control scheme of the electronically controlled suspension.	32
4-7	A coarse signal resolution in combination with an overshoot around $I_2 = 0$ causes	
	vibrations	33
4-8	A laser distance sensor is used to measure the position of the airfoil which rotates	
	around the elastic axis indicated by the black dot	34
4-9	The bending mode of the airfoil is located at 8.2 Hz	35
4-10	Output spectra for a wind speed $V = 5 \text{ m/s}$ and different spring stiffness values $k_{\alpha}$ .	36
4-11	Output spectra for different wind speed values and $k_{\alpha} = 4.2$ , $k_{h} = 1000$ , $c_{\alpha} =$	
	$0.01, c_h = 0, \text{ and } \alpha_0 = 2.5^{\circ}.$	37

4-12	Output spectra for different wind speed values and $k_{\alpha} = 4.2$ , $k_{h} = 1000$ , $c_{\alpha} = 0.01$ , $c_{h} = 0$ , and $\alpha_{0} = 5^{\circ}$ .	37
5-1	Effect of the future window $f$ on the (discrete-time) pole locations for a range of wind speeds $V \in \{0, 2.5, 5, 7.5, 10\}$ m/s.	43
5-2	Comparison of the measured response for $V = 5$ m/s with the simulated response of the corresponding local Linear Time-Invariant (LTI) model.	44
5-3	Exponential increase in computational time for increasing past window $p_g$	45
5-4	The way in which the wind speed signal varies has a significant effect on the identified LPV model.	45
5-5	Comparison of the identified <i>global</i> LPV model poles, shown as circles, with the local LTI model poles, shown as dots.	46
5-6	Comparison of the identified <i>local</i> LPV model poles, shown as circles, with the	
	local LTI model poles, shown as dots.	47
5-7	Comparison of the identified <i>glocal</i> LPV model poles, shown as circles, with the local LTI model poles, shown as dots.	48
5-8	Pole prediction differences among LPV identification algorithms.	49
5-9	Comparison of the global LPV simulation output with the measured response	49

# List of Tables

2-1	Flutter model parameters (Lee and Singh, 2007)	10
3-1	$\mathcal{H}_\infty$ -based glocal method (Vizer et al., 2013) performance for two sampling times	
	$T_s$ (Signal to Noise Ratio (SNR) = 40)	19
3-2	Flutter speed prediction error for two state dimensions $\hat{n}_x \in \{2,4\}$ (%)	22
3-3	Variance Accounted For (VAF) of estimated LPV models for two state dimensions	
	$\hat{n}_x \in \{2,4\} \ (\%)$	23
3-4	Comparison of LPV identification algorithms	25

# Acknowledgements

This MSc thesis report is written as the final part of the requirement for obtaining the degree of Master of Science in Mechanical Engineering at the Delft Center for Systems and Control (DCSC).

I would like to thank my supervisors dr.ir. Jan-Willem van Wingerden and Sachin Navalkar, MSc for their guidance and feedback during this MSc project. Furthermore, I would like to thank Kees Slinkman for his work on the experimental setup and technical support while performing the experiments. Finally, I would like to thank the staff of the walk-in workshop at 3mE for their help with the fabrication of a morphing airfoil.

This MSc thesis is supervised by:

dr.ir. J.W. van Wingerden (DCSC) S.T. Navalkar, MSc (DCSC)

Delft, University of Technology June 22, 2015 M. Visser

# Chapter 1

# Introduction

## 1-1 Motivation

The interaction between a structure and the airflow passing it not only enables us to fly a 276,800 kg airplane, see Figure 1-1a, but also resulted in the catastrophic failure of a 1.6 km long bridge only four months after the construction was completed, see Figure 1-1b.



(a) British Airways Airbus A380, 2013



(b) Tacoma Narrows bridge just before collapse, 1940



This interaction of structures, like airplanes and bridges, with the aerodynamic forces is described by the field of aeroelasticity of which the failure of the Tacoma Narrows bridge in 1940, see Figure 1-1b, is perhaps the best known example. Today this event is still frequently mentioned in for example undergraduate physics textbooks as a good example of an aeroelastic instability. However, while this failure is often presented as an example of forced resonance, the evidence actually suggests that it is the result of the phenomenon of *flutter* (Billah and Scanlan, 1991).

On these catastrophic flutter-induced vibrations, Von Kármán once noted that "Some fear flutter because they do not understand it. And some fear it because they do." (Mukhopadhyay, 2003), which aptly describes their devastating consequences and the difficulty to prevent these vibrations.

Master of Science Thesis

When looking more closely at the phenomenon of flutter it occurs when the wind speed exceeds the *flutter speed* and can be described as a "self-excited vibration in which the structure extracts energy from the air stream which often results in catastrophic structural failure" (Wright and Cooper, 2008). This unstable nature is also illustrated in Figure 1-2, where the pole locations and damping of a two degree of freedom flutter model (Wingerden, 2008) for a range of wind speeds are shown. Clearly, for wind speeds exceeding the flutter speed (V > 12.4 (m/s)) a pole moves into the right half plane which results in a negative damping.



Figure 1-2: Pole location and damping for a two degree of freedom flutter model.

Structures like bridges and airplanes have a long history in coping with flutter (Mukhopadhyay, 2003). A more recent example are wind turbines, where due to the increasing rotor size, future designs are quickly approaching the flutter speed (Griffith and Ashwill, 2011; Resor et al., 2012; Owens et al., 2013).

Clearly then, flutter is still a current issue and the ability to predict the flutter speed is of the utmost importance to prevent or suppress the self-excited vibrations, for example by adding control surfaces to the structure to actively change the airflow (Karpel, 1981).

Naturally, predicting the behavior of a system requires the engineer to have a certain amount of knowledge about the system, which is often captured in dynamic models. While these models have been derived in the process of designing the structure, the final system will always deviate from the original model, which requires the engineers to tune their models using measurements from the real system.

Alternatively, one could identify a model entirely from measurements, such that only the dynamics which are present in the data are modeled, which is often referred to as *system identification* (Ljung, 1999). Given that the data is persistently exciting and contains few noise, such that all the relevant dynamics are present in the data, this will result in more accurate models within the frequency band of interest and therefore better predictions. Furthermore, system identification is a well-established methodology for Linear Time-Invariant (LTI) systems and various suitable algorithms are available in literature, see for example the overview given by Van der Veen et al. (2013).

Unfortunately, as shown in Figure 1-2, a flutter model does not satisfy the LTI assumption and the dynamics are clearly changing as a function of the (time-varying) wind speed. As a solution to this problem recursive system identification has been proposed, which introduces forgetting of older data to track slowly varying dynamics (Houtzager, 2011). However, as forgetting old data takes time these methods show a delay in their predictions, which was found to be between 50 and 100 s by Houtzager et al. (2012).

Alternatively, instead of using LTI methods to identify Linear Time-Varying (LTV) systems, non-linear system identification methods could be used. More specifically, the class of Linear Parameter-Varying (LPV) models has been shown to be very well suited to model flutter, see for example Lau and Krener (1999); Barker and Balas (2000); Wingerden (2008). Furthermore, apart from flutter prediction, the identified LPV models could also directly be used to design controllers for flutter suppression (Prime et al., 2010; Seiler et al., 2012).

Therefore, it is interesting to investigate how LPV system identification can be used in the application of flutter prediction.

## 1-2 Problem statement

The aeroelastic phenomenon of flutter has a clear time-varying nature, as the dynamics are a function of the wind speed. Recursive methods have been proposed to track such time-varying dynamics in real-time using LTI system identification techniques, but by their nature the predictions are always delayed. On the other hand, the use of batch LPV system identification directly results in time-varying dynamic models.

However, the batch identification scheme uses pre-flutter data only, after which the LPV model is evaluated at higher wind speeds to predict the point of instability, that is the flutter speed.

Furthermore, in recent years multiple LPV identification algorithms have been proposed in literature, see for example Van Wingerden and Verhaegen (2009); Petersson (2013); Vizer et al. (2013); De Caigny et al. (2014), but often validation takes place by looking at the results inside the domain of the scheduling sequence. In the case of flutter prediction, the scheduling sequence only contains wind speeds smaller than the flutter speed, while the actual validation takes place by evaluating the LPV model outside of this region.

Therefore, the main emphasis of this MSc thesis is to compare LPV batch identification methods and assess their potential in flutter prediction using pre-flutter data.

## 1-3 Methodology

The comparison of LPV batch identification methods will be divided into two main parts, namely a *theoretical* and *experimental* part.

The *theoretical* part will first describe the phenomenon of flutter and present an aeroelastic LPV flutter model which will also be used to generate data to validate the identification algorithms, as flutter prediction is the main motivation for this work. Secondly, the topic of LPV identification will be introduced and a division is made between *global*, *local* and *glocal* 

methods (Lovera et al., 2013). Next, a representative set of algorithms is selected from these three groups and the estimation accuracy is assessed in simulation.

In the *experimental* part, an experimental setup consisting of a morphing airfoil located inside a wind tunnel is used to generate the data for the LPV batch identification algorithms. In this way, the applicability of batch LPV identification when using experimental data will be validated.

### 1-4 Contributions

The contents of Chapter 3, which involves the comparison of LPV identification algorithms for flutter prediction using simulation data, have also been included in a conference paper (Visser et al., 2015):

Visser, M., Navalkar, S., and van Wingerden, J.-W. (2015). LPV model identification for flutter prediction: A comparison of methods. In *Proceedings of the 1st IFAC Workshop on Linear Parameter Varying Systems* 

### **1-5** Structure of MSc thesis

This MSc thesis is organized as follows:

- Chapter 2: First principles flutter modeling In this chapter the topic of aeroelasticity is discussed and an LPV flutter model is presented.
- **Chapter 3: LPV identification methods** This chapter introduces the field of LPV system identification and divides the available algorithms into *global*, *local*, and *glocal* methods. Next, these methods are validated using simulation data obtained from the LPV flutter model which was presented in Chapter 2.
- **Chapter 4: Experimental setup validation** Before performing LPV identification on the experimental data, the morphing airfoil setup is presented and problems are highlighted and design improvements are proposed.
- **Chapter 5: Experimental identification** Measurements are performed on the experimental setup and the performance of the LPV identification algorithms is assessed using this data.
- Chapter 6: Conclusions and recommendations This MSc thesis concludes the report with a set of conclusions and suggestions for future work.

# Chapter 2

## First principles flutter modeling

Flutter has been introduced in Section 1-1 very briefly, so as to give a concise motivation for this research. This chapter discusses the topic of flutter more in depth, before continuing to the actual identification step in the next chapter.

The chapter will start with an introduction to the field of aeroelasticity along with an explanation of the two most common instabilities, namely divergence and flutter. Once these aeroelastic phenomenon are explained, an LPV model is presented which describes the dynamics of an airfoil as a function of the wind speed and the influence of the model parameters on the occurrence of flutter is considered.

## 2-1 Aeroelasticity

The combination of the field of aerodynamics and structural dynamics is also referred to as aeroelasticity (Bisplinghoff et al., 1996) and can further be divided into *static* and *dynamic* aeroelasticity.

Static aeroelasticity is "the study of the deflection of flexible aircraft structures under aerodynamic loads, where the forces and motions are considered to be independent of time" (Wright and Cooper, 2008). An important static aeroelastic instability is *divergence*, where the moments due to aerodynamic forces overcome the restoring moments due to structural stiffness. While divergence causes structural failure, it is often of minor importance, as the speed at which divergence occurs is usually higher than the flutter speed for modern aircraft (Fung, 1993). Which brings us to the second type of aeroelastic instabilities.

Dynamic aeroelasticity is concerned with the oscillatory effects of the aeroelastic interactions (Wright and Cooper, 2008) of which the phenomenon of flutter is the most important example.

Both the *static* aeroelastic phenomenon of divergence and the *dynamic* aeroelastic phenomenon of flutter will be discussed in the next section and the characteristics of these aeroelastic instabilities are presented.

#### 2-1-1 Divergence

As stated, divergence occurs when the moments due to aerodynamic forces overcome the restoring moments due to structural stiffness. In order to see how this phenomenon translates in the pole frequency, damping, and location for a range of wind speeds  $V \in [0, 9]$  m/s, a simple two degree of freedom binary flutter model (Wingerden, 2008) is used. The parameters as given by Lee and Singh (2007) are used, but the elastic axis is moved closer to the trailing edge, such that divergence occurs before flutter takes place. The results are shown in Figure 2-1.



(c) Pole location

Figure 2-1: Pole frequency, damping, and location for  $V \in [0, 9]$  m/s in the case of divergence.

When looking at the eigenvalues, Bergami (2008) characterizes divergence by one mode with negative damping ( $\operatorname{Re}(\lambda) > 0$ ) and an imaginary part that equals zero ( $\operatorname{Im}(\lambda) = 0$ ). In other words, because divergence is a static aeroelastic phenomenon, the poles are located on the real axis when the system becomes unstable, see Figure 2-1c.

#### 2-1-2 Flutter

When discussing flutter in this thesis it is important to note that *classical flutter* is considered, as opposed to *stall flutter*, which is a different topic. Furthermore, three variables should be considered for wing flutter: flexure, torsion, and control surface rotation (Fung, 1993). When all variables are considered a *ternary* flutter mode is considered. On the other hand *binary* flutter describes the flutter mode when only two of the three variables which were stated predominate, which will be the case in this thesis.

Binary flutter "occurs when the aerodynamic forces associated with motion in two modes of vibration cause the modes to couple in an unfavourable manner" (Wright and Cooper, 2008). To illustrate this coupling the pole location, frequency, and damping is plotted in Figure 2-2 for a range of wind speeds  $V \in [0, 16]$  m/s, for a simple two degree of freedom binary flutter model (Wingerden, 2008) using the parameters as given by Lee and Singh (2007). Note that only two pole trajectories are plotted, those with a positive imaginary part, as the other two are simply complex conjugates.



**Figure 2-2:** Pole frequency, damping, and location for  $V \in [0, 16]$  m/s in the case of flutter.

Master of Science Thesis

When looking at the frequency of the two modes, see Figure 2-2a, it is observed that for increasing wind speeds the two modes approach each other, but do not actually coincide. It is however sufficient that the modes do approach each other, but do not coalesce, for the coupling to take place. This is confirmed by the fact that the damping becomes negative, see Figure 2-2b, or that one of the pole pairs crosses the imaginary axis and becomes unstable, see Figure 2-2c.

As shown by Wright and Cooper (2008) the coalescence of the two modes does occur in specific situations, for example in the case of zero aerodynamic damping, which is not the case in this research.

When looking at the eigenvalues or poles, Bergami (2008) states that flutter is characterized by at least one mode with negative damping ( $\text{Re}(\lambda) > 0$ ) and a positive imaginary part ( $\text{Im}(\lambda) > 0$ ), which corresponds to the pole locations shown in Figure 2-2c.

As shown by Figure 2-2a the coupling of two modes is not always apparent, as the modes do not have to coalesce and the way in which they approach each other can differ between situations. Therefore, it might be clearer to state that flutter is an "unstable vibration in which the structure extracts energy from the air stream and often results in catastrophic structural failure" (Wright and Cooper, 2008).

## 2-2 Modeling

The previous section on divergence and flutter described these phenomena and their effect on the system dynamics using the flutter model as presented by Wingerden (2008). This LPV flutter model will be described in detail in this section.

Because flutter is an aeroelastic phenomenon, the models show a combination of structural dynamics and aerodynamics, between which a strong coupling exists.

For the structural dynamics, at least two degrees of freedom should be present, as *binary* flutter involves the coupling of two modes. Additionally, one could add the dynamics of the trailing edge flap to the model, to obtain three degrees of freedom to describe *ternary* flutter.

When looking at modeling of the aerodynamic forces a division can be made between quasisteady and unsteady modeling. In the quasi-steady case, the aerodynamic lift and drag forces are instantaneous (stateless) functions of flow velocity and angle of attack, which implies that there are no frequency dependent terms involved. On the other hand, unsteady aerodynamics does take into account these frequency terms and describes the behavior of the airflow around the airfoil, such as the influence of wake vortices on the airfoil lift and pitching moment.

Whether unsteady effects should be considered depends on the extent to which frequency dependent behavior occurs. This can be evaluated using the reduced frequency  $k = \frac{\omega b}{V}$ , where  $\omega$  is the pitch oscillation frequency (rad/s), b the half-chord (m), and V (m/s) the free stream wind speed. For  $k \to 0$  the aerodynamic forces become *steady*. Therefore, for  $k \ll 1$  the quasi-steady assumption is valid and is used to greatly simplify the modeling.

Alternatively, the unsteady effects could be taken into account by using for example the ternary flutter model as described by Olds (1997). Several corrections have been made to this model (Mozaffari-Jovin et al., 2013, 2014), resulting in the 8-state model as used by Zhang and Behal (2014).

When looking at the structural dynamics, this model contains three degrees of freedom, as the dynamics of the trailing edge flap are no longer neglected, which adds an extra secondorder differential equation to the equations of motion. Furthermore, when looking at the aerodynamics, two so-called *lag states* are included to account for unsteady effects.

This more complex model will not be further discussed in this work and the two degree of freedom binary flutter model discussed earlier will be presented in detail in the next section.

#### 2-2-1 Model description

A schematic overview of the airfoil for which the model will be derived is shown in Figure 2-3, which shows the airfoil suspended by two springs and subject to a wind speed V (m/s).



Figure 2-3: Schematic representation of a 'smart' airfoil (Wingerden, 2008).

The structural model which describes the rotational (*pitch*) and translational (*plunge*) modes of this airfoil is given by the following equations of motion (Wingerden, 2008):

$$\begin{bmatrix} m_t & m_w x_\alpha b \\ m_w x_\alpha b & I_\alpha \end{bmatrix} \begin{bmatrix} \ddot{h} \\ \ddot{\alpha} \end{bmatrix} + \begin{bmatrix} c_h & 0 \\ 0 & c_\alpha \end{bmatrix} \begin{bmatrix} \dot{h} \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} k_h & 0 \\ 0 & k_\alpha \end{bmatrix} \begin{bmatrix} h \\ \alpha \end{bmatrix} = \begin{bmatrix} -L \\ M \end{bmatrix}$$
(2-1)

where  $m_w$  is the mass of the wing,  $m_t$  is the total mass, b is the semi-chord or half-chord,  $I_{\alpha}$  is the moment of inertia,  $x_{\alpha}$  is the non dimensionalized distance of the center of mass from the elastic axis, the spring and damping coefficients of the plunge and pitch degree of freedom are  $k_h$ ,  $k_{\alpha}$ ,  $c_h$ , and  $c_{\alpha}$ , respectively. The inputs L and M of this model are the aerodynamic force and moment, respectively. An example of a set of parameter values for this model used in literature are shown in Table 2-1 (Lee and Singh, 2007).

Note that the control surface at the trailing edge is considered to be infinitely stiff such that the corresponding dynamics can be ignored.

Name	Value	Name	Value
ρ	$1.225 \text{ kg/m}^3$	$k_h$	2844.4 N/m
a	-0.6847 -	$k_{lpha}$	$2.82 \text{ N} \cdot \text{m/rad}$
b	$0.135 {\rm m}$	$c_h$	$27.43 \mathrm{N}\cdot\mathrm{s/m}$
$s_p$	1.0 m	$c_{lpha}$	$0.036 \ \mathrm{N} \cdot \mathrm{s}$
$m_w$	2.049 kg	$c_{l\alpha}$	6.28 -
$m_t$	12.387 kg	$c_{leta}$	3.358 -
$x_{\alpha}$	$rac{0.0873}{b} - (1+a)$ -	$c_{m\alpha}$	$(rac{1}{2}+a)\cdot c_{llpha}$ -
$I_{\alpha}$	$0.0517 + m_w x_{\alpha}^2 b^2 \text{ kg} \cdot \text{m}^2$	$c_{m\beta}$	-0.635 -

Table 2-1: Flutter model parameters (Lee and Singh, 2007)

For the quasi-steady aerodynamic forces the following expressions can be derived:

$$L = \rho b s_p \left( c_{l\alpha} \begin{bmatrix} V^2 & V & b(\frac{1}{2} - a)V \end{bmatrix} \begin{bmatrix} \alpha \\ \dot{h} \\ \dot{\alpha} \end{bmatrix} + c_{l\beta} V^2 \beta \right)$$

$$M = \rho b^2 s_p \left( c_{m\alpha} \begin{bmatrix} V^2 & V & b(\frac{1}{2} - a)V \end{bmatrix} \begin{bmatrix} \alpha \\ \dot{h} \\ \dot{\alpha} \end{bmatrix} + c_{m\beta} V^2 \beta \right)$$
(2-2)

where V is the free stream velocity,  $\rho$  the air density, a the nondimensionalized distance from the midchord to the elastic axis,  $s_p$  the span of the wing,  $c_{l\alpha}$  and  $c_{l\beta}$  are the lift coefficients per angle of attack and flap angle, and  $c_{m\alpha}$  and  $c_{m\beta}$  are the moment coefficients per angle of attack and flap angle.

#### 2-2-2 LPV model

The set of second-order differential equations (2-1) combined with the derived expressions for the aerodynamic forces (2-2) can easily be rewritten as a set of four first-order differential equations (Wingerden, 2008) with wind speed dependence. In other words, an LPV statespace model of the following form is obtained:

$$\dot{x} = \left(A_1 + A_2 V + A_3 V^2\right) x + B_3 V^2 \beta$$

$$\alpha = C_1 x$$
(2-3)

M. Visser

Master of Science Thesis

where  $x = \begin{bmatrix} h & \alpha & \dot{h} & \dot{\alpha} \end{bmatrix}^{\mathrm{T}}$  and the state matrices are defined as follows:

and the constants q are defined as follows:

$$q_{2} = \rho b s_{p} c_{l\alpha} \qquad q_{4} = \rho b^{2} s_{p} c_{m\alpha} \qquad q_{6} = \left[\frac{1}{2} - a\right] b$$
$$q_{3} = \rho b s_{p} c_{l\beta} \qquad q_{5} = \rho b^{2} s_{p} c_{m\beta}$$

### 2-3 Influence of design parameters on flutter

The previous sections have presented the mathematical binary flutter model which will be used for LPV identification for flutter prediction, but the influence of the different parameters on the flutter speed has not yet been considered. This section will discuss the most important parameters and show how they influence the occurrence of flutter.

Critical parameters for flutter are discussed in textbooks on aeroelasticity such as Fung (1993); Wright and Cooper (2008). Furthermore, when looking at the application of wind turbine rotor blades, investigation into these parameters has also been studied by Lobitz (2005); Hansen (2007).

This research listed the following fundamental parameters that increase the risk of flutter (Fung, 1993):

Wind speed The wind speed is the most obvious parameter of interest, as by keeping the wind speed low, the occurrence of flutter can be prevented.

Master of Science Thesis

- **Stiffness** The effect of the individual changes of the torsional and translational stiffness cannot be stated in such general terms but their influence on the flutter speed is very important. In general, an increase in the torsional stiffness will result in an increased flutter speed.
- **Inertia and aerodynamic couplings** The arrangement of the elastic axis and the inertia axis compared to the aerodynamic center of the wing can significantly influence flutter. For example, moving the axes close to each other reduces flutter.

As the (maximum) wind speed is frequently determined by the application, the most important parameters that can be used to influence the flutter speed are the structural stiffness and center of mass along with the location of the elastic axis.

Now that this chapter presented the required background knowledge on flutter modeling, the next chapter will continue with the application of LPV identification for flutter prediction.

M. Visser

# Chapter 3

# LPV identification methods

The previous chapter introduced the phenomenon of flutter and showed how the dynamics of such an aeroelastic system depend upon the wind speed, which naturally resulted in the formulation of an LPV model. Furthermore, as discussed in the introduction, see Chapter 1, LPV identification will be used to predict flutter. Therefore, this chapter concerns the topic of batch LPV identification.

First, a division between global, local, and glocal methods is made (Lovera et al., 2013), after which four algorithms from literature are presented. Next, the chosen simulation scenario is presented and motivated. Furthermore, the LPV identification methods are tested on simulation data sets and the results are presented. This chapter is concluded with a short synopsis.

## 3-1 Introduction

LPV identification as treated in this thesis concerns the identification of a parameterized LPV model  $H(\theta)$  from a measured input signal u, output signal y, and scheduling sequence  $\theta$ , see Eq. (3-1).

$$\begin{bmatrix} u & \theta \end{bmatrix} \rightarrow \underbrace{\begin{bmatrix} A(\theta) & B(\theta) \\ C(\theta) & D(\theta) \end{bmatrix}}_{H(\theta)} \rightarrow y$$
(3-1)

Methods to obtain  $H(\theta)$  are further divided into three main groups by Lovera et al. (2013), that of the *global*, *local*, and *glocal* methods, which will also be used in this chapter to group the investigated algorithms from literature. Before looking at these algorithms in detail, the *global*, *local*, and *glocal* concepts will be introduced and the differences are highlighted.

#### 3-1-1 Global

The global methods obtain  $H(\theta)$  from u, y, and  $\theta$ , where  $\theta$  can be an arbitrary signal varying within the operating range of the system. In other words, the scheduling sequence is varied

during the experiment along with the input signal, after which the LPV model is directly obtained from u, y, and  $\theta$ .

The concept is thus simple, as only one experiment needs to be performed to directly obtain the LPV model. Unfortunately, global methods suffer from the so-called 'curse of dimensionality' (Van Wingerden and Verhaegen, 2009), meaning that for an increasing past window p the data matrices involved in the calculations grow exponentially, resulting in numerical problems. The past window p is thus bounded from above by these numerical problems, while on the other hand, it should be increased to reduce the effect of noise and to capture significant variance in the parameter  $\theta$ . Clearly, the conflicting demands on the past window make for a trade-off between computational complexity and estimate bias.

#### 3-1-2 Local

Alternatively, a local method can be used for which m 'local' experiments for a fixed scheduling sequence  $\theta_l$  are performed to obtain a set of m LTI models  $H_l$ , see Eq. (3-2).

$$\begin{bmatrix} u & \theta_l \end{bmatrix} \to \underbrace{\begin{bmatrix} A_l & B_l \\ C_l & D_l \end{bmatrix}}_{H_l} \to y, \quad \text{for } l = 1, ..., m$$
(3-2)

Note that the *m* identification problems for fixed scheduling sequences  $\theta_l$ , as shown in Eq. (3-2), should be solved using LTI identification methods, as the parameter varying part is kept constant. For the LTI identification off-the-shelf algorithms can be used, which are for example discussed in a recent overview of closed-loop LTI subspace identification methods by Van der Veen et al. (2013).

After the *m* local models  $H_l$  are identified, these models are interpolated to obtain an LPV model. Important to note here is that this interpolation is only valid for a slowly varying scheduling sequence, as only fixed values of  $\theta$  have been used in the identification and the rate of change of the scheduling sequence  $\theta$  is neglected. However, this is not an unusual assumption, as it is a common requirement for conventional gain scheduling techniques (Shamma and Athans, 1992). Furthermore, the scheduling sequence varies at a slow rate for most applications, which includes the wind speed in the application of flutter identification.

When looking at this interpolation, it takes place between the system matrices of the parameterized LPV model  $(A(\theta_l), B(\theta_l), C(\theta_l), \text{ and } D(\theta_l))$  and the local model state matrices  $(A_l, B_l, C_l, \text{ and } D_l)$  for l = 1, ..., m. Unfortunately, LTI identification returns a non-unique state-space model, as only the input/output behavior can be extracted from the data but the state is not fixed. In other words, when the state is transformed such that  $z = T_l x$ , where  $T_l$  is an arbitrary non-singular matrix, the input/output behavior does not change. However, a state transformation does change the state matrices, see Eq. (3-3).

$$\dot{z} = T_l A T_l^{-1} z + T_l B u$$

$$y = C T_l^{-1} z + D u$$
(3-3)

Therefore, when interpolating the state matrices it is important that all m local LTI models are represented in the same state basis. Which directly points out the main drawback of the local methods, that is the requirement of a coherent state basis.

15

Once the local models are all transformed into a coherent state basis, the interpolation can take place by fitting the state matrices of the parameterized LPV model  $H(\theta_l)$  to the local state matrices  $H_l$ . This optimization problem can be written as a linear least squares problem, which can be solved efficiently (Van den Boom and De Schutter, 2012).

#### 3-1-3 Glocal

Finally, there are the glocal methods, which start with the same set of m local LTI models, see Eq. (3-2). But unlike the local methods, the combination of these local models into an LPV model is not obtained by interpolating between system matrices, but by fitting the parameterized LPV model to the local models in the frequency domain, thus looking only at the input/output behavior.

By directly looking at the input/output behavior the requirement of a coherent state basis is circumvented. Unfortunately, this comes at the cost of a more difficult optimization problem, for example non-convex and/or non-smooth optimization problem, which takes significantly more computational time and does not guarantee an optimal solution.

Furthermore, because the optimization problem can get stuck in local minima, the selection of the initial point significantly influences the final solution. In other words, an initial LPV model should be provided, which adds another challenge to the glocal methods (Petersson and Löfberg, 2014).

Now that the general concepts of *global*, *local*, and *glocal* LPV identification have been discussed, the next section will present four algorithms from literature grouped by their general concept.

### 3-2 Methods

#### **3-2-1** Global: PBSID<sub>opt</sub>

The PBSID<sub>opt</sub> algorithm for LPV systems by Van Wingerden and Verhaegen (2009) is a global identification method based on the PBSID<sub>opt</sub> LTI identification algorithm by Chiuso (2007) which allows both identification in open-loop and closed-loop operation. Furthermore, it works in discrete-time and uses an affine parameterization of the state matrices in the N scheduling variables  $\mu^{(i)}$ , as illustrated for the state matrix A in Eq. (3-4).

$$A_k = \sum_{i=1}^{N} \mu_k^{(i)} A^{(i)}$$
(3-4)

These scheduling variables  $\mu^{(i)}$  are also referred to as *basis functions*  $f^{(i)}(\theta)$  (De Caigny et al., 2014), see Eq. (3-5).

$$H(\theta) = \begin{bmatrix} A(\theta) & B(\theta) \\ C(\theta) & D(\theta) \end{bmatrix} = \sum_{i=1}^{N} f^{(i)}(\theta) \begin{bmatrix} \hat{A}^{(i)} & \hat{B}^{(i)} \\ \hat{C}^{(i)} & \hat{D}^{(i)} \end{bmatrix}$$
(3-5)

Where  $f^{(i)}(\theta)$  equals  $\theta^{i-1}$  for i = 1, ..., N in the case of an affine dependence of the state matrices on the wind speed.

Master of Science Thesis

M. Visser

The main drawback of this algorithm, as also mentioned for global methods in general, is the 'curse of dimensionality' for which Van Wingerden and Verhaegen (2009) introduce the kernel method to partly solve this problem. The kernel method uses so-called kernel matrices that have much smaller dimensions than the data matrices used in the original optimization problem (Verdult and Verhaegen, 2005). However, while the computational complexity is reduced, the kernel method does introduce a bias in the estimate.

#### 3-2-2 Local: SMILE

De Caigny et al. (2014) introduce the State-space Model Interpolation of Local Estimates (SMILE) algorithm which works in both continuous- and discrete-time and supports Multiple Input, Multiple Output (MIMO) systems. As described in the introduction on *local* methods, the main challenge lies in finding the right similarity transformations  $T_i$  such that all local models are represented in a coherent state basis. Unlike many other algorithms, the SMILE algorithm is able to find these similarity transformations without requiring user input, like manually sorting the poles of the local systems (De Caigny et al., 2011). Therefore, it is interesting to take a closer look at the way in which the SMILE method is able to represent all local models in a fixed state basis.

The approach is based on the controllability and observability properties of the local systems. First, for every local model i = 1, ..., m the maximum condition number  $\kappa_{O,i} = \max_l \kappa(O_l^{\dagger}O_l)$  is calculated for l = 1, ..., m, where  $O_i$  is the observability matrix of local model  $i, \kappa(A)$  denotes the condition number of the matrix A, and  $A^{\dagger}$  denotes the Moore-Penrose inverse of the matrix A. Next, the local model  $i_{\text{ref}} = \arg\min_i \kappa_{O,i}$  is selected as a reference model and the other local models are transformed into a coherent state basis using the similarity transformation matrix  $T_l = O_{i_{\text{ref}}}^{\dagger}O_l$ . The proof that this similarity transformation based on the observability matrices results in a coherent state basis is given by De Caigny et al. (2014) along with a similar derivation for the use of the controllability matrix.

Now that the local models are transformed into a coherent state basis, the interpolation between the state matrices is performed by solving an optimization problem which minimizes the following cost function (3-6):

$$E = \sum_{l=1}^{m} \left\| \sum_{i=1}^{N} f^{(i)}(\theta_l) \underbrace{\begin{bmatrix} \hat{A}^{(i)} & \hat{B}^{(i)} \\ \hat{C}^{(i)} & \hat{D}^{(i)} \end{bmatrix}}_{\hat{H}_i} - \underbrace{\begin{bmatrix} A^{(l)} & B^{(l)} \\ C^{(l)} & D^{(l)} \end{bmatrix}}_{H_l} \right\|_F^2$$
(3-6)

where N is the number of basis functions  $f_i(\theta_l)$  as defined in the parameterization as given in Eq. (3-5), and  $\theta_l$  the local wind speed for model l. Furthermore,  $\hat{H}_i$  are the parameters of the optimization problem.

The described optimization problem can be rewritten as a Linear Least Squares (LLS) problem which can be solved efficiently and does not show the numerical problems that are present in the *global* and *glocal* methods.

#### **3-2-3** Glocal: $\mathcal{H}_2$ norm

Petersson (2013) presents the  $H_2NL$  glocal LPV identification algorithm which aims to mini-

mize the sum of  $\mathcal{H}_2$  norms of the local error systems  $E_l = H(\theta_l) - H_l$ , see Eq. (3-7), where  $H(\theta_l)$  is the parameterized LPV model evaluated at the local wind speed  $\theta_l$  and  $H_l$  is the local LTI model. Note that  $E_l$  is evaluated using the  $\mathcal{H}_2$  system norm in the frequency domain, whereas Eq. (3-6) uses the Frobenius matrix norm.

$$\min_{\hat{A}_{i},\hat{B}_{i},\hat{C}_{i},\hat{D}_{i}} \sum_{l=1}^{m} \left\| H(\theta_{l}) - H_{l} \right\|_{2}^{2}$$
(3-7)

In Eq. (3-7) the optimization parameters are the matrices  $\hat{A}_i$ ,  $\hat{B}_i$ ,  $\hat{C}_i$ ,  $\hat{D}_i$  for i = 1, ..., N as defined in Eq. (3-5). In short, the optimization can be seen as fitting the bode plots of the LPV model evaluated at local points  $H(\theta_l)$  to the local LTI models  $H_l$  using the  $\mathcal{H}_2$  norm.

For this non-convex and non-linear optimization problem, see Eq. (3-7), the gradient is derived by Petersson (2013) to speed up the optimization and allow the use of standard unconstrained quasi-Newton solvers. In this comparison the **fminunc** solver available in MATLAB will be used. Furthermore, as noted in the introduction on *glocal* methods, the selection of the initial point for the optimization problem is very important, which is also pointed out by Petersson (2013). Initial simulations have shown that when no suitable initial LPV model is supplied to the H<sub>2</sub>NL algorithm, the optimization gets stuck in local minima, and the LPV model is not successfully fitted to the local LTI models. Therefore, in this chapter the H<sub>2</sub>NL algorithm will be initialized with the LPV model estimate as obtained by the local SMILE algorithm, see Section 3-2-2.

Finally, the optimization problem as defined in (3-7) can also be extended to include regularization to reduce the influence of errors in the data (Petersson, 2013, Ch. 4.4.2).

#### **3-2-4** Glocal: $\mathcal{H}_{\infty}$ norm

Next, a different glocal method is considered (Vizer et al., 2013), which unlike the  $\mathcal{H}_2$  norm used by Petersson (2013), uses the  $\mathcal{H}_{\infty}$  norm. The use of the  $\mathcal{H}_2$  or  $\mathcal{H}_{\infty}$  system norms results in different behavior in the optimization. This difference is explained by Skogestad and Postlethwaite (2007), which states that the  $\mathcal{H}_2$  pushes down the magnitude for all frequencies, that is all singular values over all frequencies, while the  $\mathcal{H}_{\infty}$  norm pushes down the peaks in the magnitude, that is the largest singular values. This difference is also shown in the cost function, see Eq. (3-8), where compared to the  $\mathcal{H}_2$  cost function (3-7) the sum operator is replaced by the max operator.

$$\min_{\hat{A}_i, \hat{B}_i, \hat{C}_i, \hat{D}_i} \max_{l \in \{1, \dots, m\}} \left\| H_l - H(\theta_l) \right\|_{\infty}$$
(3-8)

Note that, unlike for the other algorithms, the LPV model  $H(\theta)$  in Eq. (3-8) does not use the affine parameterization as given in Eq. (3-5), but a Linear Fractional Representation (LFR) is used to parameterize the LPV model. Vizer et al. (2013) mentions that the LFR representation is less restrictive than the affine parameterization as given in Eq. (3-5). This is also illustrated by Zhou et al. (1996) with several examples, among which an equivalent LFR is derived for the parameterization as given in Eq. (3-5), which uses the basis functions. The  $\mathcal{H}_{\infty}$  norm cost function (3-8) results in a non-smooth optimization problem. To solve this problem the approach by Apkarian and Noll (2006) is used, which is implemented in the hinfstruct function in MATLAB.

Unfortunately this method has currently only been derived for continuous-time systems, which is a drawback, because the LTI identification used to obtain the local models frequently results in discrete-time models. While it is possible to transform these discrete-time models into continuous-time by for example using a zero-order-hold or Tustin approximation (Åström and Wittenmark, 2011), this will always introduce an additional source of error.

### 3-3 Simulation scenario

Before continuing to the results it is important to present the simulation scenario to which the different methods are subjected. As mentioned in Chapter 2 on first principles flutter modeling the LPV model as presented in Section 2-2-2 will be used to generate data for the identification methods to use. This section will discuss the data generation process in more detail. Furthermore, it will present an LTI identification algorithm to provide the *local* and *glocal* methods with local state-space models.

First, for the local and glocal methods the operating points are selected as  $\bar{\theta}_l \in \{4, 6, 8, 10\}$ m/s resulting in a total of m = 4 operating points. Furthermore, the identification data is chosen to contain  $N_p = 1250$  data points, where for the local experiments this is divided by m, such that the *local* and *glocal* methods use the same amount of data as the *global* method.

Recall from Eq. (2-3) that the flutter model uses the flap angle  $\beta$  as an input for which a zero-mean uniformly distributed pseudo random signal  $\beta \in [-30^{\circ}, 30^{\circ}]$  is selected.

Next, the scheduling sequence for the global identification  $\theta(k)$  is chosen as follows (3-9):

$$\theta(k) = \bar{\theta} + A_{\theta} \sin(\frac{2\pi k}{j}) + w_{\theta}(k)$$
(3-9)

where  $\bar{\theta} = 7$  m/s is the mean wind speed,  $A_{\theta} = 1.75$  m/s the sine amplitude, j = 12.5 s the sine period, and  $w_{\theta}(k)$  a normally distributed zero-mean white noise signal with variance 0.42.

Note that when simulating the true LPV model (2-3) for the identification of local LTI models, the wind speed is also perturbed by the same noise sequence  $w_{\theta}(k)$ , see Eq. (3-10).

$$\theta_l(k) = \bar{\theta}_l + w_\theta(k) \tag{3-10}$$

Finally, a measurement noise sequence  $v_{\alpha}(k)$  is added to the output pitch angle  $\alpha$  to obtain a specified Signal to Noise Ratio (SNR).

#### 3-3-1 LTI identification

Now that the signals are defined that will be used to generate the data for the identification algorithms, it remains to shortly discuss the LTI identification algorithm which will be used to provide the *local* and *glocal* methods with a set of local state-space models.

As mentioned in the introduction to local LPV identification, off-the-shelf LTI identification algorithms can be used for this application. The overview by Van der Veen et al. (2013) suggests the use of the  $PBSID_{opt}$  algorithm (Chiuso, 2007), which can be used under both open-loop and closed-loop conditions.

Alternatively, for open-loop systems the PI-MOESP algorithm (Verhaegen, 1994) could be used, but in practice the PBSID<sub>opt</sub> algorithm was found to be more robust for this application.

### 3-4 Results

This section will present the results of the comparison between the four LPV identification algorithms for the application of flutter speed prediction, given the simulation scenario as presented in the previous section. First, the glocal  $\mathcal{H}_{\infty}$ -based method will be treated separately as it requires significantly more computational time, but more importantly the continuous-time requirement makes for an unfair comparison with the discrete-time method, as errors are introduced in the discrete-to-continuous conversion. Secondly, the flutter prediction capabilities of the three remaining methods are evaluated. Furthermore, the possibility of model reduction is investigated and the section is concluded with some remarks on the computational aspects of the different algorithms.

#### **3-4-1** Glocal $\mathcal{H}_{\infty}$ -based

In this section, the glocal  $\mathcal{H}_{\infty}$ -based method by Vizer et al. (2013) will be discussed separately from the other methods and this separate treatment will be motivated. As mentioned in Section 3-2-4 the optimization method used by the  $\mathcal{H}_{\infty}$ -based glocal method is adapted for continuous-time systems, while the LTI identification returns discrete-time local LTI models. Therefore, a d2c (discrete-to-continuous) conversion is required, which introduces an error between the equivalent discrete-time and continuous-time LTI models, which will disappear as the sampling time  $T_s$  tends to zero.

To investigate the estimated model fit, the Variance Accounted For (VAF) is used as a measure of similarity between the simulation output of the true LPV model and the estimated LPV model for a specified validation data set, see Section 3-3. Furthermore, the flutter speed  $V_f$  prediction error is defined as  $\frac{\hat{V}_f - V_f}{V_f} \cdot 100\%$ , where  $\hat{V}_f$  is the flutter speed of the estimated LPV model, and  $V_f$  the flutter speed of the true LPV model.

To investigate the effect of the sampling time on the estimated LPV model the results are shown in Table 3-1 for  $T_s \in \{0.04, 0.08\}$  s and SNR= 40. Note that for this comparison the number of samples  $N_p$  was kept constant, such that the amount of data available for the LTI model identification is not changed.

**Table 3-1:**  $\mathcal{H}_{\infty}$ -based glocal method (Vizer et al., 2013) performance for two sampling times  $T_s$  (SNR = 40)

$\mathbf{T}_{\mathrm{s}}$	VAF	$\mathbf{V}_{\mathrm{f}}$ error	Time
$0.08 \mathrm{\ s}$	90.7~%	14.4~%	$1:55 { m h}$
0.04 s	95.1 %	-9.7 %	2:14 h

It follows from Table 3-1 that the performance improves for a decreasing sampling time  $T_s$  as shown by the increased VAF and lower flutter speed  $V_f$  prediction error. Furthermore, to show that the  $\mathcal{H}_{\infty}$  norm minimization is successfully performed by the algorithm, the fit of the estimated LPV model to the local continuous-time LTI models is evaluated for a sampling time  $T_s = 0.08$  s, see Figure 3-1.



Figure 3-1:  $\mathcal{H}_{\infty}$ -based glocal LPV estimate fits the *local* continuous-time LTI models very well.

Figure 3-1 shows that the estimated LPV model fits the local continuous-time LTI models very well, which indicates that the cost function of the optimization as given in Eq. (3-8) was successfully minimized by the algorithm as presented by Vizer et al. (2013).

Therefore, when Figure 3-1 shows no error between the continuous-time LTI and LPV model, the error must have been introduced by the d2c conversion, which makes a comparison with the other discrete-time LPV identification methods unfair.

Finally, as shown in Table 3-1, the algorithm requires significantly more computational time compared to the other methods, which will be discussed in Section 3-4-4.

#### 3-4-2 Flutter prediction

Now that the previous section treated the  $\mathcal{H}_{\infty}$ -based glocal method separately, this section will present the flutter prediction capabilities of the remaining three algorithms.

When looking at the phenomenon of flutter, the system becomes unstable at the flutter speed  $V_f$ , which for discrete-time systems means that one or more poles move outside of the unit disk. Furthermore, the poles of the system can be found by calculating the eigenvalues of the state matrix. In the case of an estimated LPV model, see Eq. (3-5), for a wind speed  $\theta_k$  the stability criterion can be formulated as  $|\lambda_i(A(\theta_k))| < 1$  for  $i = 1, ..., \hat{n}_x$ , where  $\lambda_i(A)$  is the i'th eigenvalue of the matrix A, and  $\hat{n}_x$  is the system order or number of states.

In order to calculate the flutter speed of the estimated LPV models, a range of wind speeds  $\theta_k \in \{0, 1, ..., 20\}$  is selected for which the maximum absolute value of the eigenvalues of the state matrix  $p_{\max}(\theta_k) = \max_i |\lambda_i(A(\theta_k))|$  is calculated. A plot of these maximum pole amplitudes as a function of the wind speed is given in Figure 3-2.



curacy, while at higher noise values glocal methods show the best bias/variance trade-off.

Master of Science Thesis

	SMILE		$\begin{tabular}{ c c c c } SMILE & H_2NL \end{tabular}$		$\mathbf{PBSID}_{\mathrm{opt}}$	
SNR	40	5	40	5	40	5
$\hat{n}_x = 4$	-7.7 %	-16.8 %	-2.1 %	-13.2 %	0.2~%	-6.6 %
$\hat{n}_x = 2$	19.1 %	103.4 $\%$	13.1~%	38.5~%	-17.8 %	47.8~%

**Table 3-2:** Flutter speed prediction error for two state dimensions  $\hat{n}_x \in \{2, 4\}$  (%)

As shown in Figure 3-2, the wind speed  $\hat{V}_f$  at which the line  $p_{\max}(\theta_k) = 1$  is crossed can be calculated by interpolation for each method and the results are summarized in Table 3-2. Furthermore, the VAF values for  $\hat{n}_x = 4$ , as shown in Figure 3-3, are included in Table 3-3.

In Figure 3-2 and 3-3 the black dashed vertical lines indicate the region in which local LTI models have been identified at the operating points  $\bar{\theta}_l \in \{4, 6, 8, 10\}$  m/s as given in Section 3-3, see Eq. (3-10). Furthermore, the red dashed vertical line indicates the flutter speed  $V_f = 12.41$  m/s of the true LPV model. Finally, the grey markers indicate the results for different noise realizations, while the colored markers show the results for median VAF.

When looking at the results as given in the Figures 3-2 and 3-3 and Tables 3-2 and 3-3 multiple observations can be made.

Firstly, when only looking at the model fit in terms of the VAF, see Figure 3-3 and Table 3-3, a trade-off is taking place between the bias and variance of the estimates for increasing noise, see Figure 3-3a and 3-3b. For a SNR = 40, see Figure 3-3a, the PBSID<sub>opt</sub> global method shows the highest VAF and the local SMILE and glocal H<sub>2</sub>NL algorithms show no noticeable difference in VAF. On the other hand, when the noise is increased (SNR = 5), see Figure 3-3b, the local SMILE method outperforms the global PBSID<sub>opt</sub> method in terms of the bias, shown by the higher median VAF, while the variance of the local SMILE method is larger than for the PBSID<sub>opt</sub> method. Furthermore, the glocal H<sub>2</sub>NL method is able to improve both the bias and variance of the local SMILE estimate, as shown in Figure 3-3b.

Secondly, for  $\hat{n}_x = 4$  Table 3-2 shows that in the case of low noise (SNR = 40) all methods result in good flutter speed predictions. On the other hand, when the noise is increased (SNR = 5) the pole prediction for low and high wind speeds significantly deteriorates, while the point of instability, that is the flutter speed, is still predicted within acceptable bounds. Overall, the global PBSID<sub>opt</sub> shows the best flutter speed predictions both at low and high noise levels.

#### 3-4-3 Model reduction

In the previous sections the state dimension  $\hat{n}_x = 4$  of the parameterized LPV model (3-5) was chosen equal to the state dimension  $n_x$  of the true LPV model (2-3). When the data originates from experiments on physical systems the true state dimension  $n_x$  is not given a priori and  $\hat{n}_x$  should be selected based on information in the data or prior knowledge of the system. Furthermore, when a high order system is identified it might be beneficial to select  $\hat{n}_x < n_x$  to obtain a simple model of the system, which allows for faster model-based control.

This brings us to the point of *model reduction*, where the LPV model order is selected smaller than the system order  $n_x$ . For the local SMILE method this is done by selecting a lower state dimension in the LTI state-space identification algorithm, such that the local
state dimension equals  $\hat{n}_x$ , in which case the interpolation step result in a same dimension LPV estimate. On the other hand, in the case of the glocal methods the local LTI model dimension does not need to be equal to the parameterized LPV model state dimension as only the input/output behavior is considered and the state matrices are not directly interpolated.

To investigate the model fit and flutter prediction when model reduction is applied, a state dimension  $\hat{n}_x = 2$  is selected, for which the maximum pole amplitude is given in Figure 3-4 and the model fit and flutter prediction error is shown in Table 3-3 and 3-2, respectively.



**Figure 3-4:** Flutter prediction deteriorates significantly when a degree of freedom is removed  $(\hat{n}_x = 2)$ .

**Table 3-3:** VAF of estimated LPV models for two state dimensions  $\hat{n}_x \in \{2, 4\}$  (%)

	SMILE		$\mathbf{H}_{2}\mathbf{NL}$		$\mathbf{PBSID}_{\mathrm{opt}}$	
SNR	40	5	40	5	40	5
$\hat{n}_x = 4$	97.3~%	91.5~%	97.2 %	93.7~%	99.8~%	87.3 %
$\hat{n}_x = 2$	96.4 %	96.4~%	96.4 %	96.2~%	89.5~%	89.0 %

Table 3-3 shows a relatively good VAF for the reduced order estimated LPV model. On the other hand, Figure 3-4 shows very poor prediction of the maximum pole amplitude and flutter speed, as also shown in Table 3-2. In other words, while the behavior of the system is still relatively well explained by the reduced order model as shown in the good VAF values, the pole locations no longer represent the phenomenon of flutter. Note that this is no surprise as flutter requires two degrees of freedom, see the explanation in Section 2-1-2, while the reduced order model only has a single degree of freedom as  $\hat{n}_x = 2$ .

### 3-4-4 Computational aspects

The previous sections have discussed various aspects of the four LPV identification algorithms, including the model fit of the estimates, the pole prediction capabilities and the possibility of

model reduction. Finally, this section will consider the computational aspects of the different algorithms, as this significantly influences the practicality of an algorithm. It should however be noted that the computational aspects are influenced by the implementation in code and the used hardware, which makes it hard to give definitive conclusions. However, it is the aim to provide the reader with an initial indication as to whether the presented algorithms could be used for a specific application.

Regarding the presented simulation times, all calculations were performed using an Intel i7 860 processor and 4 GB of RAM running MATLAB 2014a under Windows 8.1 Pro. Furthermore, for the glocal methods it should be noted that the initial point of the optimization significantly influences the computational time.

As this chapter only concerned the LPV model identification from data generated according to a single simulation scenario, see Section 3-3, it is interesting to first consult literature to see if expressions exist which relate the computational complexity to important parameters, such as the model order or number of local models.

In the case of the H<sub>2</sub>NL method by Petersson (2013) the computational complexity per iteration is given as  $\mathcal{O}\left(m\left[n_x^2\hat{n}_x + n_x\hat{n}_x^2\right]\right)$ , where *m* is the number of local LTI models,  $n_x$  is the state dimension of the local LTI models, and  $\hat{n}_x$  is the state dimension of the parameterized LPV model (3-5).

For the local SMILE method no derivation on the computational complexity is given by the authors (De Caigny et al., 2014), but for the simulations performed in this chapter no noticeable difference was observed in the computational time when the number of basis functions N of the parameterized LPV model (3-6) or the model order  $\hat{n}_x$  were increased. On average, a computational time for the SMILE method of  $t_{\text{SMILE}} \approx 0.1$  s was observed.

In the case of the global PBSID<sub>opt</sub> method the most important parameters are the past window p and the number of basis functions N. For p = 5 and N = 3, as used in this chapter, an average computational time of  $t_{\text{PBSID}} \approx 0.6$  s was observed.

Finally, when looking at the observed computational time of the H<sub>2</sub>NL method for the simulation scenario as described in Section 3-3 (N = 3,  $\hat{n}_x = 4$ , and m = 4), a much larger variance is present on the set of observed computational times, resulting in  $t_{\text{H}_2\text{NL}} \in [4, 20]$  s.

## 3-5 Synopsis

From the presented results in Section 3-4 multiple short conclusions are drawn in this synopsis along with some possible future research directions.

First, the algorithms have shown the ability to predict the flutter speed to within around 10 % even in the presence of significant noise, see Table 3-2, which shows the potential of LPV identification for flutter prediction. For increasing noise the pole prediction at low and high wind speeds quickly deteriorates, while the point of instability is still predicted within reasonable bounds. When looking at the difference between the four algorithms in terms of flutter prediction, the global PBSID<sub>opt</sub> algorithm overall shows the smallest error.

	SMILE	$\mathbf{H}_{2}\mathbf{NL}$	$Glocal  \mathcal{H}_\infty$	$\mathbf{PBSID}_{\mathrm{opt}}$
	De Caigny et al. (2014)	Petersson (2013)	Vizer et al. (2013)	Van Wingerden and Verhaegen (2009)
Optimization	Linear Least Squares (convex)	quasi-Newton (non-convex)	Descent directions + line search (non-smooth)	Linear Least Squares (convex)
Parameterization	Basis functions $f_i(\theta)$	Basis functions $f_i(\theta)$	m LFR	Basis functions $f_i(\theta)$
Cost-function form	$\sum_{l} \ E_l\ _F^2$	$\sum_{l} \ E_l(j\omega)\ _2^2$	$\max_{l} \ E_{l}(j\omega)\ _{\infty}$	$\ E\ _{F}^{2}$
Requirements	Slowly varying $\theta$	Slowly varying $\theta$ , local stability	Slowly varying $\theta$ , local stability, continuous time	-
Bias / variance	low / high	low / medium	-	medium / low
Initialization	-	Crucial	Less crucial	-
Computational Case of Sec. 3-4-4	$t \approx 0.1 \text{ s}$	$t \approx 10 \text{ s}$	$t \approx 2$ h	$t \approx 0.6 \text{ s}$

Table 3-4: Comparison of LPV identification algorithms

Furthermore, a trade-off between the bias and variance was observed for an increasing noise level, where for significant noise (SNR = 5) the *local* method outperforms the *global* method. Moreover, the *glocal*  $\mathcal{H}_2$ -based method is able to further improve both the bias and variance of the *local* estimate. This both shows the capability of the *glocal* method to outperform the *local* method, but also shows the requirement of a good initial LPV model for the non-convex optimization problem as also pointed out by Petersson (2013).

The different aspects of the four LPV identification algorithms presented in this chapter are also summarized in Table 3-4.

From the presented results in this chapter two interesting future research directions have emerged.

Firstly, the  $\mathcal{H}_{\infty}$ -based glocal algorithm was able to fit the parameterized LPV model to the local continuous-time LTI models very well, see Figure 3-1, but the restriction of continuous-time LTI models makes it less suitable to be combined with common discrete-time LTI identification methods.

Secondly, as the *global* and *local* methods both allow for closed-loop identification of unstable systems using a stabilizing controller, it would be interesting to investigate how the *glocal* methods can include the stabilizing controller in the algorithm and compare the different methods for a closed-loop LPV identification application.

# Chapter 4

# **Experimental setup validation**

In the previous chapter simulation data was used to validate the concept of LPV model identification for flutter prediction. The next step is to validate the LPV identification methods on experimental data, for which an experimental setup has been provided to perform flutter experiments. However, because it is the first time that this setup is used, this chapter will answer the question whether the behavior of the setup corresponds to aeroelastic flutter.

To answer this question, the setup is first divided into two individual parts, namely a morphing airfoil and an elastic suspension, like we also saw in the LPV model from Section 2-2-1. These two parts will be presented separately and the observed shortcomings will be discussed. Next, the combined system is considered inside the wind tunnel and the frequency spectrum is estimated to check if the system dynamics resemble those of aeroelastic flutter.

### 4-1 Morphing airfoil

A morphing airfoil allows for a smooth change in camber of the trailing edge, as opposed to the frequently used hinged flaps, which results in a higher aerodynamic efficiency (Bilgen et al., 2013). The front of the airfoil is shaped like a NACA 0015 airfoil out of which a thin stainless steel plate extends to which piezoelectric benders are bonded on both sides, resulting in a triple layer bender. When the piezoelectric elements on top are supplied with a voltage Uand the bottom elements are supplied with the opposite voltage -U the stainless steel plate will start to bend as shown in Figure 4-1.



Figure 4-1: Morphing airfoil with a deformable trailing edge.

The deformation of the trailing edge tip from rest is denoted by  $\delta$  for which Wang and Cross (1999) derived the relation with the actuation voltage U as given in Eq. (4-1):

$$\delta = \frac{3s_{11}^m d_{33}(t_m + t_p)L^2}{2s_{11}^m (3t_m^2 t_p + 6t_m t_p^2 + 4t_p^3) + s_{11}^p t_m^3} U$$
(4-1)

Master of Science Thesis

M. Visser

where  $s_{11}^m$  is the compliance of the elastic material,  $s_{11}^p$  the compliance of the piezoelectric material,  $d_{33}$  the piezoelectric coefficient,  $t_m$  the elastic material thickness,  $t_p$  the piezoelectric material thickness, L the active length of the piezoelectric element, and U the applied voltage over the piezoelectric elements.

The piezoelectric elements are M8557-P1 Macro Fiber Composites (MFCs) (Smart Material Corp., 2015) and the stainless steel plate was measured to be 0.3 mm thick. Furthermore, a TREK Model PZD700A piezo driver (TREK, Inc., 2015) is used which supplies a maximum voltage of 700 V.

If we now calculate the maximum tip deformation for this configuration using Eq. (4-1),  $\delta = 4.8$  mm is obtained. However, we should take into account that only 63 % of the stainless steel plate is covered by the MFCs. When measuring the displacements using a Micro-Epsilon optoNCDT 1302 laser sensor (Micro-Epsilon, 2015) a value of  $\delta = 2.7$  mm was observed, which is slightly smaller than calculated.

The fabrication of the morphing airfoil, see Figure 4-2, consisted of several main steps.



(a) Individual parts



(b) Assembly of the steel and piezoelectric parts



(c) Attachment of the foam



(d) Inside the wind tunnel

Figure 4-2: Photographs of the morphing airfoil used in the experiments.

First, the steel components were laser cut to size and the NACA 0015 shaped front was CNC cut out of polystyrene foam, see Figure 4-2a. The steel parts were then glued together

28

using a metal epoxy glue and the piezoelectric elements were bonded to the trailing edge using Loctite E-120HP epoxy adhesive, see Figure 4-2b, which is recommended by the MFC manufacturer (Smart Material Corp., 2015). Next, the polystyrene foam is attached to the steel body using double sided adhesive tape, see Figure 4-2c, and finally the foam part is coated in fiberglass to provide structural rigidity to the leading edge of the morphing airfoil, see Figure 4-2d.

## 4-1-1 Wind tunnel

The morphing airfoil will be suspended inside a wind tunnel to generate the airflow required for the experiments and to control the wind speed, see Figure 4-3. A Smart Blade Visualization Wind Tunnel (Smart Blade, 2015) is used, which allows the wind speed to be controlled between 0 and 10 m/s. Furthermore, while the wind tunnel is equipped with flow visualization this will not be used in this thesis.



(a) Overview

(b) Test section

Figure 4-3: Small visualization wind tunnel inside the lab.

# 4-2 Electronically controlled suspension

The second part of the experimental setup is the elastic suspension, which uses two torque controlled Robotis Dynamixel MX-106R (Robotis, 2015) servo motors to control the stiffness electronically. This section will first present the concept and control scheme, after which some of the problems encountered are discussed.

### 4-2-1 Concept

The concept which is used is illustrated in Figure 4-4, where the two torque controlled servo motors are fixed to the blue linkages and the morphing airfoil is fixed to the center of the red linkage. To prevent horizontal motion of the airfoil, the center of the red linkage has been guided using vertical profile guide rails.

The position is measured using encoders inside the MX-106R servo motors, which provide the motor shaft angles  $\phi_1$  and  $\phi_2$  which can be related to the rotation angle  $\alpha$  and the vertical displacement h of the center of the red linkage, see Figure 4-4a. Furthermore, the servos are supplied with a torque signal  $T = \overline{T} \pm T_{\Delta}$ , which consists of a common value  $\overline{T}$ and a differential value  $T_{\Delta}$ .



Figure 4-4: Adaptive springs concept based on torque feedback control.

In order to explain this *electronically controlled suspension* concept, first the effect of the servos on the airfoil rotation  $\alpha$  and translation h is illustrated using the four illustrations given in Figure 4-4:

- $T_{\Delta} = 0$ ,  $\overline{T} = 0$  Figure 4-4a shows the system at rest, where the center of the red linkage, to which the airfoil is connected, shows no rotation  $\alpha = 0$  and translation h = 0.
- $T_{\Delta} > 0, \ \bar{T} = 0$  Figure 4-4b shows that when the blue linkages are supplied with a differential torque  $T_{\Delta} > 0$ , this results in a rotation  $\alpha > 0$  around the center of rotation of the airfoil.
- $T_{\Delta} = 0, \ \bar{T} > 0$  Alternatively, when the two motors are supplied with a collective torque  $\bar{T} > 0$ , see Figure 4-4c, the center of the airfoil translates vertically h > 0.
- $T_{\Delta} > 0, \ \overline{T} > 0$  When we combine the differential and collective torque signals, see Figure 4-4d, we obtain an arbitrary translation h > 0 and rotation  $\alpha > 0$  of the airfoil.

Note that while the examples use positive values of  $T_{\Delta}$  and  $\overline{T}$ , the concept works the same for negative values, only then it will result in a negative rotation  $\alpha < 0$  and/or negative translation h < 0.

In order to make this concept behave as if it were a rotational and translational spring and damper the differential torque signal  $T_{\Delta}$  is used to control the pitching moment on the airfoil and the common torque signal  $\overline{T}$  is used to control the vertical force on the airfoil. The relation between these signals will be treated in the next section on the control scheme. Figure 4-5 shows the suspension setup which was provided for the experiments and the concept as illustrated in Figure 4-4 containing the different linkages can easily be recognized. Furthermore, Figure 4-5c shows how the morphing airfoil is attached to the suspension and Figure 4-5d shows the suspension setup attached to the back of the wind tunnel.



(a) Overview



(b) Close-up



(c) Morphing airfoil fixed to suspension



(d) Attached to the back of the wind tunnel

Figure 4-5: Electronically controlled suspension setup.

### 4-2-2 Control scheme

The imaginary springs should supply a force  $F_s$  and torque  $T_s$  as a function of the vertical displacement h and angular displacement  $\alpha$ , respectively. Furthermore, to incorporate damping it should also supply a force  $F_d$  and torque  $T_d$  as a function of the vertical velocity  $\dot{h}$  and angular velocity  $\dot{\alpha}$ , respectively. These spring and damper forces and torque values are defined in Eq. (4-2) as follows:

$$F_s = k_h \cdot h \qquad F_d = c_h \cdot \dot{h}$$
  

$$T_s = k_\alpha \cdot \alpha \qquad T_d = c_\alpha \cdot \dot{\alpha}$$
(4-2)

These force and torque values should then be supplied using the  $T_{\Delta}$  and  $\overline{T}$  signals of the servo motors, which is shown in Figure 4-6.



Figure 4-6: Control scheme of the electronically controlled suspension.

#### 4-2-3 Encountered problems

Now that the required background knowledge on the setup has been presented the encountered problems during the use of the experimental setup will be discussed in this section.

#### Vibrations introduced by the controller

The first problem is the fact that the controller of the suspension subsystem introduces vibrations in the airfoil even when no lift force caused by the wind is present. These vibrations show up when the rotational spring constant  $k_{\alpha}$  passes above a certain threshold and increase when the  $k_{\alpha}$  is further increased.

In order to detail the origin of this problem it is important to first look into how the Dynamixel MX-106R servos are controlled.

As mentioned in the previous section presenting the concept, the servos should deliver a specified torque at the motor shaft. However, the servos are in practice current controlled, which is equivalent as torque is directly proportional to the current. Important to note here is that the output signals of the Dynamixel have a finite resolution, which in case of the current signal means that the range  $I \in \{-9.2115 \ A, \ 9.2115 \ A\}$  is divided into 4095 steps, resulting in a resolution of 4.5 mA per step (Robotis, 2015). However, because the morphing airfoil and the suspension setup are lightweight components, a current of 4.5 mA already rotates the airfoil noticeably.

In other words, the MX-106R servos are significantly overpowered for this application, which comes at the cost of a very coarse signal resolution. Therefore, because the reference current signal is rounded off to the nearest multiple of 4.5 mA it makes quite a difference when for example the signal changes from 2.2 to 2.3 mA, which results in a relatively large current change in the servo of 0 to 4.5 mA.

If we now investigate the measured signals on the setup, it was found that the current  $I_2$  to the second servo motor is approximately equal to zero around the equilibrium point, depicted in Figure 4-4a, as the required torque  $\bar{T}$  to keep the airfoil at height h = 0 approximately equals the torque  $T_{\Delta}$  to keep the airfoil at angle  $\alpha = 0$ , such that they cancel out  $T_2 = \bar{T} - T_{\Delta} \approx 0$ . When we now return to the problem at hand, we see that for increasing  $k_{\alpha}$  the reference current in the second servo  $I_2$  increases up to the point where it passes 2.25 mA and the servo rounds this value to 4.5 mA. Unfortunately, around the current I = 0 the servos often show an overshoot in the current when the first nonzero reference value is supplied and the motors start moving. This unfavorable start up behavior of the servo in turn results in a slightly lower reference current  $I_2$  to compensate for the overshoot of the airfoil angle  $\alpha$ , see Figure 4-7. This process repeats itself and the airfoil starts to vibrate.



**Figure 4-7:** A coarse signal resolution in combination with an overshoot around  $I_2 = 0$  causes vibrations.

In practice this is confirmed by the fact that the vibrations stop when a small (positive or negative) torque is supplied by hand to the airfoil, such that the current  $I_2$  is no longer oscillating around  $I_2 = 2.25$  mA and the switching stops.

#### Friction

A second problem is friction, which is present both in the servo motors and the profile rail guides which guide the vertical motion. Both sources of friction will shortly be discussed and the consequences are highlighted.

Because the Dynamixel MX-106R servo motors have a very large gear reduction ratio of 255:1 (Robotis, 2015) the motor requires only little torque to turn the output shaft. On the other hand, when we turn the output shaft the opposite is true and a small friction force of the motor is amplified such that a large torque is required to move the shaft.

For most applications this is preferable as only the position of the output shaft is of interest. However, in this application it is a problem because the springs react to changes in rotation  $\alpha$  and translation h and for these changes to be observed the motor shafts should move. The external forces, for example the lift force caused by the wind passing the airfoil, should first overcome this friction in the motors before a change in angle  $\alpha$  or height h is observed by the position encoders. And only when a change in  $\alpha$  or h is observed will the torque signals  $T_{\Delta}$  and  $\overline{T}$  be changed by the controller, see Figure 4-6.

Another source of friction are the profile rail guides which guide the vertical motion and prevent the airfoil from moving in the horizontal direction, see Figure 4-5a and 4-5b. This is clearly visible when the stiffness  $k_h$  is increased but the guides do not move. However, if I apply a force normal to the guides it starts to move, because these type of rail guides are built to operate under larger loads and the presence of friction has been optimized for such larger normal forces. Unfortunately, in this application the guides are only subjected to very small loads which results in relatively high friction.

### 4-3 System dynamics

Now that the individual components of the experimental setup have been presented and some of their experienced shortcomings have been discussed, it remains to validate whether the dynamics of the combined setup correspond to aeroelastic flutter. The measurements are performed using a dedicated laser distance sensor, see Figure 4-8, as opposed to the position encoders in the servo motors. This is to make sure that the actual position is measured, which might deviate from the motor shaft angles due to for example airfoil bending.



**Figure 4-8:** A laser distance sensor is used to measure the position of the airfoil which rotates around the elastic axis indicated by the black dot.

### 4-3-1 Airfoil bending

Initial observations for V = 0 m/s quickly showed that the airfoil is bending noticeably, while the model as described in Chapter 2 only considers the pitch and plunge degrees of freedom. By exciting the system by hand and measuring the response it was found that the bending mode has a frequency of 8.2 Hz, see Figure 4-9.

In order to check whether this is a problem for the flutter experiments, measurements are conducted with the suspension controller enabled for different rotational spring stiffness values  $k_{\alpha}$ . In this way it can be evaluated to which extend the bending mode affects the suspension controller.



(a) Height measured using laser distance sensor (b) Single-sided amplitude spectrum

Figure 4-9: The bending mode of the airfoil is located at 8.2 Hz.

These measurements are performed in the configuration as shown in Figure 4-8 where the input to the system is the scaled voltage  $U_p \in [-10, 10]$  V, which is supplied to the piezoelectric patches, indicated in blue in Figure 4-8, and the output of the system is the height  $h_l$  in mm, which is obtained using the laser distance sensor.

To sufficiently excite the dynamics in the system the piezoelectric patches are actuated with a filtered white noise signal. The filter is designed as a second order Butterworth low pass filter with a cut-off frequency of 40 Hz and the signal amplitude is scaled such that the signal spans the entire range  $U_p \in [-10, 10]$  V.

Finally, spectral analysis with frequency averaging will be applied to the measured data to estimate the frequency response of the system and evaluate the observed dynamics.

The measurements have been performed for a wind speed V = 5 m/s and the spring stiffness values  $k_{\alpha} \in \{2, 3, 3.5, 4\}$ , see Figure 4-10. Note that the spring stiffness  $k_{\alpha}$  has a unit A/rad because torque has been replace by current in the suspension controller. Furthermore, due to the problems detailed in Section 4-2-3, it was not possible to calibrate the spring stiffness such that  $k_{\alpha}$  could have the unit Nm/rad.

When looking at Figure 4-10 it is observed that for the values  $k_{\alpha} < 4$  the bending mode at 8.2 Hz, see Figure 4-9b, is dominant. However, for  $k_{\alpha} = 4$  the bending mode no longer has a significant influence and a new dominant peak at 6.2 Hz appears along with a second peak at 12.4 Hz, which is exactly twice the dominant frequency. When looking at the differences in the time domain it is observed that the system starts to oscillate significantly for  $k_{\alpha} = 4$ resulting in an increase in the Root Mean Square (RMS) value from 0.42 for  $k_{\alpha} = 3.5$  to 4.15 for  $k_{\alpha} = 4$ . The low RMS values for  $k_{\alpha} \leq 3.5$  show that the system is exciting the bending mode by small angle changes, like discussed in Section 4-2-3.

In short, for  $k_{\alpha} = 4$  the system shows more violent pitch oscillations than for lower spring stiffness values and the oscillation frequency is different to the bending mode frequency of 8.2 Hz. Furthermore, the oscillations are introduced by the suspension controller as they are also observed when no external forces are present.



**Figure 4-10:** Output spectra for a wind speed V = 5 m/s and different spring stiffness values  $k_{\alpha}$ .

### 4-3-2 Influence of the wind

The previous section showed that the suspension controller introduces vibrations in the system which increase for larger values of  $k_{\alpha}$  even when no wind is present. This already shows that the observed vibrations cannot be interpreted as being caused by aeroelastic flutter, which is a wind speed dependent phenomenon. However, the influence of the wind on the system dynamics is still of interest, as the main need for LPV system identification lies in the identification of time-varying dynamics, like the wind speed dependent dynamics in this case. Therefore, this section will investigate whether the dynamics change as a function of the wind speed.

When performing these experiments at different wind speeds, first the stiffness and damping values for the suspension controller should be selected. As discussed in Section 2-3, the suspension parameters are of great importance for the occurrence of flutter. However, in this case the vibrations are not flutter-induced and the electronically controlled suspension was shown to have some serious flaws. Therefore, the parameters will be selected based on experiments, as they cannot be calculated by simulation of the LPV flutter model of Chapter 2.

The choice of the parameters is shortly motivated as follows:

- $c_h = 0$  Measurements for different values of  $c_h$  showed no effect on the estimated frequency spectrum, therefore it was selected equal to zero.
- $c_{\alpha} = 0.01$  For a range of values for  $c_{\alpha}$  it was observed that values lower than  $c_{\alpha} = 0.01$  had no noticeable influence on the estimated spectrum. Furthermore, higher values of  $c_{\alpha}$  result in more violent oscillations but a similar frequency response in terms of the peak locations.
- $k_h = 1000$  The value of  $k_h$  mainly influences the equilibrium height  $h_0$  but does not show a significant influence on the estimated frequency spectrum. Therefore, the value  $k_h = 1000$  was selected such that the airfoil remains reasonably close to h = 0.

 $k_{\alpha} = 4.2$  The previous section showed that choosing  $k_{\alpha} \ge 4$  at V = 5 m/s results in pitch oscillations, whereas for lower values only the bending mode is excited. Additional experiments showed that for the pitch oscillations to be sustained for the entire wind speed range  $V \in [0, 10]$ , a value of  $k_{\alpha} = 4.2$  should be selected.

Furthermore, a non-zero angle of attack  $\alpha_0$  results in larger lift forces and therefore a larger changes in the dynamics. This becomes visible when we compare the estimated frequency spectra for  $\alpha_0 = 2.5^{\circ}$  and  $\alpha_0 = 5^{\circ}$ , as shown in Figure 4-11 and 4-12, respectively.



**Figure 4-11:** Output spectra for different wind speed values and  $k_{\alpha} = 4.2$ ,  $k_h = 1000$ ,  $c_{\alpha} = 0.01$ ,  $c_h = 0$ , and  $\alpha_0 = 2.5^{\circ}$ .



**Figure 4-12:** Output spectra for different wind speed values and  $k_{\alpha} = 4.2$ ,  $k_h = 1000$ ,  $c_{\alpha} = 0.01$ ,  $c_h = 0$ , and  $\alpha_0 = 5^{\circ}$ .

The estimated output spectra as shown in Figure 4-11 and 4-12 show a decreasing oscillation frequency for increasing wind speed both for  $\alpha_0 = 2.5$  and  $\alpha_0 = 5$ . Furthermore, when we

look more closely at the frequency of the multiple peaks which show up in the spectra they are all an integer multiple of the first dominant frequency (around 7 Hz). This suggests that the system actually has a single resonance frequency and the remaining resonances are so-called *superharmonics* which occur in non-linear systems (Rao and Fah, 2011).

Clearly then, the estimated frequency spectra shows a single mode along with multiple superharmonics instead of a separate pitch and plunge mode like expected. When looking at the torque signals  $T_{\Delta}$  and  $\bar{T}$  which control the stiffness and damping of the pitch and plunge modes, respectively, it shows that  $T_{\Delta}$  has a significantly larger contribution to the combined torque than  $\bar{T}$ . In other words, the single observed mode is mainly excited by the rotational spring and damper.

The goal of this section was to investigate whether the dynamics are wind speed dependent, which has been shown to be the case, see Figure 4-11 and 4-12. Furthermore, the measurements showed that a single mode is observed which is mainly excited by the rotational spring and damper controller  $C_{\alpha}$ , see Figure 4-6.

### 4-4 Recommendations

For future use of the experimental setup this section will present some recommendations regarding the encountered problems which were described in the preceding sections.

#### Morphing airfoil

Firstly, while the morphing airfoil works well and the piezoelectric patches have successfully been used to excite the system by deforming the trailing edge, the obtained deformations are quite small, which limits the change in aerodynamic lift which can be obtained. Therefore, the deformations could be magnified by using a thinner stainless steel plate for future designs.

Next, the elastic axis of the available airfoil was moved significantly to the trailing edge of the airfoil, see Figure 4-5c and 4-8, which resulted in significant airfoil bending. If the elastic axis would be located closer to the leading edge of the airfoil, the steel part connecting the elastic axis to the frame of the airfoil would be shorter which in turn decreases the bending.

#### Suspension

Regarding the electronically controlled suspension, its vibration problems could be solved by using servos with a finer signal resolution, which in practice would mean using less powerful motors. Furthermore, by adding some weight to the system the equilibrium or steady-state current could be increased or decreased such that the start up behavior around  $I_2 = 0$  is circumvented.

Furthermore, there is the problem of friction in the motors and guides. The motor friction is inherent to the concept of a servo and therefore could only be changed by using a different type of actuator. Alternatively, one could try to compensate for this behavior, which requires additional sensors to measure the actual torque on the shaft. For the vertical rail guides a different setup should be considered to reduce the friction. Like for the motor friction, one could also try to compensate for it using additional control strategies.

# 4-5 Synopsis

In order to validate the LPV identification algorithms on experimental data an experimental setup was provided to perform flutter experiments. This chapter presented this setup and showed that, due to friction and controller related problems, the behavior of the setup does not resemble aeroelastic flutter. However, vibrations are induced by the controller and the dynamics of this system were observed to be wind speed dependent, which still allows for the use of LPV identification.

This process of using the LPV identification algorithms from Chapter 3 with experimental data will be the topic of the next chapter.

# Chapter 5

# **Experimental identification**

Chapter 3 discussed four LPV model identification algorithms and showed their performance in simulation. Furthermore, to show the applicability of these algorithms on experimental data a physical setup was provided consisting of a morphing airfoil and an elastic suspension. Unfortunately, this setup was found to be unable to demonstrate the phenomenon of flutter as shown in the previous chapter. However, the measurements also showed that the dynamics are wind speed dependent and therefore time-varying, which still justifies the use of the LPV identification methods.

Therefore, this chapter will apply the LPV identification methods on experimental data and discuss the performance and highlight the encountered problems. First, the way in which the data is processed will be treated along with the application of LTI identification methods on the data to obtain local models for the *local* and *glocal* LPV identification methods. Next, the individual LPV identification algorithms are shortly discussed and important parameter choices are motivated. Finally, the results will be presented and discussed.

# 5-1 Data processing

Before we turn to the topic of data processing it is important to explain how the data is obtained, which has already partly been discussed in Section 4-3-1. In short, the system is excited using a frequency limited white noise signal  $U_p$  (V), which is supplied to the voltage amplifier which in turn drives the piezoelectric patches. Furthermore, a laser distance sensor is used to measure the position  $h_l$  in mm of the airfoil, as depicted in Figure 4-8.

Because we are dealing with LPV identification the scheduling sequence should also be measured, which is the wind speed V (m/s) in this case. The wind speed inside the wind tunnel is regulated between 0 and 10 m/s using a potentiometer on the outside of the casing and the setup uses this voltage  $U_w$  over the potentiometer, which ranges from 0 to 5 V, to calculate the wind speed as  $V = 2U_w$  in m/s. The wind tunnel can thus not track a predefined wind speed signal, but requires the user to control the wind speed by hand, after which the measured signal can be used for the LPV identification. When looking at the frequencies of interest for the experimental setup, the estimated frequency spectrum as shown in Figure 4-11 showed that the dominant frequency occurs around 7 Hz and all other peaks are superharmonics located at multiples of this dominant frequency. Furthermore, because a fourth order model will be identified like in Chapter 3 only two of these (second-order) resonance peaks can be fitted, which are the peaks which occur around 7 and 14 Hz.

If we now want to re-sample the data to fit these frequencies we should keep in mind that the new sampling frequency should be lowered with an integer multiple. Therefore, given that the measurement data is currently sampled at 2 kHz an integer multiple n = 50 was selected, such that the new Nyquist frequency equals 20 Hz and the peaks at 7 and 14 Hz are contained in this range.

Furthermore, the experiment duration was selected equal to 300 s which, given the sampling frequency of 40 Hz, results in 12,000 data points in each data set. Unlike in Chapter 3, this chapter uses the same number of 12,000 data points for the *local* (constant wind speed) and global (varying wind speed) measurements.

Next, an 8th order Butterworth band pass filter is used to reject frequencies outside the range  $f \in [0.2, 18]$  Hz. The low frequency (f < 0.2 Hz) content from the signal is ignored because, due to friction in the vertical rail guides, the vertical movement often gets stuck at different constant offset values. Furthermore, the top of the range was selected at f = 18 Hz to prevent the peak located around 21 Hz to interfere, which is a superharmonic of the dominant frequency at 7 Hz.

## 5-2 LTI identification

Now that the data has been processed, like discussed in the previous section, it will now be used for LTI identification. This is a crucial step in the process, because the *local* and *glocal* LPV identification methods will fit their LPV model to these local LTI models. In other words, they will only be as good as the set of LTI models they are provided with, which explains why this step will be discussed in more detail in this section.

As explained in Section 3-3-1, the PBSID<sub>opt</sub> algorithm (Chiuso, 2007) will be used to identify local LTI models of order  $\hat{n}_x = 4$ . When using this algorithm, the value of two important parameters should be selected, that is the past window p and the future window  $f \geq \hat{n}_x$ .

Regarding the past window p, in theory  $p \to \infty$  should result in an unbiased model. However, in practice there is a risk of over-fitting, due to the use of finite-length data sequences (Van der Veen et al., 2013). By comparing the the LTI models to the estimated frequency spectra from the previous chapter for different values of p it was found that for values higher than p = 30 the fit did not improve. On the other hand, lower values of p decrease the fit for lower frequencies. Therefore, the past window was selected as p = 30.

The next parameter to select is the future window  $f \ge \hat{n}_x$ . The experiments showed that the choice of f has a significant influence on the pole locations of the local LTI models. This is illustrated by Figure 5-1, where the discrete-time pole locations are shown for  $V \in$  $\{0, 2.5, 5, 7.5, 10\}$  m/s and  $f \in \{4, 5\}$ . Note that the markers, which indicate a pole, turn darker for an increasing wind speed.



**Figure 5-1:** Effect of the future window f on the (discrete-time) pole locations for a range of wind speeds  $V \in \{0, 2.5, 5, 7.5, 10\}$  m/s.

When comparing Figure 5-1a and 5-1b, it is clear that the mode located close to the edge of the unit disk shows no noticeable changes for f = 4 and f = 5, while the other mode does show larger differences. If we now look at the frequency corresponding to this mode, it concerns the superharmonic at 14 Hz, as shown in Figure 4-11b. By comparing the LTI models to the estimated frequency spectra as presented in the previous chapter, f = 5 resulted in the best fit.

In order to further improve the results, Tikhonov regularization with generalized cross validation regularization parameter selection is used as a part of the  $PBSID_{opt}$  method (Van Wingerden and Verhaegen, 2009). Furthermore, the input and output signals are scaled to have a variance equal to one when they are used in the identification algorithm and afterwards the identified model is re-scaled to the original units to obtain the required LTI model.

Finally, if we compare the measured data with the simulation results of the local LTI models, we observe a VAF equal to zero for the first four operating points and a slightly larger value for the last operating point. When looking at the measured response, see Figure 5-2, it shows little to no damping and a behavior which seems to correspond to limit cycle oscillations, which is a non-linear phenomenon. The simulated response of the LTI model on the other hand does show damping. Note that the oscillations will not completely be damped, because the trailing edge flap is exciting the system.

Therefore, it appears that even when the wind speed is kept constant, other phenomenon make for a clearly non-linear system of which its behavior cannot be captured by an LTI model. However, it is still interesting to see whether the *local* and *glocal* methods are able to fit an LPV model to these local LTI models and see how these models relate to a *global* LPV model, which does not use the LTI models. Therefore, the next section will try to answer these questions.



**Figure 5-2:** Comparison of the measured response for V = 5 m/s with the simulated response of the corresponding local LTI model.

### 5-3 LPV identification

Now that the previous sections processed the data and identified a set of local LTI models, this section will concern the process of obtaining an LPV model using the *global*, *local*, and *glocal* LPV identification algorithms which were treated in Chapter 3.

First, the individual algorithms are treated to show which potential advantages and shortcomings have been encountered when used with experimental data and thereafter the section will be concluded with a comparison of the three algorithms.

### 5-3-1 Global: PBSID<sub>opt</sub>

Like with the LTI PBSID<sub>opt</sub> algorithm, the global LPV PBSID<sub>opt</sub> algorithm (Van Wingerden and Verhaegen, 2009) requires the user to set the future and past window  $f_g$  and  $p_g$ . Section 5-2 showed that f = 5 was selected for the LTI case and this is also adopted for the LPV case, such that  $f_g = 5$ .

When looking at the past window  $p_g$  it is not possible to use p = 30 like in the LTI case, because for values  $p_g > 8$  the computer already runs out of memory. These numerical problems which quickly occur when  $p_g$  is increased are also referred to as the 'curse of dimensionality', see Section 3-2-1. This problem is illustrated in Figure 5-3, where the required time to run the algorithm was measured for several past window values  $p_g$  and an exponential function was fitted to the data.

The 'curse of dimensionality' thus quickly appeared and is already a serious drawback when only a 4th-order LPV is identified.

Another aspect of a *global* method is the dependence on the scheduling signal which is used; in this case the wind speed signal. To illustrate this issue, Figure 5-4 shows the pole locations of the *global* LPV model evaluated at different wind speeds, where for Figure 5-4a and 5-4b two data sets with a different wind speed signal  $V \in [0, 10]$  m/s were used.

Clearly, the pole locations differ between the two data sets and when we compare it with the LTI model pole locations, see Figure 5-1b, we see that it closely corresponds to Figure 5-4a. Therefore, it appears that something went wrong for the second data set, which is confirmed by looking at the wind speed signal, which spends much less time at lower wind



Figure 5-3: Exponential increase in computational time for increasing past window  $p_g$ .



**Figure 5-4:** The way in which the wind speed signal varies has a significant effect on the identified LPV model.

speed values, compared to the first data set. This explains why the poles in Figure 5-4 mainly differ in location for the lower wind speed values and why Figure 5-4b even shows an unstable pole for V = 0 m/s.

Now that the two main drawbacks have been discussed, which was the 'curse of dimensionality' and the influence of the scheduling sequence on the results, it remains to compare the obtained *global* LPV model pole locations to the only available benchmark, the local LTI models. This comparison is shown in Figure 5-5, where the pole locations, frequencies, and damping are plotted for a wind speed  $V \in \{0, 2.5, 5, 7.5, 10\}$  m/s. Note that the LTI model poles are indicated with dots and the poles of the *global* LPV model are indicated using circles.

Clearly, the *global* LPV model fits the local LTI models very well and both models show a complex pole at a frequency of 7 Hz and 14 Hz, which corresponds to the measurements which were discussed in Section 4-3.



**Figure 5-5:** Comparison of the identified *global* LPV model poles, shown as circles, with the local LTI model poles, shown as dots.

### 5-3-2 Local: SMILE

The next LPV identification algorithm to discuss is the *local* SMILE algorithm (De Caigny et al., 2014), which interpolates the local LTI models to obtain an LPV model. This method requires no specific parameters to be set, apart from the model dimensions which are also used by the other algorithms.

Regarding the encountered problems, the only problem which sometimes occurs is an LPV model which is not stable at all local points. However, this is often the result of fitting the measured dynamics into a too restrictive model description, which is unable to successfully describe the system. Furthermore, this is a problem for most LPV identification methods, unless a certain constraint is introduced which enforces stability for a specified scheduling sequence. This is also illustrated in Figure 5-4b, where for low wind speeds an unstable pole was identified by the *global* LPV identification algorithm.

A comparison of the *local* LPV model with the LTI models for constant wind speed values is shown in Figure 5-6.

The pole locations shown in Figure 5-6a correspond to the LTI models very well and are also similar to the results shown in Figure 5-5a for the *global* LPV model. Only slight differences are observed for the damping, see Figure 5-6c, while the frequencies fit almost perfectly, see Figure 5-6b.



**Figure 5-6:** Comparison of the identified *local* LPV model poles, shown as circles, with the local LTI model poles, shown as dots.

### **5-3-3** Glocal: $\mathcal{H}_2$ norm

Now that the global and local methods have been treated, it remains to discuss the glocal  $\mathcal{H}_2$ norm based H<sub>2</sub>NL method by Petersson (2013). Regarding this method, Section 3-2-3 already showed that, due to the non-convexity of the optimization problem, a good (stable) initial LPV model should be supplied, for which the LPV model obtained using the local SMILE method is used. However, the previous section showed that when the model parameterization is too restrictive, the obtained LPV model can be unstable at some of the local operating points.

The requirement of a stable LPV model to initialize the optimization problem which is used in a *glocal* method could thus be a drawback, given that this stable model should also be relatively close to the optimal model to prevent the optimization to get stuck in local minima.

This brings us directly to the main drawback of this *glocal* method, which is the non-convex optimization and the resulting problem of getting stuck in local minima.

In order to validate the *glocal* LPV model, the pole locations are compared to the LTI model pole locations in Figure 5-7.

Figure 5-7a shows that the pole locations fit the local LTI models very well, as also shown by the pole frequency and damping in Figure 5-7b and 5-7c, respectively. When comparing the results with the *local* LPV model, see Figure 5-6, slight differences are visible, but overall the results are very similar. This can be explained by the fact that the *local* LPV model was used as an initial point, which already fitted the local LTI models very well.



**Figure 5-7:** Comparison of the identified *glocal* LPV model poles, shown as circles, with the local LTI model poles, shown as dots.

### 5-3-4 Comparison

Now that the previous sections have discussed the individual LPV identification algorithms and compared them to the LTI models, this section will compare the different algorithms and try to relate the results to the theoretical results discussed in Chapter 3. Therefore, we will look at the same two aspects which were treated in Chapter 3, namely *pole prediction* and *model fit*, which is evaluated using the VAF.

#### Pole prediction

The discussion on the individual algorithms showed that the pole locations are fitted very well by all algorithms, which shows the ability of the LPV identification methods to successfully capture the time-varying behavior of the system. However, predicting poles outside the operating regime of the wind speed  $V \in [0, 10]$  m/s was not considered.

The maximum pole amplitude, which was also used in Chapter 3, is shown in Figure 5-8 for  $V \in [0, 15]$  m/s.

Clearly, the three methods predict very different pole locations, as the glocal H<sub>2</sub>NL method shows unstable poles for V > 10 m/s, while the global and local method do not predict instability for this wind speed range.

Unfortunately, no measurement data is available to validate these predictions. Furthermore, it should be noted that because the behavior of the experimental setup was shown to not correspond to aeroelastic flutter, see Chapter 4, no conclusions can be drawn on the topic of flutter prediction based on the predictions shown in Figure 5-8.



Figure 5-8: Pole prediction differences among LPV identification algorithms.

#### Model fit

Another aspect of the identified LPV models which was evaluated in Chapter 3 was the model fit to the measured data. In other words, where the pole fit looks at the identified dynamical models, the model fit can also be evaluated by comparing the output signals obtained in simulation or measurement in the time domain, as was shown in Figure 3-3.

In the case of Chapter 3, the reference signal was obtained by simulation of the *true* LPV model, while in this case the reference signal is obtained from measurements on the experimental setup.

When we compute the VAF for the simulated output of the *global*, *local*, and *glocal* LPV model, it was found that it is approximately equal to zero, which indicates that the fit is very bad. To further investigate this low fit, the time domain signals of the measured response and the simulated response for the *global* LPV model are shown in Figure 5-9.



Figure 5-9: Comparison of the global LPV simulation output with the measured response.

At the start, the signals coincide very well, but the simulated output quickly decreases in amplitude and shows a slightly different oscillation frequency resulting in a very bad fit very quickly. Note that the response does not dampen out completely as the system is excited by the trailing edge flap. It thus appears that the physical system is too complex to be described by the four state LPV model parameterization which is used in this chapter. This corresponds to the model fit of the *local* LTI models, as described in Section 5-2, which showed a similar response and bad fit.

It thus appears that the significant problems in the setup, which were detailed in the previous chapter, result in non-linear dynamics which are not a function of the wind speed and can therefore not be adequately modeled by the selected LPV model structure.

### 5-4 Synopsis

Although the experimental setup was shown to be unsuitable to model the dynamics of flutter, it did show time-varying dynamics and the process of identifying an LPV model has been discussed in this chapter.

This process of identification starts with data processing and it was shown that in order to successfully capture the relevant dynamics the data should be sampled at the right frequency and a band pass filter can be used to specify a frequency band of interest.

Once the data has successfully been re-sampled and filtered, LTI identification should be applied to obtain a set of local models which describe the system for a fixed wind speed value. These models were identified using the  $PBSID_{opt}$  LTI algorithm (Chiuso, 2007) for which the selection of the future and past window was shown to require special attention, as it has a significant influence on the identified LTI models.

Next, the LPV identification algorithms were applied on the prepared data and LTI models, after which several important drawbacks were highlighted.

For the PBSID<sub>opt</sub> global LPV identification algorithm the two main drawbacks are the computational burden, or 'curse of dimensionality', and the dependence on the scheduling sequence content. The 'curse of dimensionality' turned out to become a problem very quickly due to the exponential increase in computational time as a function of the past window.

On the other hand, the SMILE *local* LPV identification algorithm shows no significant practical problems, given that the model parameterization is not too restrictive to fit the dynamics present in the data.

Finally, the  $H_2NL$  glocal LPV identification algorithm its main drawback is the nonconvexity of the optimization problem, which often results in sub-optimal results, due to the optimization algorithm getting stuck in local minima. This is partly circumvented by supplying a good initial LPV model, given that this model is stable for each local operating point.

Overall, all three identified LPV models show a very good fit within the operating regime, that is the range of fixed wind speeds for which the LTI models have been identified. Regarding the ability to predict the flutter speed by looking at the poles outside this range of wind speeds, no conclusions can be drawn, given that the practical setup does not represent the phenomenon of flutter. Furthermore, due to non-linearities in the setup the time-domain signals could not be successfully compared between measurement and simulation.

# Chapter 6

# **Conclusions and recommendations**

Flutter is an aeroelastic phenomenon resulting in catastrophic vibrations and failure of structures such as aircraft and bridges (Mukhopadhyay, 2003). Another recent example of such structures are wind turbines, which will become more prone to flutter in the near future, due to the increasing rotor diameters (Griffith and Ashwill, 2011; Resor et al., 2012; Owens et al., 2013). Therefore, the topic of flutter prediction is still relevant today and the ability to predict the flutter speed is of the utmost importance.

When predicting the behavior of a dynamic system, often LTI models are used. However, flutter is a time-varying phenomenon as it depends on the wind speed. Recursive methods use LTI methods to capture LTV dynamics by forgetting older data, this forgetting however takes time and introduces a delay in the prediction. Therefore, the use of batch LPV system identification was proposed to predict the flutter speed using pre-flutter data.

Before diving into the identification topic, the phenomenon of flutter was explained in detail and an LPV flutter model was presented, which will be used for the flutter prediction simulations.

First, LPV identification methods were divided into *global*, *local*, and *glocal* methods (Lovera et al., 2013). The *global* methods identify the LPV model from a single experiment, during which the scheduling sequence is varied. The *local* and *glocal* methods on the other hand use multiple experiments where the scheduling sequence is kept constant and afterwards interpolate these local results to obtain an LPV model.

From these different groups of methods one *global*, one *local*, and two *glocal* methods were selected from literature to validate the concept of flutter prediction.

However, it was observed that the  $\mathcal{H}_{\infty}$ -norm based glocal method (Vizer et al., 2013) is currently only implemented in continuous-time, while the other methods all work in discretetime. Further investigation showed that this made for an unfair comparison, because an error is introduced by transforming the local discrete-time LTI models into continuous-time for the  $\mathcal{H}_{\infty}$ -norm based glocal method. Therefore, this method was not further included in the comparison for flutter prediction.

The remaining three methods were used to identify an LPV flutter model from pre-flutter data for a SNR equal to 5 and 40. Using these LPV models the flutter speed was predicted and it was shown that it could be predicted to within 10 % even in the presence of significant noise (SNR = 5), which showed the potential of LPV model identification to predict the flutter speed. For increasing noise the pole prediction deteriorates quickly at low and high wind speeds, while the point of instability is still predicted within reasonable bounds. When comparing the results between the different methods it shows that the *global* PBSID<sub>opt</sub> algorithm overall shows the smallest prediction error.

Furthermore, a trade-off between the bias and variance was observed for an increasing noise level, where for significant noise (SNR = 5) the *local* method outperforms the *global* method. Moreover, the *glocal*  $\mathcal{H}_2$ -based method is able to further improve both the bias and variance of the *local* estimate. This both shows the capability of the *glocal* method to outperform the *local* method, but also shows the requirement of a good initial LPV model for the non-convex optimization problem as also pointed out by Petersson (2013).

These different aspects of the four LPV identification algorithms used in the comparison were also summarized in Table 3-4.

In order to validate the results obtained in simulation on an experimental setup, a setup was provided to perform flutter experiments. However, as it was the first time that this setup would be used, the behavior of the setup should be described in order to check whether it corresponds to aeroelastic flutter.

The setup consists of a morphing airfoil and an electronically controlled elastic suspension. While the morphing airfoil was shown to function properly, albeit the deformations of the trailing edge were relatively small, significant problems were encountered in the suspension system.

First it was shown that the suspension controller excites the system in such a way that it is constantly vibrating, even when it is not externally excited. The origin of the problem was found to lie in the coarse signal resolution of the servo motors in combination with a unfavorable friction around the equilibrium point of the motors. Furthermore, due to the large gear reduction of the servos it requires a significant torque on the output shaft to move the motor, which is problematic, because the lift force acting on the airfoil should first overcome this friction before the suspension can counteract this movement like a conventional spring would.

Another source of friction was showed to be the vertical rail guides, which are designed to work under large loads and for a small load, like the case in this setup, show a relatively large friction.

After the individual components of the setup were discussed, the response of the combined system was measured for different constant wind speed values, which showed that the system vibrates in a single mode and higher frequency peaks in the spectrum correspond to so-called *superharmonics*, which occur in non-linear systems (Rao and Fah, 2011). Furthermore, the frequency of the dominant mode, along with the superharmonics, vary as a function of the wind speed. Therefore, LPV model identification can still be used to capture this time-varying behavior.

Before the actual LPV identification was applied on the experimental data, the need for re-sampling and filtering was shown to select the frequency band of interest for the model. Furthermore, the importance of selecting the right past and future window parameters in the LTI  $PBSID_{opt}$  algorithm was stressed, as it directly changes the pole locations of the identified models.

Unfortunately, a comparison between the measured data and the simulated response of the identified LTI models showed a very bad fit, which was ascribed to the non-linear behavior of the setup, which could not be captured by the LTI models. Nevertheless, it is still interesting to see whether the *local* and *glocal* methods are able to fit an LPV model to these local LTI models and see how these models relate to a *global* LPV model, which does not use the LTI models.

Regarding the use of the LPV identification methods on experimental data several of the main drawbacks of these methods were highlighted.

First, the global method suffers from the 'curse of dimensionality', which in practice puts an upper bound on the past window because the algorithm runs out of memory. In this thesis an upper bound of p = 8 was found, which shows that the PBSID<sub>opt</sub> global method (Van Wingerden and Verhaegen, 2009) is only suitable to identify low order LPV models. Furthermore, the variation of the scheduling sequence was shown to influence the results, which shows that special attention should be paid to the scheduling signal such that the entire operating regime is sufficiently visited during the experiment.

The *local* SMILE method (De Caigny et al., 2014) on the other hand showed little practical problems, as long as the LPV model parameterization is not too restrictive to describe the measured dynamics.

Finally, the glocal  $H_2NL$  method (Petersson, 2013) requires a stable initial LPV model which is relatively close to the optimal model, such that the optimization does not get stuck in local minima. This greatly limits the potential of this method to be used stand-alone, but this can be solved by using it in combination with for example the *local* SMILE method to provide a suitable initial model.

Regarding the validation of the concept of flutter prediction by using experimental data, no conclusions can be drawn at this point, because the behavior of the experimental setup was shown to not correspond to flutter.

### 6-1 Recommendations for future research

Regarding the LPV identification algorithms and their use for flutter prediction several recommendations are made:

- The  $\mathcal{H}_{\infty}$ -norm based glocal method showed good results in continuous-time, which makes it interesting to adapt this method for use with discrete-time systems.
- While the global and local methods presented in this chapter work well with unstable systems, because the PBSID<sub>opt</sub> identification can take place in closed-loop with a stabilizing controller, this is currently not the case for the glocal methods, because the  $\mathcal{H}_2$  and  $\mathcal{H}_{\infty}$  norm are not defined for unstable systems. Note that while the norms are defined on the error system  $E_l = ||\mathcal{H}_l \mathcal{H}(\theta_l)||$ , for an unstable local LTI model  $\mathcal{H}_l$ , the error system is also unstable, because the summation of two systems combines the individual state matrices A in a block diagonal matrix of which the eigenvalues are simply a combination of the eigenvalues of the individual state matrices A. It might

therefore be interesting to include the stabilizing controller in the optimization problem to allow the *glocal* methods to identify unstable systems in closed-loop.

• Because the experimental setup was shown to be unsuitable to validate the flutter predictions, new experiments are required to validate the conclusions on flutter prediction in simulation in practice.

Furthermore, regarding the experimental setup the following recommendations are made:

- The morphing airfoil could use a thinner stainless steel plate to which the piezoelectric patches are bonded, in order to obtain larger deformations and larger changes in the lift force.
- Furthermore, the elastic axis of the airfoil is currently moved far to the trailing edge using a thin steel component, which introduces significant airfoil bending. By moving the elastic axis closer to the leading edge, this steel component is shortened and the influence of airfoil bending is attenuated.
- The suspension setup could be improved by using smaller motors with a finer signal resolution. Furthermore, some counterweights could be added to keep the current to the second motor away from  $I_2 = 0$  and circumvent the friction occurring at this point.
- Friction in the servos could perhaps be compensated for using control when additional sensors are included to measure the actual torque at the elastic axis, instead of using the current in the motors to estimate the torque. Alternatively, the friction could be removed or attenuated by changing the servos for a different type of actuator which exhibits less friction.
- As the vertical rail guides show significant friction they should be replaced or altered to reduce the damping which is introduced in the vertical movement.

# Bibliography

- Apkarian, P. and Noll, D. (2006). Nonsmooth  $\mathcal{H}_{\infty}$  synthesis. Automatic Control, IEEE Transactions on, 51(1):71–86. doi:10.1109/TAC.2005.860290
- Åström, K. J. and Wittenmark, B. (2011). Computer-controlled systems: theory and design. Courier Dover Publications.
- Barker, J. M. and Balas, G. J. (2000). Comparing linear parameter-varying gain-scheduled control techniques for active flutter suppression. *Journal of Guidance, Control, and Dynamics*, 23(5):948–955. doi:10.2514/2.4637
- Bergami, L. (2008). Aeroelastic stability of a 2D airfoil section equipped with a trailing edge flap. Technical Report Risø-R-1663(EN), Risø National Laboratory, Roskilde, Denmark.
- Bilgen, O., Kochersberger, K. B., and Inman, D. J. (2013). Wing morphing design using macro-fiber composites. *Smart Composites: Mechanics and Design*, page 169.
- Billah, K. Y. and Scanlan, R. H. (1991). Resonance, Tacoma Narrows bridge failure, and undergraduate physics textbooks. *Am. J. Phys*, 59(2):118–124.
- Bisplinghoff, R. L., Ashley, H., and Halfman, R. L. (1996). Aeroelasticity. Dover Science, New York.
- Chiuso, A. (2007). The role of vector autoregressive modeling in predictor-based subspace identification. *Automatica*, 43(6):1034–1048. doi:10.1016/j.automatica.2006.12.009
- De Caigny, J., Camino, J. F., and Swevers, J. (2011). Interpolation-based modeling of MIMO LPV systems. *IEEE Transactions on Control Systems Technology*, 19(1):46–63. doi:10.1109/TCST.2010.2078509
- De Caigny, J., Pintelon, R., Camino, J. F., and Swevers, J. (2014). Interpolated modeling of LPV systems. Control Systems Technology, IEEE Transactions on, 22(6):2232–2246. doi:10.1109/TCST.2014.2300510
- Fung, Y.-c. (1993). An introduction to the theory of aeroelasticity. Courier Dover Publications.

Master of Science Thesis

- Griffith, D. T. and Ashwill, T. D. (2011). The Sandia 100-meter all-glass baseline wind turbine blade: SNL100-00. Technical Report SAND2011-3779, Sandia National Laboratories, Albuquerque, June.
- Hansen, M. H. (2007). Aeroelastic instability problems for wind turbines. *Wind Energy*, 10(6):551–577. doi:10.1002/we.242
- Houtzager, I. (2011). Towards Data-Driven Control for Modern Wind Turbines. PhD thesis, Delft Center for Systems and Control, Mechanical Maritime and Materials Engineering, Delft University of Technology.
- Houtzager, I., van Wingerden, J.-W., and Verhaegen, M. (2012). Recursive predictorbased subspace identification with application to the real-time closed-loop tracking of flutter. *IEEE Transactions on Control Systems Technology*, 20(4):934–949. doi:10.1109/tcst.2011.2157694
- Karpel, M. (1981). Design for active and passive flutter suppression and gust alleviation. PhD thesis, Stanford University; Department of Aeronautics and Astronautics.
- Lau, E. and Krener, A. (1999). LPV control of two dimensional wing flutter. In Proceedings of the 38th IEEE Conference on Decision and Control, volume 3, pages 3005–3010. IEEE. doi:10.1109/CDC.1999.831394
- Lee, K. W. and Singh, S. N. (2007). Global robust control of an aeroelastic system using output feedback. *Journal of guidance, control, and dynamics*, 30(1):271–275. doi:10.2514/1.22940
- Ljung, L. (1999). System Identification: Theory for the User. Prentice Hall information and system sciences series. Prentice Hall PTR, 2nd edition.
- Lobitz, D. W. (2005). Parameter sensitivities affecting the flutter speed of a MW-sized blade. Journal of solar energy engineering, 127(4):538–543.
- Lovera, M., Bergamasco, M., and Casella, F. (2013). Lpv modelling and identification: An overview. In Sename, O., Gaspar, P., and Bokor, J., editors, *Robust Control and Linear Parameter Varying Approaches*, volume 437 of *Lecture Notes in Control and Information Sciences*, pages 3–24. Springer Berlin Heidelberg. doi:10.1007/978-3-642-36110-4\_1
- Micro-Epsilon (2015). Micro-epsilon optoNCDT 1302 compact laser sensor. http://www.micro-epsilon.com/displacement-position-sensors/laser-sensor/ optoNCDT\_1302/index.html. Accessed: 06-09-2015.
- Mozaffari-Jovin, S., Firouz-Abadi, R., Roshanian, J., and Ghaffari, A. (2013). Discussion on "Active aeroelastic control of 2-d wing-flap systems operating in an incompressible flowfield and impacted by a blast pulse" by librescu et al., journal of sound and vibration 283 (3-5) (2005) 685-706. Journal of Sound and Vibration, 332(13):3351–3358. doi:10.1016/j.jsv.2012.10.006
- Mozaffari-Jovin, S., Firouz-Abadi, R., Roshanian, J., and Ghaffari, A. (2014). Corrigendum to: Discussion on "active aeroelastic control of 2-D wing-flap systems operating in an incompressible flowfield and impacted by a blast pulse" by librescu et al., journal of sound and vibration 283 (3–5) (2005) 685–706 [journal of sound vibration 332 (13) (2013) 3351–3358]. Journal of Sound and Vibration, 333(25):7084–7087. doi:10.1016/j.jsv.2014.06.021

- Mukhopadhyay, V. (2003). Historical perspective on analysis and control of aeroelastic responses. *Journal of Guidance, Control, and Dynamics*, 26(5):673–684. doi:10.2514/2.5108
- Olds, S. D. (1997). Modeling and LQR control of a two-dimensional airfoil. Master's thesis, Virginia Polytechnic Institute and State University.
- Owens, B. C., Griffith, D. T., Resor, B. R., and Hurtado, J. E. (2013). Impact of modeling approach on flutter predictions for very large wind turbine blade designs. In *AHS 69th Annual Forum*.
- Petersson, D. (2013). A Nonlinear Optimization Approach to  $\mathcal{H}_2$ -Optimal Modeling and Control. PhD thesis, Department of Electrical Engineering, Linköping University, Sweden.
- Petersson, D. and Löfberg, J. (2014). Optimisation-based modelling of LPV systems using an -objective. *International Journal of Control*, 87(8):1536–1548. doi:10.1080/00207179.2013.878477
- Prime, Z., Cazzolato, B., Doolan, C., and Strganac, T. (2010). Linear-parameter-varying control of an improved three-degree-of-freedom aeroelastic model. *Journal of Guidance*, *Control, and Dynamics*, 33(2):615–619. doi:10.2514/1.45657
- Rao, S. S. and Fah, Y. F. (2011). Mechanical vibrations.
- Resor, B. R., Owens, B. C., and Griffith, D. T. (2012). Aeroelastic instability of very large wind turbine blades. In *European Wind Energy Conference Annual Event, Copenhagen, Denmark.*
- Robotis (2015). Robotis dynamixel MX-106R. http://support.robotis.com/en/product/ dynamixel/mx\_series/mx-106.htm. Accessed: 06-09-2015.
- Seiler, P., Balas, G. J., and Packard, A. (2012). Linear parameter-varying control for the X-53 active aeroelastic wing. In *Control of Linear Parameter Varying Systems with Applications*, pages 483–512. Springer. doi:10.1007/978-1-4614-1833-7\_19
- Shamma, J. S. and Athans, M. (1992). Gain scheduling: potential hazards and possible remedies. *Control Systems*, *IEEE*, 12(3):101–107. doi:10.1109/37.165527
- Skogestad, S. and Postlethwaite, I. (2007). Multivariable Feedback Control: Analysis and Design. John Wiley & Sons, Ltd, 2nd edition.
- Smart Blade (2015). Smart blade visualization wind tunnel. http://smart-blade.com/ products-services/visualization-wind-tunnel.html. Accessed: 06-19-2015.
- Smart Material Corp. (2015). Smart material home of the MFC. http://www. smart-material.com/MFC-product-main.html. Accessed: 06-09-2015.
- TREK, Inc. (2015). TREK model PZD700A M/S high-voltage power amplifier / piezo driver. http://www.trekinc.com/products/PZD700Ams.asp. Accessed: 06-09-2015.
- Van den Boom, T. and De Schutter, B. (2012). Optimization in Systems and Control: Lecture Notes for the Course SC4091. TU Delft.

M. Visser

- Van der Veen, G., van Wingerden, J.-W., Bergamasco, M., Lovera, M., and Verhaegen, M. (2013). Closed-loop subspace identification methods: an overview. *IET Control Theory & Applications*, 7(10):1339–1358. doi:10.1049/iet-cta.2012.0653
- Van Wingerden, J.-W. and Verhaegen, M. (2009). Subspace identification of bilinear and LPV systems for open-and closed-loop data. Automatica, 45(2):372–381.
- Verdult, V. and Verhaegen, M. (2005). Kernel methods for subspace identification of multivariable LPV and bilinear systems. Automatica, 41(9):1557–1565.
- Verhaegen, M. (1994). Identification of the deterministic part of MIMO state space models given in innovations form from input-output data. Automatica, 30(1):61–74.
- Visser, M., Navalkar, S., and van Wingerden, J.-W. (2015). LPV model identification for flutter prediction: A comparison of methods. In *Proceedings of the 1st IFAC Workshop on Linear Parameter Varying Systems*.
- Vizer, D., Mercere, G., Prot, O., Laroche, E., and Lovera, M. (2013). Linear fractional LPV model identification from local experiments: an  $\mathcal{H}_{\infty}$ -based optimization technique. In *IEEE 52nd Annual Conference on Decision and Control (CDC)*, pages 4559–4564. IEEE. doi:10.1109/CDC.2013.6760594
- Wang, Q.-M. and Cross, L. E. (1999). Constitutive equations of symmetrical triple layer piezoelectric benders. Ultrasonics, Ferroelectrics, and Frequency Control, IEEE Transactions on, 46(6):1343–1351. doi:10.1109/58.808857
- Wingerden, J.-W. v. (2008). Control of wind turbines with 'Smart' rotors: proof of concept & LPV subspace identification. PhD thesis, Mechanical Maritime and Materials Engineering, Delft University of Technology.
- Wright, J. R. and Cooper, J. E. (2008). Introduction to aircraft aeroelasticity and loads. John Wiley & Sons.
- Zhang, K. and Behal, A. (2014). Continuous robust control for aeroelastic vibration control of a 2-D airfoil under unsteady flow. *Journal of Vibration and Control*, page 1077546314554821. doi:10.1177/1077546314554821
- Zhou, K., Doyle, J. C., Glover, K., et al. (1996). Robust and optimal control. Prentice Hall New Jersey.
## Glossary

## List of Acronyms

 $\mathbf{3mE}\,$  Mechanical, Maritime and Materials Engineering.

**DCSC** Delft Center for Systems and Control.

**LFR** Linear Fractional Representation.

**LLS** Linear Least Squares.

 ${\bf LPV}$  Linear Parameter-Varying.

LTI Linear Time-Invariant.

 $\mathbf{LTV}$  Linear Time-Varying.

MFC Macro Fiber Composite.

**MIMO** Multiple Input, Multiple Output.

**RMS** Root Mean Square.

**SMILE** State-space Model Interpolation of Local Estimates.

 ${\bf SNR}\,$  Signal to Noise Ratio.

**VAF** Variance Accounted For.