UNIVERSITY OF SOUTHAMPTON

Iterative Learning Control for Load Management in Wind Turbines with Smart Rotor Blades

by

Weronika Natalia Nowicka

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
Faculty of Engineering and Physical Sciences
School of Electronics and Computer Science

June 2020

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING AND PHYSICAL SCIENCES
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

by Weronika Natalia Nowicka

Control of aerodynamic loads is a crucial issue in keeping wind energy economically competitive with traditional energy sources. Loads on wind turbine blades can be managed through collective and individual pitch control, however recently there has been significant research on application of active flow control devices which can alter the flow locally, spanwise on the blade. This work investigates the use of iterative learning control for load control in wind turbines with smart rotor blades providing a significant extention of the previous research on model-free design and a substantial contribution on the model-based approach. Iterative learning control is capable of rejecting periodic disturbances in systems performing repetitive tasks and in this particular application it is used to modify the blade section aerodynamics such that the fluctuations in load due to periodic disturbances on the blades are minimized.

At first, a computational fluid dynamics flow model with a basic structure iterative learning control law is used, where the controller's gains are chosen without the use of a model of the dynamics akin to auto-tuning design. Model-free design demonstrates the potential of this algorithm for wind turbines control but is limited in what it can deliver, especially as testing is computationally ineffective as it requires running the full computational fluid dynamics simulation each time. Subsequently, model based design is considered where a Proper Orthogonal Decomposition based reduced order model is constructed and used to design and test the norm optimal iterative learning control scheme. Construction of a reduced order model requires running the full computational fluid dynamics simulation only once and various controllers can be designed and evaluated using this low-dimensional model. The performance of designed controllers is evaluated in simulation for the state-space model and in full computational fluid dynamics test and the results show that aerodynamic load on the blade can be successfully controlled by iterative learning control.

Contents

List of Figures

Nomenclature

t	time
k	step within a cycle in discrete representation
Δt	time step
$u(t)$, $u(k)$	control input signal
$y(t)$, $y(k)$	output signal
y_{ref}	desired output
$e(t)$, $e(k)$	tracking error
$\mathbf{x}(t)$, $\mathbf{x}(k)$	state vector
A, B, C, D	state space matrices
i	trial number
α	trial length
μ	controller's gain
G	the matrix representing input to output relation of the system
R, Q	weighting matrices
$*$	dimensional quantity
c	chord length
V_∞	mean free stream velocity
p	pressure
ρ	air density
L	lift
n, j	indices
γ_n	strength of vortex panel n
λ_n	strength of source panel n
$\mathbf{x} = (x, y)$	spatial coordinates
$\mathbf{V}_0$	free stream component of the flow
$\mathbf{v}_p(\mathbf{x}, t)$	velocity field generated by the panels
$\mathbf{v}_{vj}$	the velocity field generated by an individual vortex
Γ	strength of a vortex
v_f	time function assuring that the control is turned on smoothly
s	distance along the streamline
T	period of oscillation

ω	angular frequency of oscillations $2\pi/T$
A	amplitude of oscillations
L_{tar}	target value of the lift used as desired output y_{ref}
μ_0	proportional gain
μ_1	ILC gain (fixed)
$\mu(i)$	ILC gain (varying from trial to trial)
δ	controller's phase shift
$\mathcal{L}_2$	error 2-norm
$\mathcal{L}_\infty$	error ∞-norm
λ	coefficient determining actuator's speed of response

z	physical space
ζ	computational space
r	radius (grid)
θ	angle (grid)
J	area mapping
$\mathbf{u} = (u, v)$	the flow velocity with components in x and y direction
$\mathbf{u}_m$	steady mean component of the flow
$\hat{\mathbf{u}}$	unsteady component of the flow
$\mathbf{u}_0 = \binom{A}{0}$	oscillatory component of the flow
$\mathbf{u}_c$	velocity field generated by an airfoil with unit circulation
$\boldsymbol{\phi}_j(\mathbf{x})$, $\boldsymbol{\phi}_i(\mathbf{x})$	spatial POD modes
$a_j(t)$, $a_i(t)$	POD coefficients
$<,>$	inner product
N	number of modes used for the flow reconstruction
κ	sum of the control inputs from the time the control is switched on
E_j	energy in mode j

Abbreviations:

AFC	Active Flow Control
AoA	Angle of Attack
BEM	Blade Element Momentum
CFD	Computational Fluid Dynamics
COE	Cost of Energy
DNS	Direct Numerical Simulation
ILC	Iterative Learning Control
LES	Large Eddy Simulation
LMI	Linear Matrix Inequalities
LQG	Linear Quadratic Gaussian
MOR	Model Order Reduction
MPC	Model Predictive Control

MPPT	Maximum Power Point Tracking
MSE	Mean Square Error
NOILC	Norm Optimal ILC
NS	Navier Stokes
O&M	Operation and Maintenance
PCA	Principal Component Alalysis
PID	Proportional plus Integral plus Derivative
PIV	Particle Image Velocimetry
POD	Proper Orgonal Decomposition
RANS	Reynolds-averaged Navier-Stokes
RC	Repetitive Control
ROM	Reduced Order Model
SVD	Singular Value Decomposition
TE	Trailing Edge
TRPOD	Trust Region Proper Orthogonal Decomposition
TSR	Tip Speed Ratio
UAV	Unmanned Aerial Vehicles

Declaration of Autorship

I, Weronika Natalia Nowicka, declare that the thesis entitled *Iterative Learning Control for Load Management in Wind Turbines with Smart Rotor Blades* and the work presented in the thesis are both my own, and have been generated by me as the result of my own original research. I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University;
- where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- parts of this work have been published as: Nowicka et al. (2017) and Nowicka et al. (2018)

Signed: ....

Date: ....

Acknowledgements

I gratefully acknowledge my supervisors Prof. Owen Tutty, Prof. Eric Rogers and Dr Bing Chu for all the help I got from them during my research.

Also, I would like to thank my family, especially my parents, and my friends for giving sense to my life. This thesis wouldn't have been written without their support.

Special thanks to Southampton University Zumba+ Society for taking away the stress associated with my PhD, to my snooker team for supporting me winning BUCS and to my VLCCC friends for making our office the best working environment in the world.

It all started in Southampton.

To happiness...

Chapter 1

Introduction

Wind energy is a fast-growing field recognized as a cost effective and environmentally friendly energy source (Pao and Johnson, 2009). This trend will continue as limited fossil fuel reserve and energy security are one of the main current public concerns. Over the past three decades the typical size of wind turbine rotors has increased drastically and is expected to reach 190m in 2030 (Wiser et al., 2016). Accordingly, the aerodynamic loads on the blades have increased. Passive control (e.g. stall) and collective pitch control of such huge and flexible structures become ineffective as for big horizontal-axis wind turbines the wind varies in space and time, hence it is different not only for different blades but also along each blade. This last fact motivates the need for developing advanced active control systems and strategies in order to react to disturbances more precisely, i.e. locally (Houtzager et al., 2013). Aerodynamic load control of wind turbines involves altering the lift force on the rotor blades. This aim can be obtained in numerous ways: by varying flow velocity (variable speed), blade incidence angle (variable pitch) or blade length. The modern control approach includes modification of blade section aerodynamics and is known as active flow control (AFC). Advanced active control devices such as trailing-edge flaps, microtabs, plasma actuators and vortex generators are termed 'smart devices' or 'smart rotors' (Johnson et al., 2008).

An AFC approach requires the choice of a suitable control algorithm which is critical for the performance of the overall system. Various advanced control approaches have been investigated recently for wind turbine applications. Literature surveys are given in Rezaei (2015) and Collis et al. (2004) and the methods considered include, among others, nonlinear control, optimal control, robust control and predictive control. For the majority of these approaches a model of the system must be known. The other possible scheme that could be applied for wind turbines application is iterative learning control (ILC). The results of the recent research of Blackwell et al. (2016) showed a good potential of ILC for aerodynamic load control in wind turbines and these results are sufficient to merit further investigation in this field.

ILC is a feedforward approach for systems operating in a repetitive manner over a finite duration. It is applicable to systems that complete the same finite duration task over and over again and is often combined with a feedback control loop. Each execution is commonly termed a trial (or pass) and the finite duration is known as the trial length. Once each trial is complete resetting to the starting location occurs ready for the start of the next trial. Once a trial is complete all information generated is available for use in constructing the control signal for the next trial and thereby improve performance from trial-to-trial. In fact, this strategy can also be applied to systems where there is a stoppage between successive executions of a task, i.e., it is not required to reset to the starting location.

The main purpose of ILC controlled systems is to track a predefined reference trajectory such that the tracking error is minimal. The information about the error and control input signal from the past trials is stored in system's memory as data and it is used to calculate the control input signal for the future trials so the convergence of the tracking error through repeated trials is achieved. Such convergence cannot be obtained with classical algorithms such as proportional–integral–derivative controller (PID) or optimal control where the error remains the same from trial to trial. Moreover, in contrast to complex control schemes, e.g. neural networks based schemes, in ILC the structure of the controller remains the same during the operation whereas only the control input signal is modified (Bristow et al., 2006). For these reasons ILC can be considered as a very powerful and relatively simple tool.

One starting point for background on ILC, including controllers with dynamics, are the survey papers Bristow et al. (2006) and Ahn et al. (2007b). Successful applications of ILC include gantry robots, e.g., Cai (2009) and robotic-assisted upper limb stroke rehabilitation, e.g. Meadmore et al. (2012) and Rogers and Tutty (2016). Several attempts at applying ILC schemes for aerospace applications have been reported. In Laks et al. (2011) the use of ILC to control of the transitions between variable pitch wind turbine operating regions is proposed, however only a few applications of ILC for the load control of turbines with smart rotor blades can be found in current research literature. In Blackwell (2015) and Blackwell et al. (2016) such an approach is implemented and tested using computational fluid dynamics (CFD) model of the flow. This last paper examines model-free ILC design for mitigation of load disturbances, but model-based ILC laws are not considered therein and are indicated as an obvious area for future research.

The purpose of this research is to examine and compare various ILC algorithms for load management in wind turbines with smart rotors, where a major contribution is the construction of a finite dimensional approximate model using a Proper Orthogonal Decomposition (POD) setting and the designing of model-based ILC schemes for this model. Firstly, the CFD panel code simulating the flow passing a wind turbine smart blade from Blackwell et al. (2016) is used (with some modifications) to test several model-free ILC designs. Subsequently, the CFD code is used to generate the data snapshot for POD–based modelling and a state-space representation which is suitable for designing model based ILC schemes is derived. Finally, the design procedure of such control schemes is shown and the simulation results are given. The designs are validated in CFD simulation and a detailed comparison of the results is given.

This thesis is organized as follows. In Chapter 2 the literature review of the field is given. This includes the background on wind turbine load control, the description of AFC approach and the theoretical background on ILC. The flow configuration and the CFD model used to simulate the flow past an airfoil are described in Chapter 3. The performance of such a system controlled by a fixed gain model-free ILC is evaluated and the control scheme with a varying gain ILC is developed. In Chapter 4 model order reduction of the flow via POD is developed and the resulting models are verified by comparing the output of the models with the output from the CFD simulation. In Chapter 5 the model obtained in Chapter 4 is used to design model-based ILC schemes and the performance of such schemes is evaluated. Conclusions and future research directions are given in the last chapter.

Published output to date from this research are as follows:

- W. Nowicka, "Iterative Learning Control for Load Reduction in Wind Turbines with Smart Rotor Blades", a poster presented at American Control Conference, Boston 2016;
- W. Nowicka, B. Chu, O. Tutty, E. Rogers, "Load Reduction in Wind Turbines with Smart Rotors Using Trial Varying Iterative Learning Control Law", Proceedings of the American Control Conference, pp. 1377–1382, Seattle 2017;
- W. Nowicka, B. Chu, O. Tutty, E. Rogers, "Wind Turbine Aerodynamic Load Fluctuation Reduction Using Model Based Iterative Learning Control", Proceedings of the American Control Conference, pp. 6384–6389, Milwaukee 2018.

Related research output includes:

- Hladowski, Y.Chen, W. Nowicka, K. Galkowski, E. Rogers, "Iterative Learning Control Laws with Full Dynamics", Proceedings of the American Control Conference, pp. 1377–1382, Boston 2016;
- L. Hladowski, K. Galkowski, W. Nowicka, E. Rogers, "Repetitive Process Based Design and Experimental Verification of a Dynamic Control Law", Control Engineering Practice, Vol. 46, No. 6, 157–165, 2016.

Chapter 2

Literature Review

In this chapter the literature review on load control of wind turbines is given together with the details on active flow control (AFC) approach, control theory and Iterative Learning Control (ILC).

2.1 Trends in Wind Energy Production

Wind energy is acknowledged as a cost effective source of energy, however the current trend is to increase the rotor diameter to maximize the energy capture and minimize the cost per MWh. Wind turbines together with other renewables are believed to assure energy security and mitigation of greenhouse gas emissions in the future. According to the Global Status Report (REN21, 2018) more than 90 countries in the world use commercial wind turbines to produce energy. Over the past 20 years the world total installed capacity increased from 7GW to 539GW in 2017 with more than half of the world's wind power capacity added over a past few years and 11% added in 2017 alone (Fig. 2.1). It is expected that the global capacity will reach 700GW in 2020. The typical

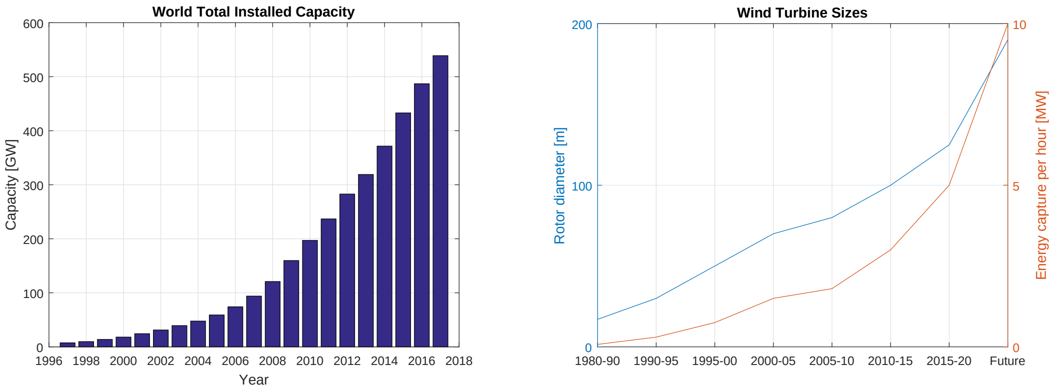


Figure 2.1: Evolution of wind energy convertion systems

rotor diameter has increased drastically from 17m in 1980 to 125m currently and the near future prototypes of offshore turbines are expected to have a rotor diameter of 190m by 2030 (Fig. 2.2). In many countries, particularly in Europe, old wind turbines are being replaced with fewer, larger, taller, more efficient and more reliable machines with improved software and control mechanisms.

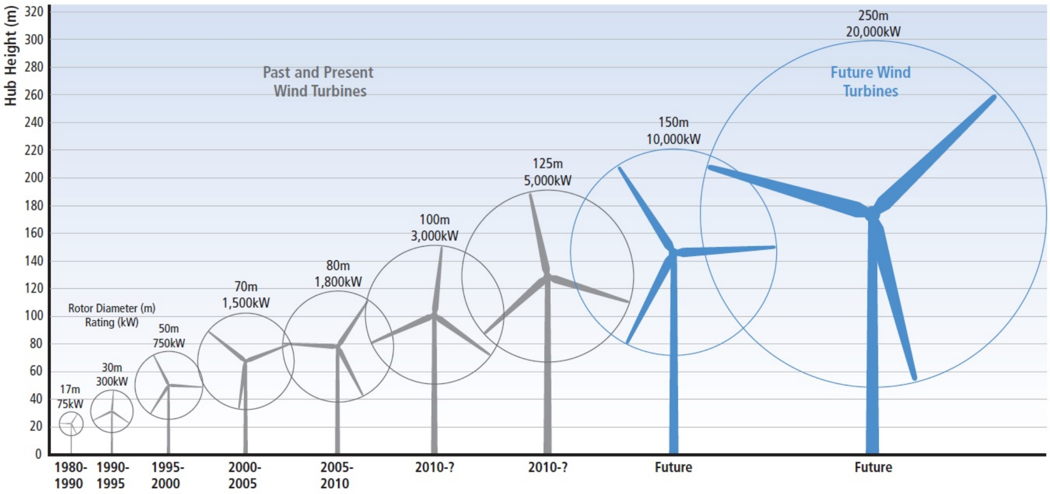


FIGURE 2.2: Increase in size of commercial wind turbines (Edenhofer et al., 2012)

A developing trend in wind energy provision is to operate offshore. An estimated 4.3GW of offshore capacity was added in 2017 with total world capacity exceeding 18.8GW (REN21, 2018). The distance from the shore has also increased with an average of over 43km in Europe. Reduction of cost, especially with long distance offshore operation, is a crucial issue. To remain economically competitive with traditional energy sources the cost of energy (COE) should be lowered. This cost is calculated using equation (2.1) (Johnson et al., 2010), which is the ratio of the capital cost and operation and the maintenance cost (O&M) to the total energy capture in a turbine's lifetime.

$$COE = \frac{Capital\,Cost + O\,\&M\,costs}{Lifetime\,Energy\,Capture} \tag{2.1}$$

Possible methods of decreasing COE include increasing the turbine size and therefore the energy capture, reducing the amount of materials to lower the capital cost or reducing the downtime and O&M costs by constructing more reliable turbines (Johnson et al., 2008). Reduction of O&M can be achieved by designing effective control systems for rotor loads that will provide a decrease in fatigue and extreme loads on the components. This, in turn, will reduce maintenance, improve system reliability and produce an increase in component lifespan. For these reasons the research in wind turbine load control field is highly required.

2.2 Control of Wind Turbines

Wind turbine control plays a very important role as it enables a better energy capture together with alleviation of mechanical and aerodynamical loads. Wind turbine control objectives include improving power production in its safe operating region (below rated wind speed) and preventing the unsafe operation in high wind speeds (above rated speed) by limiting the rotor speed and torque (Bianchi et al., 2007; Menezes and Araujo, 2018). A typical wind turbine power curve with its operation regions is shown in Fig. 2.3.

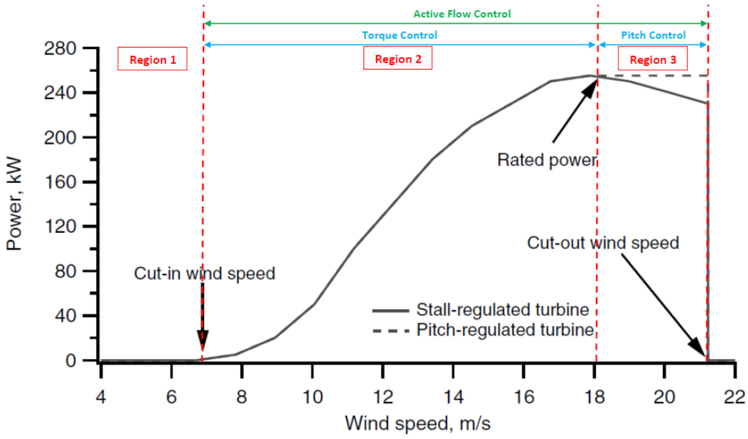


FIGURE 2.3: Operating regions of a typical wind turbine (Manwell and McGowan, 2009)

In Region 1 the wind speed is too low to generate enough power to compensate for the cost of operation and turbines remain switched off. In Region 2 the main objective of control is to assure the optimal power production using Maximum Power Point Tracking (MPPT) strategy. The optimum power coefficient depends on a Tip Speed Ratio (TSR) defined as $TSR = \frac{R\omega}{v}$, where R is the rotor radius, ω is the rotational speed of the rotor and v is the wind speed. Thus, for every wind speed there exists an optimal rotational speed and more power can be generated by adjusting the rotor speed.

In Region 3 the main control objectives change and the power should be limited to the rated power in order to ensure safe operation and mitigate mechanical loads. This can be achieved via pitch control. Since the transition between Region 2 and 3 is challenging in terms of the control strategy some sources also distinguish the Region 2.5, where the main objective is to ensure a smooth transition (Pao and Johnson, 2009; Laks et al., 2011). When the wind speed exceeds the cut-out value turbines must be switched off (Region 4).

Summarising, several tasks must be completed before implementing suitable control systems: (1) defining control objectives, (2) choosing appropriate control strategy, (3) deciding how the strategy should be implemented physically. In this work we consider a problem of aerodynamic load control and the remainder of this section focuses on this particular approach. The detailed description of other control strategies (e.g. power control) is not given here and it can be found in other sources, including research papers cited above.

2.2.1 Aerodynamic Modelling and Load Control

Wind turbines control design requires estimation of power and aerodynamic loads acting upon the blades and the rotor. The integrated aeroelastic model for predicting performance and structural dynamics is essential for designing of specific parts and the overall system (Sorensen, 2011).

2.2.1.1 Aerodynamic Modelling

One of the early reviews of the rotor performance analysis is given in Hansen and Butterfield (1993). The most widespread performance analysis method, proven simple and accurate for various flow conditions, is the blade element momentum theory (BEM) by Glauert. This approach assumes that the blade can be divided into a number of independent sections and the aerodynamic forces for each section can be calculated using axial and angular momentum balances. The resulting forces for all sections are summed giving the overall forces. BEM method has proven to be reasonably accurate with low computational cost in many simple cases. The loads are computed by combining the tabulated 2D airfoil data with 1D momentum theory, hence obtaining the airfoil data from wind-tunnel measurements or computations is required as a prestep. A well known example of the 2D airfoil data is XFOIL code (Drela, 1985). However, to construct the airfoil data for a rotating blade it needs to be corrected for the 3D and rotational effects (Sorensen, 2011). BEM method is based on the steady flow assumption and doesn't account for the wake effect. Including dynamic inflow effects requires finding a method better suited for analysis of complex flows, where one of possible approaches is using vortex wake methods suitable for simulating incompressible unsteady flows. Despite its limitations BEM forms a backbone for other techniques including the methods for calculating the dynamic structural loads which a rotor will experience due to aerodynamic forces.

Hansen and Butterfield (1993) divide unsteady aerodynamic analysis into two main categories: dynamic stall experiments and dynamic inflow. Initially, in 1980s, it was believed that the unsteady effects could be neglected for performance and load analysis. However, later it was shown that dynamic stall occurs under a variety of inflow conditions as wind turbines are subjected to wind shear and atmospheric turbulence as well as the effects of other turbines in wind farms. This results in number of forces that vary in time and space and should be included in analysis. Hansen and Butterfield (1993) suggest that dynamic stall and dynamic inflow can increase some of the operating loads by up to 100%.

In the more recent review by Sorensen (2011) results from computational fluid dynamics (CFD) simulations of wind turbines and their wakes are given along with the conventional BEM method. CFD tools have been an active research since 1970, however their use was limited by their computational complexity. With advances in computing the first full Navier-Stokes simulation for a complete rotor blade was carried out in Sorensen and Hansen (1998) and many more methods have been published ever since. Two main approaches for CFD calculations include solving Reynolds-averaged Navier-Stokes equations (RANS) and performing large-eddy simulations (LES). Despite improvements in computing power these methods remain computationally expensive. In Snel (1998) using the Euler equations is indicated as an alternative to using the Navier-Stokes equations. The Euler equations can be seen as the non-viscous form of the Navier-Stokes equations and could form a good approximation. Vortex wake methods are an example of the Euler solution, where the trailing and shed vorticity is convected into the wake with the total velocity and the evolution of the vortex wake is calculated in time. Such approach is used in this research. Clearly there are a range of approaches of increasing complexity that can be used when modelling the flow over a turbine blade. Here we will use a relatively simple and cheap approach in order to concentrate on the control aspects of the problem. We do, however, include nonlinear and unsteady effects in our model. The details on the modelling assumptions are given in section 3.1.

While aerodynamic analysis of single wind turbines is a mature subject recent research efforts focus on flow and turbulence phenomena arising from the fact that wind turbines are usually clustered in small groups or in large wind farms. In wind farms wakes created by individual turbines interact with each other and with atmospheric flow phenomena leading to increased turbulence. The existing methods for modelling of wakes are also reviewed in Sorensen (2011) and the recent review of wind farm flow phenomenon and design methods is given in Stevens and Meneveau (2017). Another review by Vermeer et al. (2003) focuses on the effect of both near and far wakes for single turbines and wind farms. All reviews agree that there are still gaps in the knowledge that must be addressed as all of the methods are based on some sets of assumptions. Simpler tools are unable to fully represent the physics of the flow and CFD simulations are expensive computationally and not always suitable for engineering design and efficient implementation (Stevens and Meneveau, 2017).

Finally, most of the modelling methods are yet to be examined in detail for smart rotor control applications. In Andersen (2007) a Deformable Trailing Edge Geometry (DTEG) has been added to a dynamic stall model predicting the unsteady aerodynamic forces. The deflection angle of the DTEG (i.e. the flap) is limited to ±5 degrees. In Lackner and Kuik (2010) the quasisteady model of a wing with a trailing-edge flap is presented and the detailed analysis of the unsteady effects are indicated as the future work direction.

Summarising, the detailed aerodynamic modelling and control of wind turbines with smart rotors (including various sensors and actuators) remain an open field of reaserch. The following part of this chapter focuses on trends and aspects in aerodynamic control, details on active flow control (AFC) and suitable control design methods.

2.2.1.2 Aerodynamic Load Control

Wind turbine blades are subject to fluctuating aerodynamic forces involving stochastic and deterministic disturbances (Blackwell et al., 2016). The stochastic disturbances occur because of the variable nature of the wind. Deterministic forces include the effects of yaw misalignment, stator-rotor interaction and atmospheric boundary layer. The load disturbances caused by effects such as wind shear, tower shadow or yaw motion are cyclic as they arise due to the rotation of the rotor (Manwell and McGowan, 2009). Wind shear (wind gradient), is a difference in wind speed or direction over a short distance in the atmosphere. Precisely, the mean speed increases with height. Moreover, the actual wind speed varies in time and direction at different locations due to turbulence (van der Tempel, 2006). Hence, the flow past the blade contains a periodic component which becomes even larger when the rotor size increases. The other effect of aerodynamic forces in downwind turbines is tower shadow, which refers to the wind speed deficit just behind the turbine's tower. The effects of wind shear, turbulence and tower shadow are illustrated in Fig. 2.4.

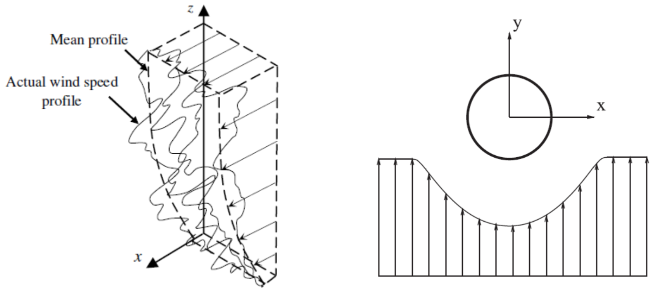


Figure 2.4: Left: Wind speed profile (van der Tempel, 2006); Right: The top view from the turbine tower illustrating the tower shadow (Houtzager et al., 2013)

The wind speed profile also depends on the type of terrain. The rougher the terrain the thicker the affected layer as shown in Fig. 2.5. For rough terrains (cities, parks) the increase in velocity with height will be more gradual and hence the difference between the velocity at the rotor tip and the rotor bottom will be higher. For flat terrain, especially sea, this difference at the same hight will be much lower (Plate, 1971).

The analysis becomes even more challenging when considering big wind farms. Various flow phenomena in wind farms are shown in Fig. 2.6.

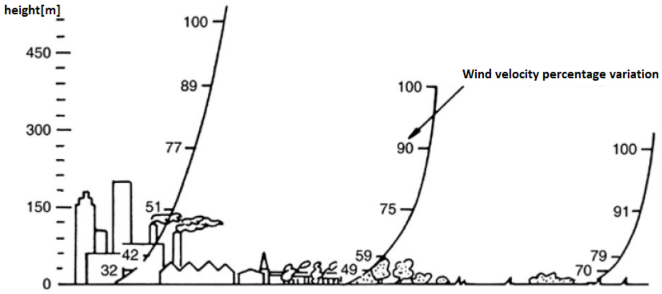


Figure 2.5: Atmospheric boundary layer (ABL) profiles over varying terrain (Katsaprakakis and Christakis, 2012)

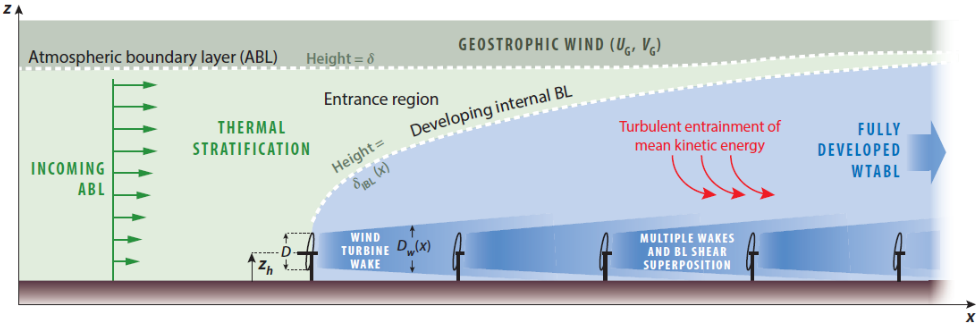


Figure 2.6: Various flow phenomena in wind farms (Stevens and Meneveau, 2017)

Both, deterministic and stochastic components, create loads that require management. There exist two basic kinds of loads: ultimate (extreme) and fatigue. Ultimate loads refer to extreme wind conditions leading to very high loads that wind turbines might experience occasionally. Turbines must be designed such that they can withstand these extreme loads. However, normal operation is also accompanied by varying loads that can cause the fatigue damage of the components. Fatigue damage can be defined as increasing inability to withstand loads applied multiple times (Manwell and McGowan, 2009). That means that an old component could fail at lower loads compared to its new equivalent and therefore mitigation of fatigue loads leads to elongation of the components' lifetime.

Aerodynamic load control of wind turbines refers to the modification of the lift force on the blades. This purpose can be achieved by any of the following or their combination (Johnson et al., 2008):

- varying the rotor speed,
- varying the blade pitch angle,
- varying the blade length,
- modifying the blade section aerodynamics.

The early turbine prototypes (1950s) were controlled passively (stall regulation) and had a constant rotor speed. In 1980s the turbines with two possible speeds and full-span pitch control appeared, before the full variable speed and variable pitch rotors were developed in 1990s (Thresher and Dodge, 1998). In 2000s the individual pitch control for load reduction was introduced (Bossanyi and Hassan, 2003). As the turbine blades became larger the individual pitch actuators with advanced control schemes (e.g. Linear Quadratic Gaussian, LQG) were used. Moreover, research into 'intelligent' control of the rotor loads began. The modern approach includes more flexible structures on the blades with sophisticated control algorithms which can be implemented by the latest software tools. This new approach called active flow control incorporates devices such as trailing-edge flaps or microtabs which are called 'smart rotors'. The actuation in such systems is very fast which is a big advantage compared to traditional pitch control. An overview of load control techniques described above is given in the Fig. 2.7.

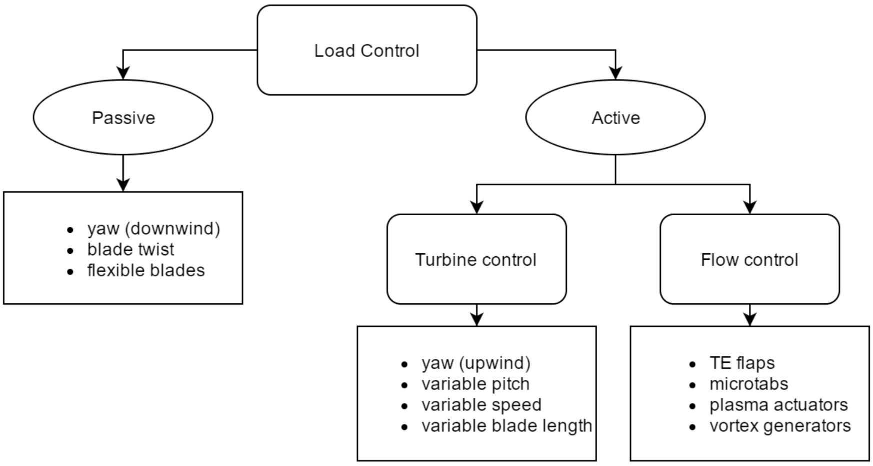


FIGURE 2.7: Control techniques (Johnson et al., 2008)

2.2.2 Active Flow Control

AFC devices are placed along the span of the rotor blade (e.g. on the trailing-edge) and act by modifying the local flow and therefore the lift (Castaignet et al., 2011). Implementation of smart devices requires placing sensors along the span of the rotor blades. In such wind turbine control system there is a master controller responsible for pitching the blades and communicating with blade controllers and the individual blade controllers responsible for active flow control. Blade controllers receive the signals from the sensors, communicate with the master controller, and then with actuators which change the system's response according to the particular control algorithm chosen during the design process. Summarising, active feedback control based on local measurements is realized (Johnson et al., 2008). A possible layout of a smart blade is shown in Fig. 2.8.

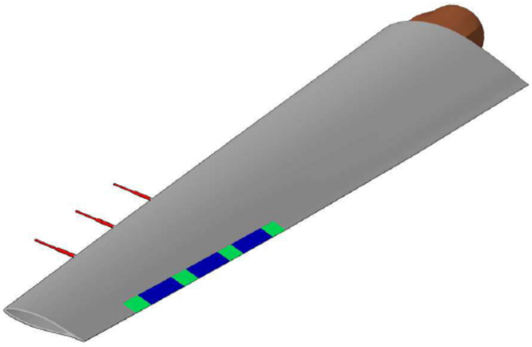


Figure 2.8: A blade with trailing-edge flaps (blue) and Pitot tubes (red) (Castaignet et al., 2011)

Johnson et al. (2008) review a number of techniques that can be used for wind turbine applications. Fifteen types of AFC devices described in the review are given in Table. 2.1. Most of these devices have been developed and implemented for other industry fields such as aircraft design or unmanned aerial vehicles (UAV) proving the concept of load reduction by using smart control. Some of them, such as trailing edge flaps and microtabs, have shown the potential for wind turbine applications in numerical experiments and wind tunnel tests. Others, such as circulation control, have been tested for aircraft or UAVs (Montanya and Marshall, 2007; Kanistras et al., 2014) but are yet to be investigated for wind turbines. In Marrant and Holten (2006) a software package based on BEM is used to assess four devices: trailing-edge flap, camber control, microtabs and active twist and the results indicate a good capability of reducing fatigue damage. In Barlas and van Kuik (2010) an extensive review of actuators, sensors and controls is given. This review suggests that there is a huge step to be taken from research stage to product stage, with more research investigation on design parameters for aerodynamic control devices needed before full wind turbine simulations are possibe. In Johnson et al. (2010) computational experiments and wind tunnel tests have shown potential for lift control using microtabs, but the system requires future research and validation on full-scale turbines. Finally, trailing-edge flaps have now been tested in a wind tunnel (Lee et al., 2014) and on full-scale turbines in Castaignet et al. (2014) (27m Vestas V27 turbine) and Berg et al. (2014) (9m CX-100 blade) but potential application on much larger commercial turbines requires further research.

The most common kinds of actuators in engineering are pneumatic, hydraulic and electrical motors. Such traditional actuators are used for the blade pitch and yaw control. AFC requires extremely fast actuation without an overly complicated mechanical system. Electrical motors can be used, e.g. for flap control, but smart material actuators are potentially more suitable because of their fast response (Barlas and van Kuik, 2010). Smart materials are able to change their properties in a controlled fashion in response to the external stimuli. They include, among others, piezoelectric actuators, shape memory polymers, and magnetostrictive materials. AFC system requires suitable sensors which

Table 2.1: AFC devices

Name	Description
Traditional Trailing-Edge Flaps	small movable flaps that alter the lift force by deflection
Nontraditional Trailing-Edge Flaps	similar to traditional trailing-edge flaps but using smart materials
Microtabs	small translational parts placed close to the trailing edge
Miniature Trailing-Edge Effectors (MiTEs)	translational devices similar to microtabs located at the trailing edge
Microflaps	rotating parts at the trailing edge
Active Stall Strips	devices that separate the flow close to the leading edge
Vortex Generators	devices placed close to the leading edge that delays local flow separation by creating a vortex
Blowing and Suction	blowing/suction devices that add/remove the high/low-momentum fluid into the boundary layer
Circulation Control	thin jets that add high-momentum fluid at the trailing-edge
Plasma Actuators	devices that alter the circulation by creating an electric field
Vortex Generator Jets	devices that create a streamwise vortex in order to delay stall
High-Frequency Micro Vortex Generator Jets	devices that generate periodic vortices by using mechanical motion
Synthetic Jets	jets similar to vortex generators but not requiring a high-momentum fluid source
Active Flexible Wall	devices placed closed to the leading edge that introduce disturbances into the boundary layer
Shape Change Airfoil	a device that change the shape of the airfoil physically

are light, have little sensitivity to temperature and long term robustness and stability of measurements. Conventional, optical or microelectromechanical (MEMS) sensors can be used. Finally, AFC requires the choice of a suitable control algorithm which is crucial for the performance of the overall system. Many approaches have been proposed ranging from a simple Proportional plus Integral plus Derivative (PID) controller to more advanced techniques.

As described at the beginning of section 2.2, before applying any controls the control objectives must be defined. In operating Region 3 (Fig. 2.3) pitching mechanism would still be used in order to limit power, whereas AFC devices would react quickly and reduce oscillatory, high-frequency loads. However, Johnson et al. (2008) gives other possible benefits of AFC if applied in other operating regions including:

- increasing the blade lift at low wind speeds and therefore allowing earlier cut-in,
- enabling the blade to operate on higher lift curve,
- aerodynamic performance improvement and noise reduction,
- countering tower shadow every revolution (downwind turbines).

Summarising, in an AFC system sensors communicate with controls updating the information about the actual flow properties and if the change is needed the controller communicates with actuators which activate the control devices. AFC devices change the local flow phenomenon that is required to be controlled (e.g. lift, drag) and the sensors measure this changes to begin the whole cycle again. Hence, active feedback flow control can be divided into three main categories: sensors/controls, actuators/devices and flow phenomena which have been described in this section and can be summarised by Fig. 2.9.

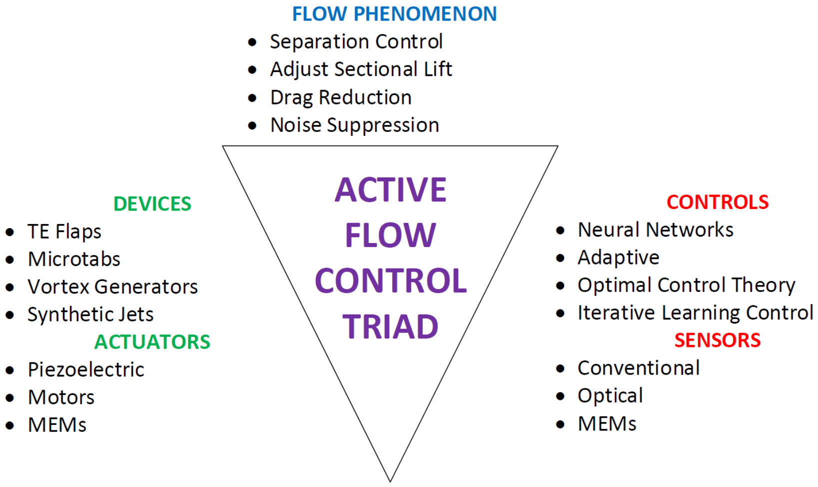


FIGURE 2.9: AFC triad (Johnson et al., 2008)

Finally, the practical limitations of all components must be addressed when designing the overall system shown in Fig. 2.9. A useful review of various actuators for active flow control and their limitations is given in Cattafesta and Sheplak (2011). This review highlights that when designing a control system various sensor issues and dynamic requirements on actuators must be addressed. Moreover, other unique aspects of flow actuation must be considered including analysis of the interraction of the fluid and actuator. Cattafesta discusses various types of actuators including fluidic, moving surface and plasma together with their strength and drawbacks and the application examples. The importance of the actuator's bandwidth and time response are also discussed. As a conclusion, in many cases not all actuators are able to meet the design criteria. For example, a rotary valve is unable to produce the excitation signal containing multiple frequency components at any instant in time. Conversely, piezoelectric or electrodynamic-driven synthetic jets or plasma do meet this requirement and and can also provide extremely fast actuation. In Kotsonis (2011) high-speed particle image velocimetry (PIV) technique is used to show that the flow reacts to actuation almost instantly with initial velocity components already starting as early as $0.1ms$ after the start of the actuation and developing within a few ms. In Narayanaswamy et al. (2010) a pulsed-plasma synthetic or sparkjet is proved to generate peak velocities of up to $250m/s$ at frequencies up to $5kHz$. These examples show that, although no device is perfect, the future of active flow control is bright, especially with incorporation of very fast smart materials (Cattafesta and Sheplak, 2011).

The modelling methods, the flow phenomena, actuators, sensors and various AFC devices have now been reviewed. The next section focuses on control algorithms that could be applied.

2.2.3 Control Design Methods

Several survey papers on control design methods that could be used in wind turbines applications have been published. In Bossanyi and Hassan (2000) a traditional closed-loop configurations such as proportional-integral (PI) and proportional-integral-derivative (PID) are indicated as starting points for designing controllers and other possible solutions such as self-tuning controllers, Linear Quadratic Gaussian (LQG), H_∞ (H-infinity), fuzzy logic controllers and neural network methods are listed. However, as the state-of-the-art on AFC was limited at the time of this publication (2000), it focuses mostly on collective and individual pitch control applications. In Collis et al. (2004) the main control approaches that are suitable for AFC are described in detail together with the issues that can appear during implementation. Collis classifies control methods into three main categories: open-loop control, closed-loop control (feedback) and optimal control.

In the open-loop operation the actuator parameters are set to constant value and do not depend on the change in the state of the flow (Collis et al., 2004). Therefore, the closed-loop configuration appears more appropriate for AFC. In the closed-loop approach the measurements from the sensors are used to alter the system state such that it goes to the target value in time. A simple example of a possible closed-loop approach is shown in Fig. 2.10. Here r denotes the reference value for the output, u system input, y output, w disturbance and v sensor noise.

The control input $\mathbf{u}(t)$ can be determined from the error value $\mathbf{e}(t)$ using, e.g. the traditional closed-loop configurations such as PI and PID which, in time domain, are given by (2.2) and (2.3) respectively (Franklin et al., 2015):

$$u(t) = k_P e(t) + k_I \int_{t_0}^{t} e(\tau) d\tau \tag{2.2}$$

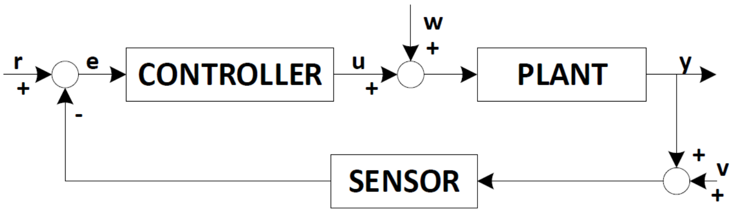


Figure 2.10: Feedback control loop

$$u(t) = k_P e(t) + k_I \int_{t_0}^{t} e(\tau)d\tau + k_D \frac{de(t)}{dt} \tag{2.3}$$

Here k_P, k_I, k_D are the controller parameters that must be tuned. While designing such controllers sufficient gain and phase margins must be allowed in order to provide robustness.

Modern control theory proposes using state-space methods for designing feedback control systems (however the name 'modern' can be misleading as it actually dates for late 50s). In such methods state-variable description of the system is used to design the controller and that usually requires including the state estimator into the system in order to improve the performance in presence of disturbances. A control input signal is then determined from the state estimate and fed back to the system as shown in Fig. 2.11. For the purposes of such a design a linear system can be described using the state-variable form (Franklin et al., 2015)

$$\begin{aligned}\frac{d\mathbf{x}}{dt} &= \mathbf{A}\mathbf{x} + \mathbf{B}u \\ y &= \mathbf{C}\mathbf{x} + Du\end{aligned} \tag{2.4}$$

where $\mathbf{x}$ is the state vector, $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and D are the state space matrices and the control signal is given by

$$u = -\mathbf{K}\hat{\mathbf{x}} \tag{2.5}$$

where $\mathbf{K}$ denotes the matrix of controller's gains and $\hat{\mathbf{x}}$ is the state estimator.

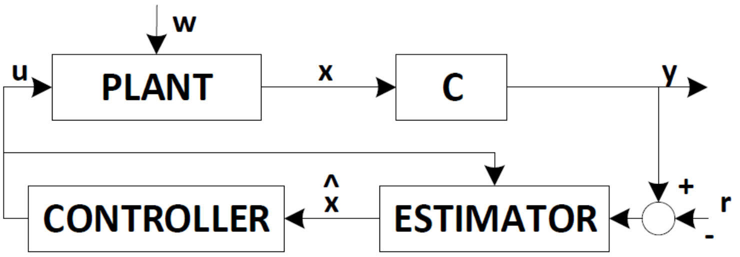


Figure 2.11: Modern control diagram (Franklin et al., 2015)

When the worst-case variant of disturbances is assumed in the controller design the approach is called $\mathcal{H}_\infty$ (H-infinity) and it guarantees that the stability will be preserved for any external disturbances. Another approach assumes the disturbances to be Gaussian white noise and is called Linear Quadratic Gaussian (LQG, $\mathcal{H}_2$). The drawback of these methods is the fact that implementation of such controllers requires the availability of the mathematical model of the state. As using full Navier-Stokes simulation is computationally expensive, reduced order models (ROM) could be used (Collis et al., 2004). The other possible approach is optimal control. In such methodology the open-loop control signal that minimizes the specified objective functional is calculated. These approaches are designed, in general, for linear problems, however, numerous studies show that they could be applied to unsteady flows. Early examples include, among others, Bewley et al. (2001) where direct numerical simulation (DNS) and optimal control theory are used for drag reduction of a turbulent flow in a plane channel and Kim (2003) which describes application of linear optimal control theory for skin-friction drag reduction in turbulent boundary layers.

In a more recent survey paper Rezaei (2015) the categorization of control algorithms that have been applied in wind industry is proposed and more methods are listed including adaptive control, nonlinear control and model predictive control (MPC). The most recent survey paper by Menezes and Araujo (2018) focuses not only on pitch control applications but also on developments in smart rotor applications controlled using MPC, H_∞ and ILC. Finally, Houtzager et al. (2013) proposed Repetitive Control (RC) for AFC applications, however the main drawback of RC is a significant decrease in performance if the controller's period doesn't match the actual disturbance signal's period.

This thesis focuses on the application of ILC for flow control in wind turbine application and a detailed description and classification of ILC algorithms is provided in the next section. More information on other methods can be found in survey papers listed above and in standard sources for control algorithms. ILC has been applied for wind turbine load control in Tutty et al. (2013) and Blackwell et al. (2016), where a simple model-free structure controller has been examined for various operating conditions. The examination of more sophisticated model-based designs is recommended for future research. In Castaignet et al. (2011) the MPC controller is used for active load reduction in the full-scale test on a turbine with a trailing-edge flap achieving 14% load reduction. In Li and Balas (2013) the adaptive control approach is examined in simulation for the system with microtabs showing a good performance and stability for vibration suppression. In Rice and Verhaegen (2010) robust and distributed control of a smart blade is discussed. This last study emphasizes the challenges in designing feedback controllers which provide high performance and are reliable to modelling uncertainties. Summarising, there is still a high demand on more research on modelling and simulation before these control methods can be applied for full-scale turbines.

2.3 Iterative Learning Control

Research on ILC theory and applications began after the first publication of the approach, widely credited to Arimoto et al. (1984, 1985). Since then it has been a very active area of research both in terms of theory and applications. The survey papers Bristow et al. (2006) and Ahn et al. (2007b) are one starting point for the literature. Also there are a number of research texts, e.g. Moore (1993) and Ahn et al. (2007a). Across all research in ILC and applications the following key features are found:

1. The method is used for systems operating in a repetitive manner, i.e. executing the same task many times. This task is denoted as trial, it is finite, and it has a fixed length.
2. After each trial the system returns to the initial position.
3. The main purpose is to follow a pre-specified trajectory and force the tracking error to decrease from trial to trial using the past experience. In the ideal case as trial number goes to infinity the error should approach zero.
4. The algorithm is non-causal (the output depends on future inputs) but possible to implement as the data from the previous trials is used: at the given point k of trial i data obtained for the next points $(k+1,\dots,k+l)$, but taken from the previous trial $i-1$, is used to make the scheme causal.

The idea of ILC is given in Fig. 2.12. Here $u_i(k)$ and $y_i(k)$ denote input and output signals at time instance k and trial i. At each trial the current input and the tracking error defined as the difference between the reference trajectory and the output $e_i(k) = y_{ref}(k) - y_i(k)$ is stored in system's memory and used by the controller to calculate the input signal for the next trial u_{i+1}.

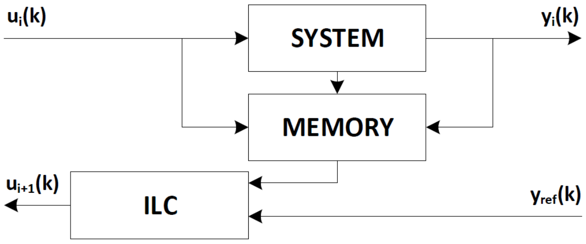


FIGURE 2.12: The idea of ILC (Ahn et al., 2007a)

2.3.1 Basic ILC Types

Three basic ILC contral laws were proposed in Arimoto et al. (1985). The first has a structure

$$u_{i+1}(k) = u_i(k) + \mu e_i(k+1) \tag{2.6}$$

where μ is the controller gain. This is a basic ILC configuration and it is called a derivative type (D-type) ILC due to the fact that it is non-causal with respect to time (contains the $e_i(k+1)$ term which is an equivalent of the error's derivative $\frac{d}{dt}e_i(t)$ in continuous-time version). However, as mentioned before, it can be implemented as the data from the past trials is stored in memory so the future time data from the past trials can be used. The second law

$$u_{i+1}(k) = u_i(k) + \mu e_i(k) \tag{2.7}$$

is called a proportional-type (P-type) ILC as the input update does not depend on future time data. The third option is the mixed-type design (PD-type)

$$u_{i+1}(k) = u_i(k) + \mu_1 e_i(k+1) + \mu_2 e_i(k) \tag{2.8}$$

or a PID-type if it includes also a term containing the error's integral.

Another basic ILC controller is called a phase-lead ILC and has a form

$$u_{i+1}(k) = u_i(k) + \mu e_i(k+\delta) \tag{2.9}$$

where $\delta \in \mathbb{Z}$. In such controller even further time instance data is used. Moreover, ILC can use only most recent previous trial data (first-order) or data from multiple previous trials (higher-order, HOILC) (Ahn et al., 2007b).

2.3.2 Model-based ILC

Basic ILC structures can be designed without using the information about the system's model, referred to as model-free ILC. Designing more sophisticated ILC structures requires knowing the system's model and it is described in this subsection. Model-based ILC can be designed using various system representations, including time-domain and frequency-domain analysis (Bristow et al., 2006).

2.3.2.1 Gradient and Norm Optimal ILC

One of the prominent ILC approaches that can provide the fast error convergence of the system are gradient-based methods, i.e. gradient ILC and norm optimal ILC (NOILC). These methods use time-domain analysis using a so called lifted-system framework (Bristow et al., 2006) where the input-output relation is given by the equation (2.12) and input, output and error signals can be written in the form of supervectors.

The following state-space model is considered (Rogers et al., 2010; Dinh et al., 2014)

$$\begin{aligned} \mathbf{x}(k+1) &= A\mathbf{x}(k) + B\mathbf{u}(k), \quad x(0) = x_0, \quad 0 \leq k \leq \alpha \\ \mathbf{y}(k) &= C\mathbf{x}(k) \end{aligned} \tag{2.10}$$

where α is the number of samples, $\mathbf{x}(k)$ is the $n \times 1$ vector, $\mathbf{u}(k)$ is the $l \times 1$ vector, $\mathbf{y}(k)$ is the $m \times 1$ vector and A, B, and C are the system matrices of corresponding dimensions. As α is finite the input and the output can be written in the form of supervectors

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}(1) \\ \mathbf{y}(2) \\ \vdots \\ \mathbf{y}(\alpha) \end{bmatrix} \quad \mathbf{u} = \begin{bmatrix} \mathbf{u}(0) \\ \mathbf{u}(1) \\ \vdots \\ \mathbf{u}(\alpha-1) \end{bmatrix} \tag{2.11}$$

and the input to output relation of the linear system is given by

$$\mathbf{y} = G\mathbf{u} + \mathbf{y}_0 \tag{2.12}$$

where the G matrix is given by

$$G = \begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ CA^2B & CAB & CB & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{\alpha-1}B & CA^{\alpha-2}B & \dots & \dots & CB \end{bmatrix} \tag{2.13}$$

and y_0 denotes the response to initial conditions given by

$$\mathbf{y}_0 = [(CA)^T(CA^2)^T...(CA^\alpha)^T]^T\mathbf{x}_0 \tag{2.14}$$

which can be absorbed to the reference trajectory $\mathbf{y}_{ref}$. Note that if $CB = 0$ then the requirement of ILC trial-to-trial error convergence does not hold and further analysis is required, see, for example, Hladowski et al. (2011) and Paszke et al. (2016). Moreover, a matrix with a large condition number (the ratio between the maximum and minimum singular value) is said to be ill-conditioned and can indicate instability.

The purpose of gradient methods is to minimize a particular quadratic cost function so that the error convergence through iteration is provided. Typically, the cost function at trial $i+1$ has the form (Rogers et al., 2010; Dinh et al., 2014)

$$J_{i+1}(\mathbf{u}_{i+1}) = ||\mathbf{e}_{i+1}||^2_{\mathcal{Y}} + ||\mathbf{u}_{i+1} - \mathbf{u}_i||^2_{\mathcal{U}} \tag{2.15}$$

where $\mathcal{U}$ and $\mathcal{Y}$ are the input and output real Hilbert spaces with inner products $\langle \mathbf{u}_1, \mathbf{u}_2 \rangle_{\mathcal{U}} = \mathbf{u}_1^T R \mathbf{u}_2$ and $\langle \mathbf{y}_1, \mathbf{y}_2 \rangle_{\mathcal{Y}} = \mathbf{y}_1^T Q \mathbf{y}_2$ such that $||\mathbf{u}||^2_{\mathcal{U}} = \langle \mathbf{u}, \mathbf{u} \rangle_{\mathcal{U}} = \mathbf{u}^T R \mathbf{u}$,

$||\mathbf{e}||^2_{\mathcal{Y}} = \langle \mathbf{e}, \mathbf{e} \rangle_{\mathcal{Y}} = \mathbf{e}^T Q \mathbf{e}$. R and Q are symmetric positive semi-definite weighting matrices adjusting the balance between the system robustness and the rate of error convergence respectively.

In norm optimal ILC the input on trial $i+1$ is calculated as the solution of the minimum norm optimisation problem

$$\arg\min_{u_{i+1}} J_{i+1}(\mathbf{u}_{i+1}) \tag{2.16}$$

where the minimum solution for u_{i+1} can be obtained from the stationary condition by Frèchet differentiation of (2.15) with respect to u_{i+1} giving (Rogers et al., 2010)

$$\mathbf{u}_{i+1} = \mathbf{u}_i + G^* \mathbf{e}_{i+1} \tag{2.17}$$

where $G^* : \mathcal{Y}^* \rightarrow \mathcal{U}^*$ is the adjoint operator of the input-output relation matrix $G : \mathcal{U} \rightarrow \mathcal{Y}$ with respect to the weighted inner product equations given by

$$G^* = R^{-1} G^T Q \tag{2.18}$$

Using $\mathbf{e} = \mathbf{y}_{ref} - G\mathbf{u}$ the error update relation becomes

$$\mathbf{e}_{i+1} = (I + GG^*)^{-1} \mathbf{e}_i \tag{2.19}$$

and substituting (2.19) to (2.17) we obtain

$$\mathbf{u}_{i+1} = \mathbf{u}_i + G^*(I + GG^*)^{-1} \mathbf{e}_i \tag{2.20}$$

The control design problem is to construct an input sequence $\{\mathbf{u}_i\}_{i \geq 0}$ that when applied forces the error sequence $\{\mathbf{e}_i\}_{i \geq 0}$ to converge in i, i.e.,

$$\lim_{i \rightarrow \infty} ||\mathbf{e}_i|| = 0, \; \lim_{i \rightarrow \infty} ||\mathbf{u}_i - \mathbf{u}_\infty|| = 0 \tag{2.21}$$

where $||\cdot||$ is a signal norm in a suitably chosen function space with a norm-based topology and $\mathbf{u}_\infty$ is termed the learned control in some of the literature. For performance it is also necessary to ensure that the dynamics produced along the trials (in k) are also acceptable, where this issue is application dependent. In the case of norm optimal ILC the weighting matrices G and R set the balance between the speed of the error convergence against the input signal strength (Rogers et al., 2007).

In gradient ILC the gradient decent optimisation method is used to minimise the cost function (2.15) along the direction of the gradient of the error norm by setting the input update to

$$\mathbf{u}_{i+1} = \mathbf{u}_i + \beta G^* \mathbf{e}_i \tag{2.22}$$

where β is a positive scalar representing the step size to be chosen at each trial and can be calculated using (Rogers et al., 2010; Dinh et al., 2014)

$$\beta = \frac{\mathbf{e}_i^T QGG^* \mathbf{e}_i}{||GG^* \mathbf{e}_i||^2_{\mathcal{Y}} + ||G^* \mathbf{e}_i||^2_{\mathcal{U}}} \tag{2.23}$$

One of the advantages of using the lifted system framework is that it can be extended to multivariable, time-varying and non-linear systems (Bristow et al., 2006).

2.3.2.2 Frequency Domain Approach

ILC scheme can be designed using frequency domain analysis with the z-domain representation (Bristow et al., 2006). Z-transform requires signals defined over infinite time horizon and hence in ILC applications where finite trial length is used it is considered an approximation of the ILC system. In this case the model in z-transform domain is used where the discrete transfer function of the system's plant $P(z)$ is

$$P(z) = \frac{N(z)}{D(z)} \tag{2.24}$$

where z is the z-transform variable, the roots of the numerator $N(z)$ are called system's zeros and the roots of the denominator $D(z)$ are called system's poles.

One of possible configurations of ILC scheme is given in Fig. 2.13.

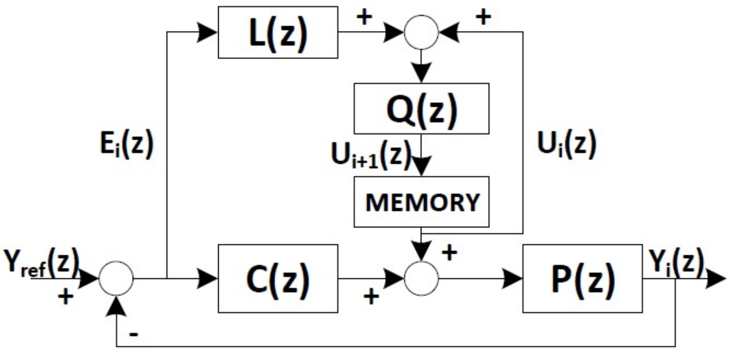


FIGURE 2.13: Two stage ILC design in frequency domain

This is a two stage design where the feedback controller $C(z)$ is usually designed as a PID controller at first. Subsequently, the feedforward ILC scheme is added, where $L(z)$ and $Q(z)$ are the learning and robustness filters respectively. From the diagram the error on current trial can be calculated as

$$E_i(z) = Y_{ref}(z)\frac{1}{1+P(z)C(z)} - U_i(z)\frac{P(z)}{1+P(z)C(z)} \tag{2.25}$$

where $\frac{P(z)}{1+P(z)C(z)}$ is referred to as the process sensitivity $S_p(z)$ and $U(z)$, $Y(z)$, $E(z)$ denote z-transform of the input, output and error signals respectively.

Let's assume now that $Y_{ref}(z) = 0$ like in Steinbuch and van de Molengraft (2000), then

$$E_i(z) = -U_i(z)S_p(z); \quad E_{i+1}(z) = -U_{i+1}(z)S_p(z) \tag{2.26}$$

The learning update is:

$$U_{i+1}(z) = Q(z)[U_i(z) + zL(z)E_i(z)] \tag{2.27}$$

where z multiplying $L(z)$ emphasizes the forward time shift used in learning (Bristow et al., 2006).

Using (2.26) and (2.27) the error in the next trial can be calculated as

$$E_{i+1}(z) = Q(z)[1 - zL(z)S_p(z)]E_i(z) \tag{2.28}$$

and the monotonic convergence will take place if the frequency response of the z-domain system given by setting $z = e^{j\omega}$ for frequency ω in $[-\pi, \pi]$ meets the following stability condition (Bristow et al., 2006)

$$\sup_{\omega=[-\pi,\pi]} \left|Q(z)[1 - zL(z)S_p(z)]\right| < 1 \tag{2.29}$$

Summarising, the most obvious choice for the learning filter is an exact inverse of the process sensitivity which would bring the error to zero in one step (dead-beat). In practice, it is difficult to find an exact inverse due to two issues that arise while inversing the process sensitivity. Firstly, the denominator of $S_p(z)$ is usually of a higher order than the numerator and hence the inverse is non-causal. This is not an issue in ILC as its specific properties enable practical implementation of non-causal learning filters as explained in section 2.3 (the full time sequence of data is available from previous trials). Secondly, unstable zeros of $S_p(z)$ map to unstable poles of the filter $L(z)$, which is unacceptable. Thus, the learning filter is usually designed as the approximation (Steinbuch and van de Molengraft, 2000)

$$L(z) \approx S_p(z)^{-1} \tag{2.30}$$

which can be found using a stable inversion approach, e.g. Zero Phase Error Tracking Algorithm for Digital Control (ZPETC) proposed by Tomizuka (1987), where the stable zeros of $S_p(z)$ are inverted and the unstable zeros are used to scale the steady-state gain. Finally, the $Q(z)$ filter is designed as a low-pass filter (e.g. Butterworth) to counteract for the mismatch between the filter $L(z)$ and the model's dynamics that can appear in high frequencies and hence make sure that the convergence condition (2.29) is satisfied for all frequencies. What is more, forward-backward method is used for filtering to make sure that $Q(z)$ doesn't affect the phase of the input signal.

2.3.2.3 Two-dimensional Systems Theory Based ILC

An alternative system description that allows simultaneous consideration of trial-to-trial error convergence and transient response along the trial is to use the two-dimensional (2D) systems setting which is based on treating trial number i as one direction of information propagation and time step k as the other (Hladowski et al., 2008, 2016). In this approach the ILC design is based on the discrete linear repetitive process of a state-space form

$$\begin{aligned} x_{i+1}(k+1) &= \widehat{A}x_{i+1}(k) + \widehat{B}u_{i+1}(k) + \widehat{B}_0 y_i(k) \\ y_{i+1}(k) &= \widehat{C}x_{i+1}(k) + \widehat{D}u_{i+1}(k) + \widehat{D}_0 y_i(k) \end{aligned} \tag{2.31}$$

Let's consider a system in the standard state-space form

$$\begin{aligned} x_i(k+1) &= Ax_i(k) + Bu_i(k) \\ y_i(k) &= Cx_i(k) \end{aligned} \tag{2.32}$$

For 2D systems setting ILC the control input for the next trial is

$$u_{i+1}(k) = u_i(k) + \Delta u_{i+1}(k) \tag{2.33}$$

with

$$\Delta u_{i+1}(k) = K_1\eta_{i+1}(k+1) + K_2 e_i(k+1) \tag{2.34}$$

where $\eta_{i+1}(k+1) = x_{i+1}(k) - x_i(k)$ and K_1, K_2 are the controller gain matrices which can be found using optimization techniques (e.g. by solving Linear Matrix Inequalities, LMI). Then, the controlled ILC dynamics can be written as

$$\begin{aligned} \eta_{i+1}(k+1) &= \hat{A}\eta_{i+1}(k) + \hat{B}_0 e_i(k) \\ e_{i+1}(k) &= \hat{C}\eta_{i+1}(k) + \hat{D}_0 e_i(k) \end{aligned} \tag{2.35}$$

where $\hat{A} = A + BK_1$, $\hat{B}_0 = BK_2$, $\hat{C} = -C(A + BK_1)$, and $\hat{D}_0 = I - CBK_2$.

The equation (2.35) is a particular case of the linear repetitive process (2.31) with trial state vector η, trial profile vector e and zero current trial input. Such controller is static and in some cases the use of LMI techniques to find the gains can be problematic and lead to poorly conditioned solution. This problem can be solved by implementing controllers with their own dynamics (Hladowski et al., 2016)

$$\eta^c_{i+1}(k+1) = A_c\eta^c_{i+1}(k) + B_{0c}e_i(k) \tag{2.36}$$

where $\eta^c_{i+1}(k) = x^c_{i+1}(k-1) - x^c_i(k-1)$ is a trial-to-trial increment of the controller's internal state vector $x^c_i(k) \in \mathbb{R}^{n_c}$ with $n \neq n_c$. Then, the dynamic control law is

$$\Delta u_{i+1}(k) = C_c \eta^c_{i+1}(k+1) + E_c \eta_{i+1}(k+1) + D_c e_i(k+1) \tag{2.37}$$

The detailed methodology of finding optimal gains and stability analysis for both, static and dynamic, laws are given in Hladowski et al. (2016). The same authors proposed the $2D$ systems theory based ILC for linear time-varying discrete systems in Hladowski et al. (2007).

2.3.2.4 Other Model-based ILC Schemes

There is a number of other model-based ILC methods:

- Plant Inversion Methods – use the inverted plant dynamics as the learning function $u_{i+1}(k) = u_i(k) + G^{-1}e_i(k)$ (Bristow et al., 2006). The main drawback of this method is that the inversion may lead to unstable filter and it can be avoided using a stable inversion approach (Kinosita et al., 2002);
- Robust ILC - use H_∞ methods to synthesise controllers providing robust performance, e.g. synthesis of an H_∞ suboptimal controller in de Roover (1996);
- Constrained ILC - take system's constraints into account, e.g. work of Chu and Owens (2010);
- Point-to-Point ILC - consider only a subset of critical tracking points, e.g. work of Chen (2017).

2.3.3 Issues in ILC

In order to evaluate the performance of the algorithm some criteria on how to measure the performance must be chosen. In Rogers et al. (2010) three aspects that should be considered are given:

1. minimum tracking error (subject to chosen error norm),
2. time to reach the minimum error (convergence speed),
3. long-term performance (any signs of performance degradation over time).

In order to calculate the tracking error at each trial and the convergence speed some error norm must be chosen, e.g. mean square error (MSE). Moreover, the long-term performance must also be taken into consideration since one of the common problems that can arise in ILC is degradation of performance over time. The example of the error norm obtained for a control system with long-term performance degradation is shown in Fig. 2.14 (Rogers et al., 2010).

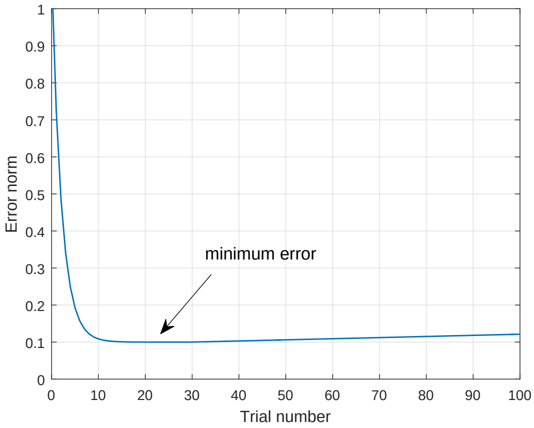


Figure 2.14: Error norm curve for an ILC system with performance degradation

It can be observed that initially the error norm decreases, but over larger number of trials it starts to grow again. This behaviour of ILC systems is still not completely understood (Rogers et al., 2010), however possible reasons have been analysed in the literature. For instance, in Longman (2000) the growth or decay with repetitions for the error components at different frequencies are examined to understand the total error growth phenomenon. The analysis shows that, in practice, a zero phase lowpass filter might be required to stop the learning process for high frequencies. Summarising, when testing the designed ILC it is important to not only look at the initial error norm convergence rate but also run a long simulation to fully assess the quality of the controller.

The other issue arising while implementing ILC described in Cai (2009) is resetting the system to initial conditions. In practice, this is a very hard task and therefore it is usual to set up a region around the ideal position where resetting is assumed to be fulfilled rather than resetting it to an exact initial position. The effects of initial condition errors have been investigated by many researchers including Lee and Bien (1996) who proposes a method that is robust against varying initial conditions.

2.4 Summary

The literature review on load control of wind turbines with smart rotors is given in this chapter. Such control involves modification of the blade section aerodynamics by providing the rotor blade with sensors, AFC devices and a suitable control algorithm. Both, AFC devices and possible control approaches have been described together with a general wind turbine load control concept and motivation. Moreover, the background on the ILC algorithm is given together with the classification of ILC algorithms and description of the problems that can arise during implementation.

Wind energy is currently the fastest growing branch of energy industry and the research on improved control schemes that can provide the satisfactory aerodynamic load performance of the system is highly demanded. Load management is a crucial issue in keeping wind energy economically competitive and it becomes even more important as the rotor size is still increasing and the offshore wind structures are getting more significant. Accessing such structures for repair is expensive and an efficient load management could lead to improved reliability and extended lifetime of the components. These are the main motivating factors for this research.

ILC algorithms are able to provide a good along the trial performance together with trial-to-trial error convergence for systems operating repetitively. This cannot be achieved with majority of other control approaches. A number of control algorithms suitable for wind turbines with smart rotors have been investigated recently, including simple model-free ILC schemes, but the true potency of ILC in this branch is yet to be investigated.

This thesis develops substantial new results on ILC for wind turbine applications with particular emphasis on load management by smart rotor blades. The results include further development of simple structure laws whose coefficients are chosen by tuning methods (akin to auto-tuning for three term controllers) and a major extension to model based design based on a finite dimensional approximate model of the dynamics constructed by the use of a Proper Orthogonal Decomposition. All designs are supported by the results of applying them to a CFD based model of the underlying dynamics.

Chapter 3

Model-free ILC for Load Management in Wind Turbines

In this chapter the implementation of a model-free ILC design for load control of the wind turbines with smart rotors blade is described and the simulation results are shown. A CFD model based on a panel method is used to simulate the flow past an airfoil. The airfoil is generated using Kàrmàn-Trefftz transform (Burington, 1940).

3.1 Flow Model

In this section a CFD panel code is used to simulate the flow past an airfoil. Such a representation is adequate for preliminary investigation of the control algorithms which could be used, when fully developed, in future experiments on full scale turbines and is less expensive computationally than full Navier-Stokes simulation. The key assumptions of this approach can be summarised as:

1. The problem is considered in a non-dimensional form.
2. The airfoil profile is generated using Kàrmàn-Trefftz transform.
3. The flow over a $2D$ airfoil is simulated and the boundary conditions at the body are satisfied using the panel method.
4. The flow is assumed to be inviscid and extreme cases when separation is provoked are not considered.
5. The motion of the vortices is found by solving the Euler equations (a numerical solution can be found using any time-stepping method, e.g. Runge-Kutta methods).

6. The wake effect is simulated by releasing vortices from the trailing edge at each time step.

7. The lift is calculated from the pressure distribution using the unsteady Bernoulli's equation.

8. The AFC devices are modelled in a generic manner by altering the strength of the new vortex generated at the trailing edge at each time step.

The detailed description of these assumptions is given in the following subsections.

3.1.1 Non-dimensional Data

The velocity and the chord length of the blade section vary along the blade and therefore the time t^* for the flow to pass the blade section also changes. Here, this time is considered as non-dimensional using the chord length c^* and the mean free stream velocity V_∞^* as reference according to

$$t^* = \frac{c^*}{V_\infty^*} t \tag{3.1}$$

where the asterisk * denotes a dimensional quantity. The pressure p^* is non-dimensionalised using

$$p^* = \rho^* (V_\infty^*)^2 p \tag{3.2}$$

where ρ^* denotes the air density. The lift per unit span L^* is non-dimensionalised using

$$L^* = \rho^* (V_\infty^*)^2 c^* L \tag{3.3}$$

The non-dimensionalised quantities will be used in the remainder of this work.

3.1.2 Airfoil Profile

The airfoil used in this research is generated using conformal transform. Such profile is not one of the standard airfoil types used in wind turbines, but it has a similar shape and properties to the standard profiles (e.g. NREL S825) and it significantly simplifies some of the calculations in the following part of this research (section 4.2), where the properties of the regular grid around this airfoil are used to calculate the velocity gradients on the grid.

A wide range of airfoils can be generated by applying Joukowski transform (Burington, 1940) to a circle in ζ space according to

$$z = \zeta + \frac{1}{\zeta} \tag{3.4}$$

where $z = x + iy$ is a complex variable in the new space (airfoil profile) and $\zeta = X + iY$ is a variable in original space (circle). Varying the coordinates of the centre of the circle results in different shapes of the resulting airfoil. The airfoils obtained using Joukowski transform have a cusp at the trailing edge. Kàrmàn-Trefftz transform is related to Joukowski transform and it enables the generation of airfoils with non-zero trailing edge angle α. The conformal transformation (Burington, 1940)

$$\frac{z-n}{z+n} = \left(\frac{\zeta-1}{\zeta+1}\right)^n \tag{3.5}$$

with n set to slightly smaller than 2 (e.g. 1.9) maps the ζ-plane circle onto the Kàrmàn-Trefftz airfoil profile in z-plane. The trailing edge angle α is related to n by

$$n = 2 - \frac{\alpha}{\pi} \tag{3.6}$$

Finally, the profile of the airfoil can be calculated using

$$z = n\frac{(1+\frac{1}{\zeta})^n + (1-\frac{1}{\zeta})^n}{(1+\frac{1}{\zeta})^n - (1-\frac{1}{\zeta})^n} \tag{3.7}$$

In this work the parameter $n = 1.9$ and the coordinates of the centre of the circle $x_0 = -0.05$ and $y_0 = 0.2$ are used to generate the profile. The resulting airfoil is given in Fig. 3.1. The profile is also normalized using the chord length.

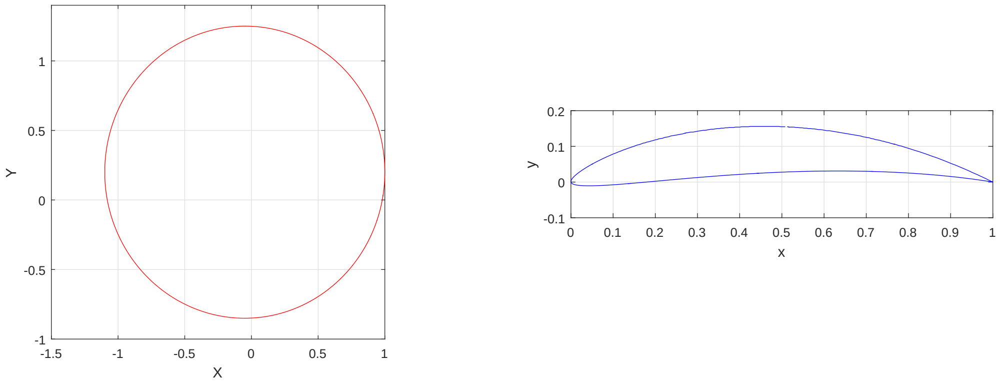


FIGURE 3.1: The circle and the resulting airfoil profile

3.1.3 Model Description

A panel method (Anderson, 2011) with N vortex and N source panels is used to satisfy the boundary condition of the zero normal velocity at the surface of the body. The vorticity γ_n is constant within each vortex panel n but varies between the panels. In this work 400 vortex panels are placed just above the surface. Similarly 400 source panels with strengths λ_n varying between the panels are placed just below the surface. The panels are clustered towards the trailing edge and their distribution is shown in Fig. 3.2 where the beginning and end of each panel is marked with a red circle.

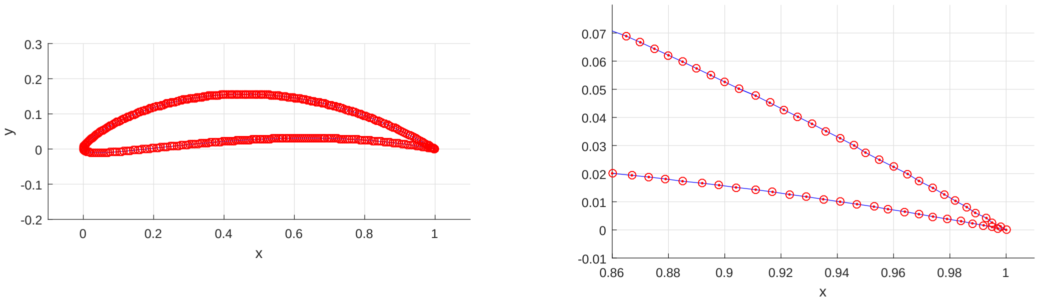


Figure 3.2: The distribution of the panels: entire airfoil (left), trailing edge (right)

The panels produce a system that consists of $2N$ equations and $2N$ unknowns - γ_n and λ_n. The purpose of the panel technique is to solve for γ_n and λ_n such that the surface of the body becomes a streamline. This problem is not well posed (no unique solution exists) with the solution varying with the lift on the body. A further condition must be imposed which results in an overdetermined system. In particular, the additional condition resulting from the Kelvin's circulation theorem is used to make the system overdetermined so the solution can be obtained using least-square approximation. This theorem states that the circulation around a closed curve moving with the fluid remains constant with time. Adding this additional condition results in overdetermined system of $2N+1$ equations and the solution can be approximated using the least-squares method as in Clarke and Tutty (1994).

The flow consists of the free-stream component $\mathbf{V}_0$, the velocity field which is generated by N vortex panels $\mathbf{v}_p(\mathbf{x},t) = (v_{px}, v_{py})$ and discrete vortices that are shed into the wake. The governing equation for a $2D$ inviscid incompressible fluid is

$$\frac{D\omega}{Dt} = \frac{\partial \omega}{\partial t} + v_x \frac{\partial \omega}{\partial x} + v_y \frac{\partial \omega}{\partial y} = 0 \tag{3.8}$$

where $D/Dt = \partial/\partial t + v_x \partial/\partial x + v_y \partial/\partial y$ denotes the material derivative and $\omega = \partial v_y/\partial x - \partial v_x/\partial y$ denotes the vorticity. This equation states that a collection of discrete vortices is convected with the flow and every individual vortex moves according to

$$\frac{d\mathbf{x}_v}{dt} = \mathbf{v}(\mathbf{x}_v, t) \tag{3.9}$$

where $\mathbf{x}_v$ is the vortex position. Moreover, the velocity field of an individual vortex $\mathbf{v}_{vj}$ is given by

$$\mathbf{v}_{vj} = \Gamma_j \frac{(-(y - y_{vj}), x - x_{vj})}{|\mathbf{x} - \mathbf{x}_{vj}|^2} F(|\mathbf{x} - \mathbf{x}_{vj}|) \tag{3.10}$$

where Γ_j denotes the strength of the vortex and

$$F(s) = \int_0^s \gamma(s) ds \tag{3.11}$$

where $\gamma(s)$ denotes the Gaussian distribution function.

Finally, the complete velocity field is given by

$$\mathbf{v}(\mathbf{x}, t) = \mathbf{V}_0 + \mathbf{v}_p + \mathbf{v}_v \tag{3.12}$$

where

$$\mathbf{v}_v = \sum_{j=1}^{M} \mathbf{v}_{vj} \tag{3.13}$$

Summarising at this stage, a non-linear system of equations is produced due to the interaction between the vortices and the body. A numerical solution to the problem can be obtained using second-order Runge-Kutta method as in Clarke and Tutty (1994)

$$\hat{\mathbf{x}}_{vj} = \mathbf{x}_{vj}(k) + \frac{1}{2}\Delta t \mathbf{v}\left(\mathbf{x}_{vj}(k), t_k\right) \tag{3.14}$$

$$\mathbf{x}_{vj}(k+1) = \mathbf{x}_{vj}(k) + \Delta t \mathbf{v}\left(\hat{\mathbf{x}}_{vj}, t_{k+\frac{1}{2}}\right) \tag{3.15}$$

where Δt denotes the time step used and $t_k = k\Delta t$.

To simulate the wake effect at each time step a new vortex with a strength

$$\Gamma_{nc} = -\frac{1}{2}\Delta t\left(v_a^2 - v_b^2\right) \tag{3.16}$$

is created downstream of the trailing edge as shown in the Fig. 3.3 (Cottet and Koumoutsakos, 2000).

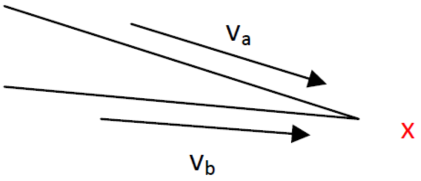


Figure 3.3: Vortex generation point (x)

Here v_a and v_b denote the velocity tangential to the upper and lower surfaces at the trailing edge.

The basic panel code has been validated by comparing it with the standard source Drela (1985) in the previous work of Blackwell (2015). Here, a similar model to the one in Blackwell (2015) is used with a few modifications. The model includes the wake generated from the trailing edge (vorticity is shed into the flow from the trailing edge) and the lift is calculated from the pressure distribution using the unsteady Bernoulli's equation.

3.1.4 Lift Estimation

The surface of the airfoil is a streamline with the velocity tangential to the surface denoted by u and the normal velocity equal to zero. Let s denote the distance along the streamline as shown in Fig. 3.4.

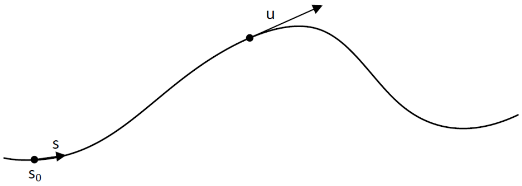


FIGURE 3.4: Streamline

The momentum equation for inviscid irrotational flow along a streamline is (Anderson, 2011)

$$\rho^* \frac{\partial u^*}{\partial t^*} + \rho^* u^* \frac{\partial u^*}{\partial s^*} = -\frac{\partial p^*}{\partial s^*} \tag{3.17}$$

where asterisk * denotes a dimensional quantity. Nondimensionalisation using $s^* = c^* s$, $u^* = V_\infty^* u$, $p^* = \rho^* (V_\infty^*)^2 p$ and $t^* = \frac{c^*}{V_\infty^*} t$ gives

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial s} = -\frac{\partial p}{\partial s} \tag{3.18}$$

or

$$\frac{\partial u}{\partial t} + \frac{1}{2} \frac{\partial u^2}{\partial s} = -\frac{\partial p}{\partial s} \tag{3.19}$$

Integrating along the streamline from $s{=}s_0$ with $u{=}u_0$ and $p{=}p_0$ gives

$$\int_0^s \frac{\partial u}{\partial t} ds + \frac{1}{2} \left(u_s^2 - u_0^2 \right) = -(p_s - p_0) \tag{3.20}$$

Since p_0 is an arbitrary function of t, we can choose $p_0 = -\frac{1}{2} u_0^2$ so that

$$p_s = -\left[\int_0^s \frac{\partial u}{\partial t} ds + \frac{1}{2} u_s^2 \right] \tag{3.21}$$

Now consider a panel n running from s_n to s_{n+1} as in Fig. 3.5, where

$$\Delta x = x_{n+1} - x_n \tag{3.22}$$

$$\Delta y = y_{n+1} - y_n \tag{3.23}$$

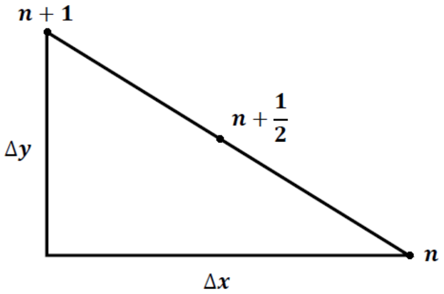


Figure 3.5: A panel

$$l_n = \left[\Delta x^2 + \Delta y^2\right]^{\frac{1}{2}} \tag{3.24}$$

$$\mathbf{t} = \frac{(\Delta x, \Delta y)}{l_n} \tag{3.25}$$

$$\mathbf{n} = \frac{(\Delta y, -\Delta x)}{l_n} \tag{3.26}$$

The panel code calculates the velocities at the centre of the panel so that the velocity along the streamline at $s_{n+\frac{1}{2}}$ is given by

$$u_{n+\frac{1}{2}} = \mathbf{u}_{n+\frac{1}{2}} \cdot \mathbf{t} \tag{3.27}$$

where $u_{n+\frac{1}{2}}$ is the velocity along the streamline and $\mathbf{u}_{n+\frac{1}{2}}$ denotes the velocity vector with components in $\binom{x}{y}$ at $s_{n+\frac{1}{2}}$.

The contribution to the pressure from the integral along panel n is

$$\Delta p_{n+\frac{1}{2}} = \frac{\partial}{\partial t}\int_{s_n}^{s_{n+1}} u ds = \frac{u^k_{n+\frac{1}{2}} - u^{k-1}_{n+\frac{1}{2}}}{\Delta t} l_n \tag{3.28}$$

where k denotes the discrete step ($t = k\Delta t$).

The pressure at the midpoint of the panel is given by

$$p_{\frac{1}{2}} = -\left[\Delta p_{\frac{1}{2}} + \frac{1}{2}u^2_{\frac{1}{2}}\right], \quad n = 0 \tag{3.29}$$

$$p_{n+\frac{1}{2}} = -\left[\Delta p_{n+\frac{1}{2}} + \sum_{i=1}^{n}\frac{1}{2}(\Delta p_{i+\frac{1}{2}} + \Delta p_{i-\frac{1}{2}}) + \frac{1}{2}u^2_{n+\frac{1}{2}}\right], \quad n \geq 1 \tag{3.30}$$

Finally the lift on each panel is equal to the y component of the pressure

$$\delta L_n = p_{n+\frac{1}{2}}\Delta x = p_{n+\frac{1}{2}} l_n t_x \tag{3.31}$$

3.1.5 Modelling Smart Rotors

The smart devices are modelled by modifying the circulation generated on the trailing edge. In the no control case this vortex is described by the equation (3.16). In the controlled case, at every time step a new vortex generated from the trailing edge will have a strength

$$\Gamma_c = u \tag{3.32}$$

where u denotes the control input. To assure that the control action is turned on smoothly a blend of two actions is used during approximately first two cycles since switching the control on according to

$$\Gamma = v_f \Gamma_{nc} + (1 - v_f)\Gamma_c \tag{3.33}$$

where v_f is a function of time slowly decreasing its value from 1 to 0, whereas Γ_{nc} and Γ_c are the vortices generated in no control and controlled cases given by (3.16) and (3.32) respectively.

Altering the circulation on the trailing edge modifies the lift and represents devices such as microtabs or plasma actuators which also act by generating vortices or changing the flow on the trailing edge. This is a generic approach to model smart rotors with no specific device chosen. Choosing a specific device would involve detailed modelling of its influence on the local flow and hence the lift and it is a topic for further research. However, our approach is probably closest to the situation with plasma actuators as they involve no moving parts and have a very fast response.

3.2 ILC Based Load Reduction

The CFD model described in the previous section is used to examine the ILC for load control in wind turbines with smart rotors. In this part of the research a model-free ILC is designed where the controller's gain is found akin to auto-tuning design without using any information of system's dynamics.

3.2.1 Iterative Learning Control for Load Control

In this work the effect of the rotation is modelled by assuming that the freestream velocity is periodic as described by equation (3.34). This is a simplified approach assumming the constant angle of attack (AoA) of 0°. In practice, the AoA would vary as well as the blade rotates. This would add a further cyclic element to the system that would not change the basic control approach based on ILC, but should be taken into account in the equations modelling the motion of the vortices in future analysis.

The flow far upstream the airfoil is

$$V_{0x}(k) = 1 + A\,sin(\frac{2\pi k\Delta t}{T}) \tag{3.34}$$

where A denotes the amplitude of the oscillation and T denotes the period of turbine's rotation. The discrete version of the signals is used in the remainder of this chapter with $k = 0, 1, ..., \alpha - 1$ denoting the step within a cycle and $\alpha = T/\Delta t$ denoting the number of steps in one cycle, where Δt is the time step.

Remark 3.1. *In this work a constant period and a fixed time step are assumed in the control computation. However, variable periods of rotation can be considered by using a constant increment in angle of the blade and varying the time step accordingly. Previous work by Rogers and Tutty (2016) has shown this produces very similar control behaviour.*

The lift obtained for such flow will be periodic and the control objective can be defined as rejecting periodic disturbances by keeping the lift constant. This can be achieved by altering the lift on the rotor blades such that the error between the lift and the desired (constant) value for the lift is minimal. The error at step k is given by

$$e(k) = L_{ref}(k) - L(k) \tag{3.35}$$

where the desired value of the lift L_{ref} can be specified by setting $A = 0$ in equation (3.34) and the lift at step k can be estimated from pressure sensor readings which can be extracted from the flow model. The lift is a non-dimensional quantity as described in subsection 3.1.1.

In Fig. 3.6 the lift obtained for simulating the flow given by (3.34) with $A = 0.1$, $T = 2.5$ and $\Delta t = 0.005$ is shown together with the desired value of the lift and the error. For the regular wind turbine operation the period T is much larger than $T = 2.5$ (at least one order of magnitude larger). In this chapter a smaller value is used since a single CFD simulation is expensive computationally and multiple simulations have to be performed. However, a smaller value of the period represents a more extreme case of a quickly rotating turbine which is more challenging for the control scheme. Therefore, it is assumed that if a good control performance can be achieved for this more extreme case of a shorter period then increasing the rotation period should represent a simpler case from the controller's perspective. Larger period values will be used in the following chapters of this thesis. Similarly, the amplitude of oscillation comming from the change in wind speed would also be lower than 10% for a medium-sized offshore wind turbine. However, this variation depends on many factors, including the hub height and terrain type, and 10% is chosen as the test value to represent a significant variation in freestream.

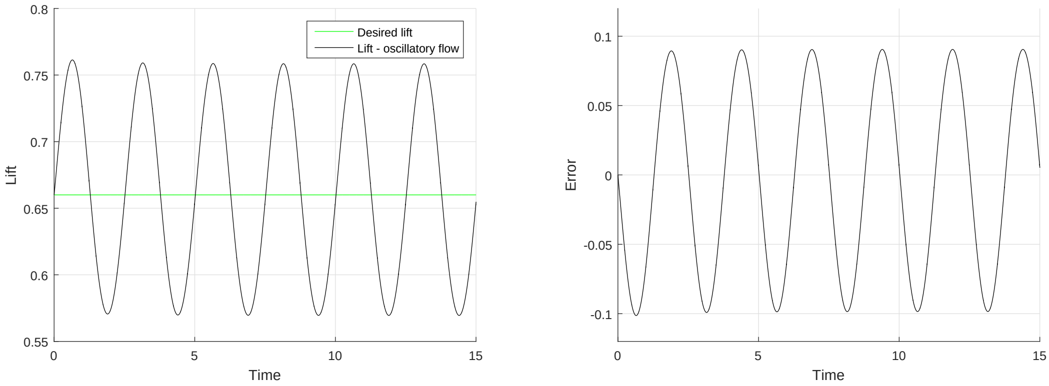


FIGURE 3.6: Lift (left) and error (right) obtained for the oscillatory flow of Eq. (3.34) - no control case

To reduce the lift fluctuations a phase-lead ILC controller is implemented at first where the input update at trial i is given by

$$\hat{u}_i(k) = u_{i-1}(k) + \mu_1 \Delta t e_{i-1}(k+\delta) \tag{3.36}$$

where μ_1 denotes the controller's gain and δ is the phase shift.

The control is switched on at $t = 3T$ as the CFD simulation requires approximately two cycles of operation to develop the wake. For first tests the phase shift is set to $\delta = 0$ and various values of the controller's gain μ_1 are examined. For the controller with the gain $\mu_1 = 1$ initially the error decreases slightly but then grows after approximately 10 trials as it can be seen in Fig. 3.7 (magenta). For any higher gain the error rapidly becomes unbounded in several trials. However, choosing a smaller value of the gain leads to a reduction in oscillations (blue line in Fig. 3.7).

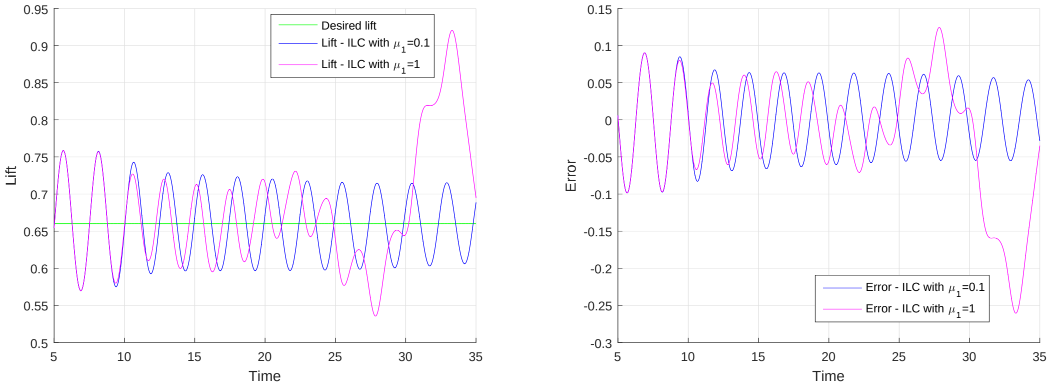


FIGURE 3.7: Lift (left) and error (right) obtained for the system with the ILC controller of Eq. (3.36)

Two norms are used to measure the performance: 2-norm and ∞-norm can be calculated for each trial i according to (3.37) and (3.38) respectively

$$\mathcal{L}_2 = \sqrt{\frac{1}{\alpha} \cdot \sum_{k=1}^{\alpha} (e(k))^2} \tag{3.37}$$

$$\mathcal{L}_\infty = max|e(k)| \tag{3.38}$$

For the controller with $\mu_1 = 0.1$ the 2-norm $\mathcal{L}_2 = 4.7 \cdot 10^{-2}$ is obtained after 10 trials compared to $\mathcal{L}_2 = 6.7 \cdot 10^{-2}$ for the no control case. The ∞-norm is $\mathcal{L}_\infty = 6.7 \cdot 10^{-2}$ and $\mathcal{L}_\infty = 9.8 \cdot 10^{-2}$ respectively so the reduction of 30% for both norms is achieved. Choosing even smaller value of the gain does not improve the performance (very similar results are obtained for $\mu_1 = 0.01$) and the same for choosing different phase shift values δ. Therefore, the further reduction in the error cannot be obtained with this design.

Significantly improved results can be obtained using the combination of two controllers: ILC and a proportional controller (P). For such system the input update is

$$u_i(k_t) = \hat{u}_i(k) + u(k_t) \tag{3.39}$$

where: $k_t = i\alpha + k$ is the total number of steps, $\hat{u}_i(k)$ is the ILC update and $u(k_t)$ and is the proportional controller update given by

$$u(k_t) = \mu_0 \Delta t e(k_t - 1) \tag{3.40}$$

where μ_0 is the P controller's gain.

The proportional controller provides a good error convergence on its own as can be seen in Fig. 3.8. After just 5 trials the error is significantly reduced for controllers with $\mu_0 = 20$ and $\mu_0 = 50$ (over 90% reduction in ∞-norm). However, increasing the gain to $\mu_0 = 60$ makes the system unstable so higher gains cannot be used and the further convergence cannot be achieved.

The combination of two controllers P and ILC provides the best convergence. In such case the oscillations in lift can be almost completely damped and both error norms are reduced by two orders of magnitude in less than 5 trials. This is shown in Fig. 3.9 (right) in logarithmic scale where the controller with $\mu_0 = 50$ and $\mu_1 = 1$ is tested. After more trials basically perfect performance is achieved. However, one should notice here that this is a perfect case where no delay in actuation is assumed and no additional vortices are introduced to the flow. Results for such cases are evaluated in the remaining part of this chapter.

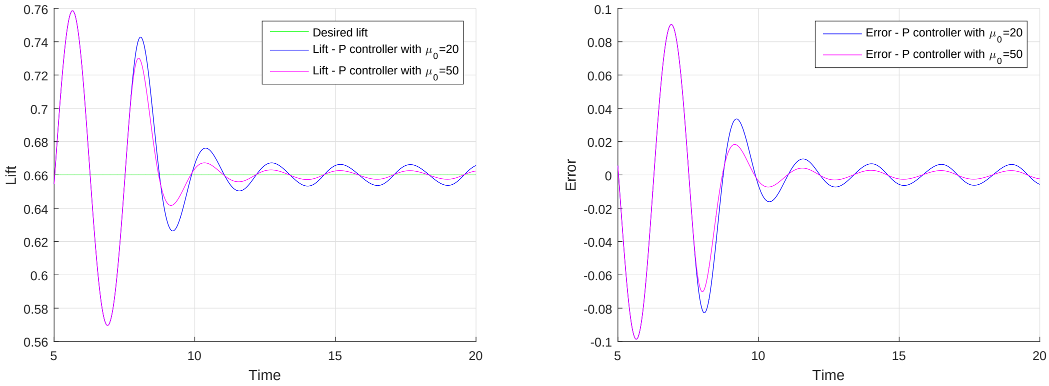


FIGURE 3.8: Lift (left) and error (right) obtained for the system with the P controller of Eq. (3.40)

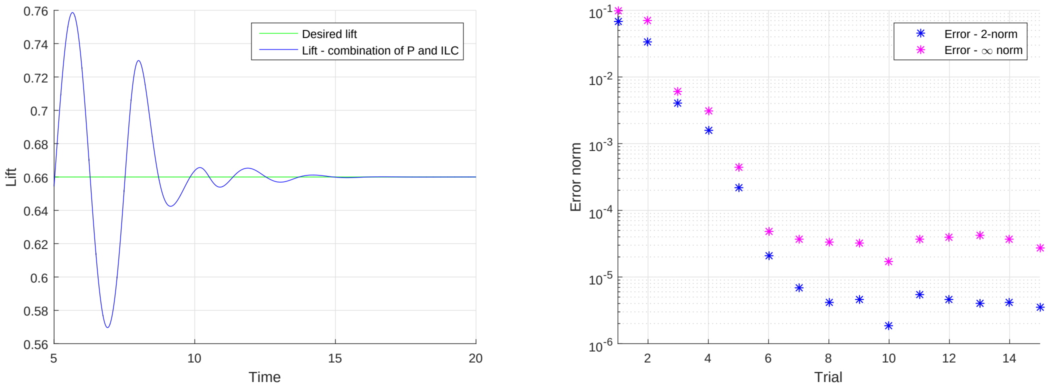


FIGURE 3.9: Lift (left) and error norm (right) obtained for the system controlled by the combination of P and ILC given by Eq. (3.39)

3.2.2 Actuator Dynamics Analysis

The behaviour obtained in the previous subsection is not realistic in many cases as the actuator's dynamics are not taken into account. In a more realistic case, the controller calculates the required input update but the actuator is not able to reach this required value immediately. In this work the lag in actuation is modelled using a first-order differential equation given by

$$\frac{d\hat{u}}{dt} = \lambda(u - \hat{u}) \tag{3.41}$$

where u denotes the input control signal calculated using (3.39), $\hat{u}$ is the current control signal, and λ is the coefficient that determines the speed of response. The control input signal is delayed as demonstrated in Fig. 3.10 where the delay of a step signal (left) and sinusoidal signal (right) are shown for different values of the delay coefficient λ from the range $0.1 : 100$.

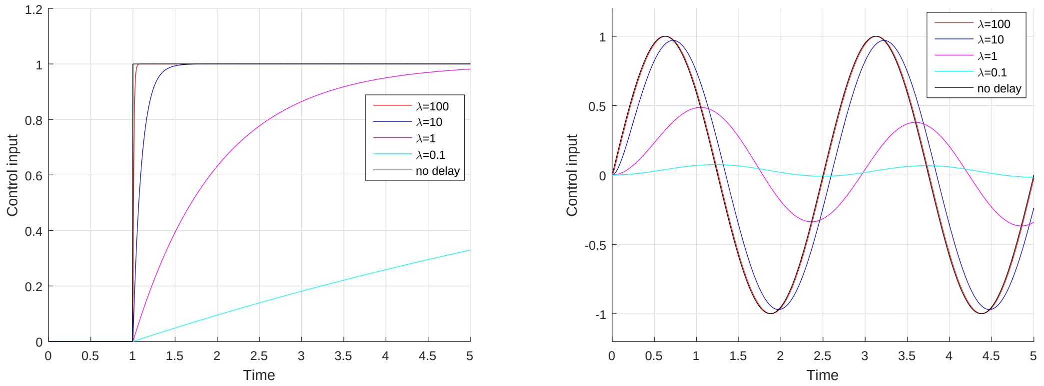


FIGURE 3.10: Effect of the actuation delay: step signal (left) and sinusoidal signal (right)

It can be observed that the smaller the λ value the bigger the delay with a very significant delay for $\lambda = 0.1$. In non-dimensional term the actuator will reach 90% of the desired input value in $t_{0.9} = \frac{2.3}{\lambda}$. For the non-dimensional period $T = 2.5$ and $\lambda = 10$ this time will be equal to $t_{0.9} = 0.23$ which corresponds to approximately 10% of the non-dimensional period T. As previously discussed this non-dimensional period will be much larger in practice, for example for $T = 200$ the coefficient $\lambda = 0.1$ will correspond to the actuator reaching 90% of the desired value in over 10% of the rotation period. In dimensional terms assumming rotation period of $4-5s$ the actuators would have to react in less than $0.5s$ to represent a delay of 10% of the actual period of rotation. This seems to be potentially achievable even with slower actuators ($\pm 10°$ maximum deflection with $10°/s$ deflection rate for a conventional trailing edge flaps in Berg et al. (2009)). The fast actuators using smart materials, e.g. plasma actuators, would correspond to much faster response coefficient λ responding within miliseconds.

The controller that demonstrated the best convergence in tests described in the previous subsection has been tested for this delayed system for various λ coefficients. Compared to the no control case a significant reduction of the error can be obtained for $\lambda = 100$ and $\lambda = 10$, but the oscillations are not completely damped like for the no delay case as shown in the Fig. 3.11. For $\lambda = 1$ the performance is acceptable but the remaining oscillation is higher than for $\lambda = 100$ and $\lambda = 10$. For extremely slow actuation $\lambda = 0.1$ the controller fails: convergence does not occur and the error grows compared to the no control case (not shown here). This is not surprising since this corresponds to a delay much higher than the rotation period.

Increasing the value of the gain μ_1 from $\mu_1 = 1$ to $\mu_1 = 10$ provides slightly better convergence, which indicates that increasing the ILC update gain could potentially produce further convergence (Fig. 3.12 (right)). Unfortunately, this is not the case and choosing higher gains, e.g. $\mu_1 = 30$ and $\mu_1 = 50$ introduces significant disturbances and decreases the performance of the system. This is shown in Fig. 3.12 (left). Conversely,

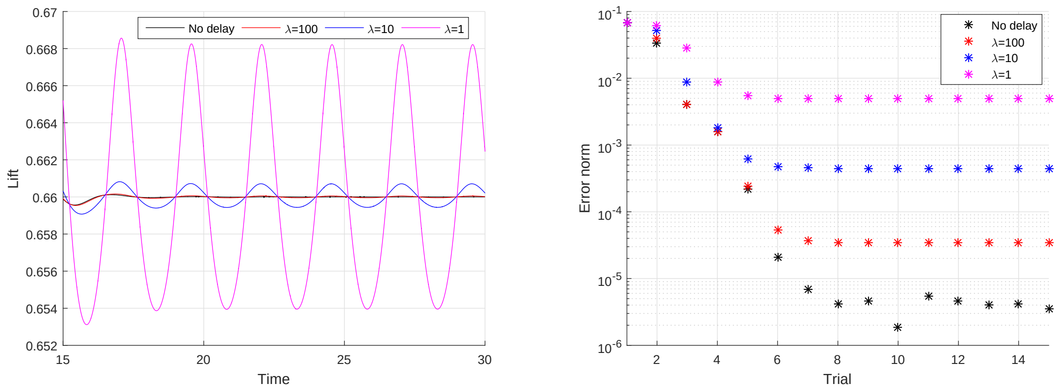


FIGURE 3.11: Performance of the combination of P and ILC for different actuator delays: lift (left) and error 2-norm (right)

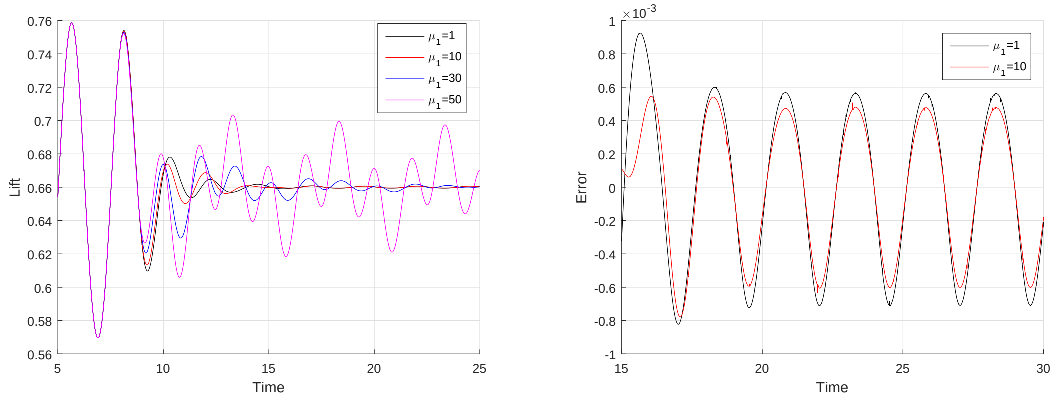


FIGURE 3.12: Lift obtained for delay $\lambda = 10$ and increased values of the gain μ_1 in the controller (left) and error obtained for the controllers with $\mu_1 = 1$ and $\mu_1 = 10$ (right)

with lower gain value the oscillatory component remains constant after several trials and no further convergence can be observed. These tests have been performed for $\lambda = 10$ which corresponds to the delay of 10% of the rotation period (however similar behaviour can be observed for higher delays). In the remainder of this section varying-gain ILC is tested in order to examine if further convergence can be obtained with this control design.

3.2.3 Varying Gain ILC

The results obtained in section 3.2.2 show that a good error convergence can be obtained for the system controlled by the combination of P and ILC controllers for the actuation delays faster than $\lambda = 1$. However, unlike for no delay case, the complete error rejection cannot be obtained and there is some oscillation left in the lift. This oscillation is more significant for higher delays. Moreover, choosing higher ILC update gain μ_1 only introduces more oscillation in the initial trials. In order to improve the performance the varying gain ILC controller can be implemented. The aim of such a controller is to damp the remaining fluctuation in the latter trials without aggravating the performance in the initial trials.

The control law now has the form described by equation (3.39) with the ILC component replaced by

$$\hat{u}_i(k) = u_{i-1}(k) + \mu(i)\Delta t e_{i-1}(k+\delta) \tag{3.42}$$

where $\mu(i)$ is the function of the trial number.

A large range of gain functions have been tested. The gain value $\mu_1(i)$ rises exponentially for approximately $20 - 30$ trials and then grows logarithmically to prevent it reaching too high value, with the effect of damaging the system. A selection of the functions is given in Table 3.1 and Fig. 3.13 (left). The tests have been performed for an actuator speed coefficient of $\lambda = 10$ and $\lambda = 5$ corresponding to approximately 10% and 20% of the period of rotation representing a relatively slow actuator. A longer simulation is required to examine if it is possible to achieve further error convergence in latter trials, therefore 75 trials have been simulated The value of the phase shift $\delta = 0$ is kept as changing it does not provide any significant improvements.

Table 3.1: Gain functions

μ_1	$constant = 1$
μ_{11}	$e^{0.1 \cdot i} \quad for\, i = 1:32$ $e^{0.1 \cdot 32} + log(10 \cdot (i-31)) \quad for\, i = 33:80$
μ_{12}	$e^{0.1 \cdot i} \quad for\, i = 1:32$ $e^{0.1 \cdot 32} + 4 \cdot log(i-31) \quad for\, i = 33:80$
μ_{13}	$e^{0.13 \cdot i} \quad for\, i = 1:23$ $e^{0.13 \cdot 23} + log(10 \cdot (i-22)) \quad for\, i = 24:80$
μ_{14}	$e^{0.13 \cdot i} \quad for\, i = 1:23$ $e^{0.13 \cdot 23} + 4 \cdot log(i-22) \quad for\, i = 24:80$

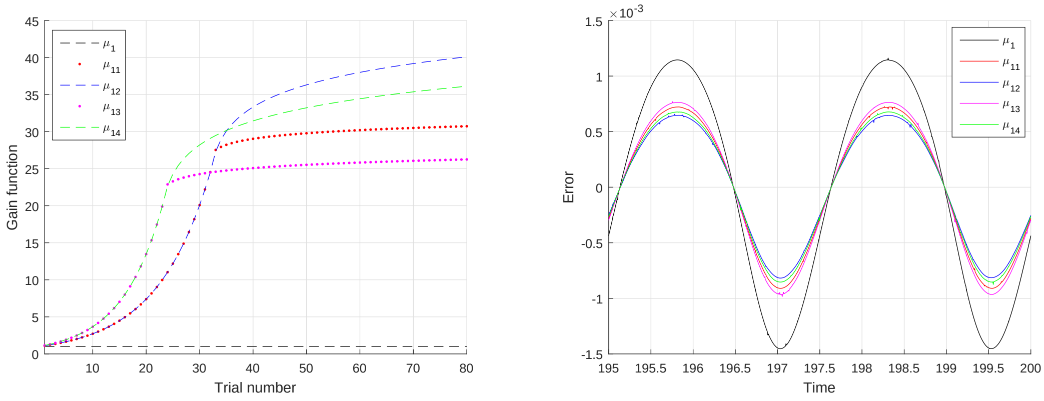


Figure 3.13: Different gain functions and error signals obtained after 75 trials

The error norms as a function of the trial number are given in Fig. 3.14 and the values of the error norms for the last trial and the total 2-norm for trials 6-75 are given in Table 3.2. The results are plotted starting from $i = 6$ after the error for the fixed gain ILC stabilizes and no further convergence can be obtained (black dashed line). It is observed that the fluctuations in error can be decreased in further trials by applying a controller with the varying gain. The values of $\mathcal{L}_2$ and $\mathcal{L}_\infty$ norms achieved after 75 trials are almost twice smaller than for the fixed gain ILC (Table 3.2). The total 2-norm for trials $6 - 75$ calculated as the sum of 2-norm for all trials decreases by $30 - 35\%$ compared to the fixed gain case for both delay coefficients λ. This appears not to be a massive difference considering the fact that the error has been already reduced by two orders of magnitude compared to the no control case. However, such reduction could make a difference in blade fatigue after years of operation. The fastest convergence is achieved for μ_{14} and the total 2-norm is the smallest for this case.

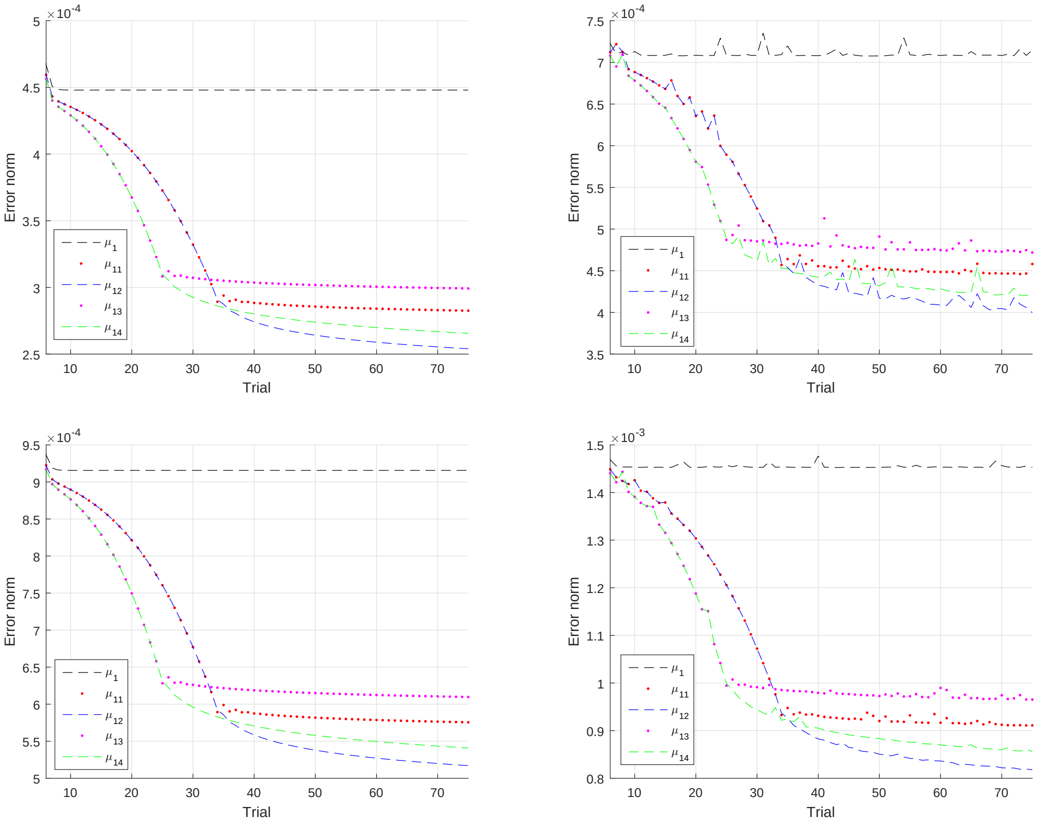


FIGURE 3.14: $\mathcal{L}_2$ (left) and $\mathcal{L}_\infty$ (right) norms obtained for controllers with different gain functions for delay coefficients $\lambda = 10$ (top) and $\lambda = 5$ (bottom)

The last important issue is to check the input signal transients. The input signal for the fixed gain controller and the varying gain controller with gain μ_{14} are given in Fig. 3.15 for initial trials (left) and later trials (right). The input signals are very similar yet they give a significant difference in the error. No unacceptable behaviour (i.e. high peaks) in the input signal is observed.

TABLE 3.2: Error norms after 75 trials

delay coefficient λ	gain function	$\mathcal{L}_2 \cdot 10^4$	$\mathcal{L}_\infty \cdot 10^4$	total $\mathcal{L}_2 \cdot 10^4$
10	μ_1 (fixed)	4.5	7.1	314
10	μ_{11}	2.8	4.6	230
10	μ_{12}	2.5	4.0	221
10	μ_{13}	3.0	4.7	230
10	μ_{14}	2.6	4.2	217
5	μ_1 (fixed)	9.1	15	641
5	μ_{11}	5.8	9.1	469
5	μ_{12}	5.2	8.2	451
5	μ_{13}	6.1	9.6	469
5	μ_{14}	5.4	8.6	442

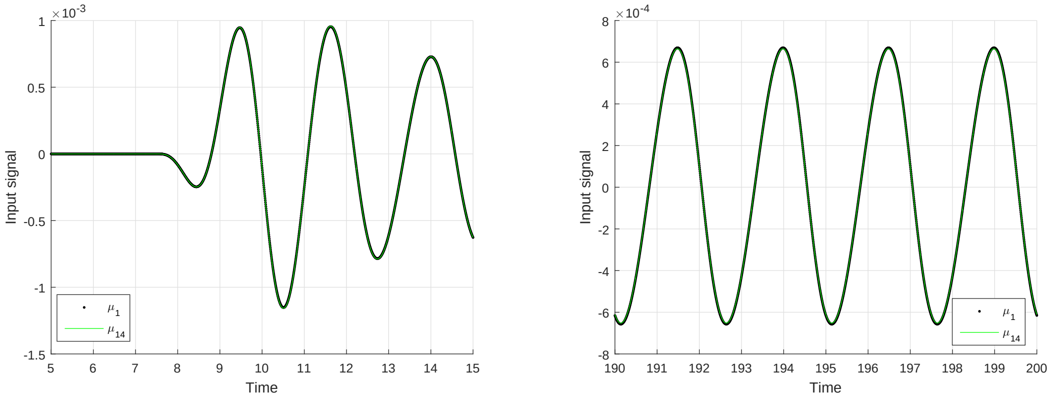


FIGURE 3.15: Input signal of the controllers with gains μ_1 and μ_{14} for initial trials (left) and latter trials (right)

3.2.4 Validation of Model-free Designs for the Flow with Vortical Disturbances

In the previous subsection it is assumed that the flow is periodic and therefore the variation in the lift comes purely from the oscillatory component of the flow (Eq. 3.34) and it can be successfully damped by combination of P and ILC controllers. To validate the designs for non-periodic flow conditions the additional upstream vortices are now introduced to the flow. Such vortices will generate peaks in the lift while passing the airfoil. The flow still consists of the periodic component coming from the variation in the freestream but additional vortices are introduced upstream the body at the beginning of the simulation to represent non-deterministic conditions. This is done in the panel code where upstream vortices are added in and propagated in the same manner as the wake vortices. The strengths of these vortices are selected such that they generate significant disturbances and hence have a substancial effect on the lift as they pass the wing. Initially this is obtained with the vortex strength of $\Gamma = 0.1$ and subsequently the strength is increased to $\Gamma = 0.3$ to represent severe conditions. With this approach the variation in the lift comes not only from the significant oscillation in the freestream velocity but from large non-periodic disturbances of a kind that would be expected to occur in practice. Other physical effects such as variation in AoA or gusts could be investigated in future research.

In the first test two upstream vortices are introduced at $x_{v1} = (-20, -0.35)$ and $x_{v2} = (-30, 0.25)$ at the start of the simulation, both with strength $\Gamma = 0.1$, and hence vortex 1 will pass below the airfoil approximately at time $t = 20$ and vortex 2 above the airfoil at $t = 30$. In Fig. 3.16 (left) the lift obtained for a no control case is compared to the lift obtained for the controlled system. In Fig. 3.16 (right) the error obtained for the same robustness test is shown for no delay (fixed gain controller), $\lambda = 10$ and $\lambda = 5$ (varying gain controller). The error reduction is best for the no delay case (as expected) and slightly worse for higher delays, however compared to the no control case good error attenuation is obtained for all cases.

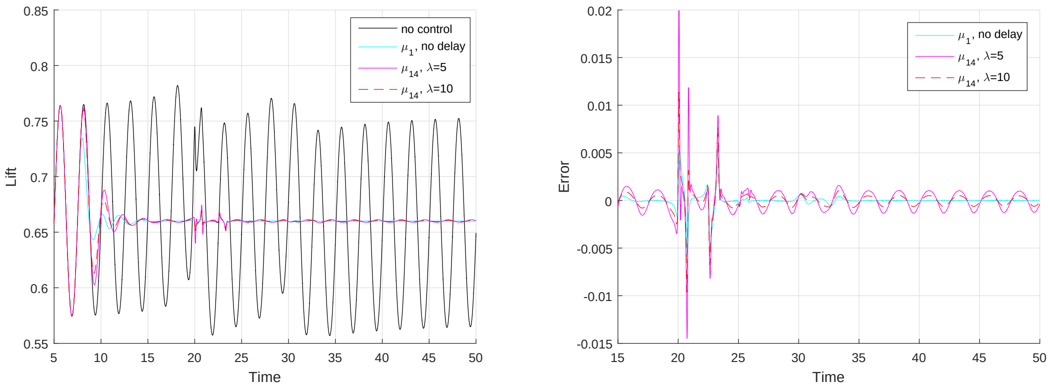


FIGURE 3.16: Robustness test 1 for non-deterministic flow: lift (left) and error (right)

For the second robustness test three vortices are introduced at $x_{v1} = (-20, -0.15)$, $x_{v2} = (-30, 0.25)$ and $x_{v3} = (-35, -0.25)$ with strengths increased to $\Gamma_1 = 0.2$, $\Gamma_2 = 0.1$ and $\Gamma_3 = 0.3$ respectively. Results for the no control case and control for delay $\lambda = 10$ and $\lambda = 5$ are shown in Fig. 3.17.

For the third robustness test, three vortices with higher strengths are introduced with the third vortex passing much closer to the body at $x_{v1} = (-20, -0.15)$, $x_{v2} = (-30, 0.25)$ and $x_{v3} = (-35, -0.1)$. This represents more severe conditions. It is observed that for the higher ILC gain μ_{14} the error is slightly higher than for the fixed gain with a lower value $\mu_1 = 1$ when the strong vortex passes the body (Fig. 3.18). Conversely, the attenuation of the periodic oscillatory component is better for the varying gain ILC. The reason behind it is that ILC can only learn from repeating disturbances.

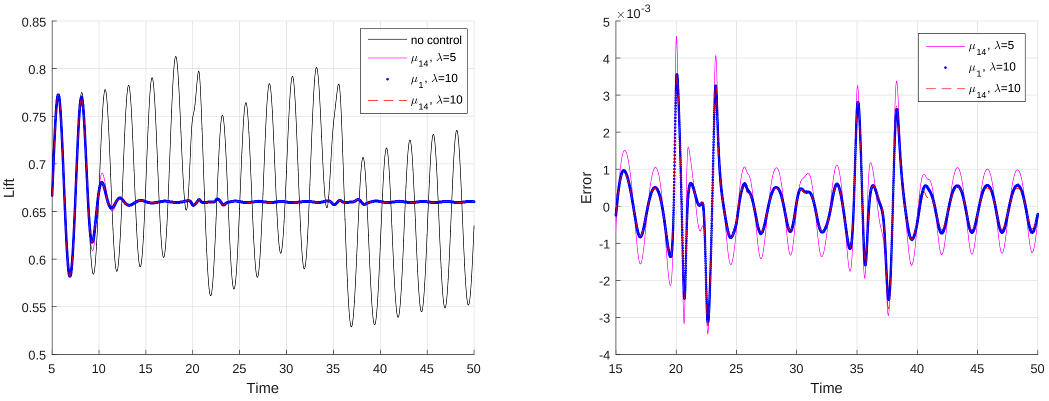


FIGURE 3.17: Robustness test 2 for non-deterministic flow: lift (left) and error (right)

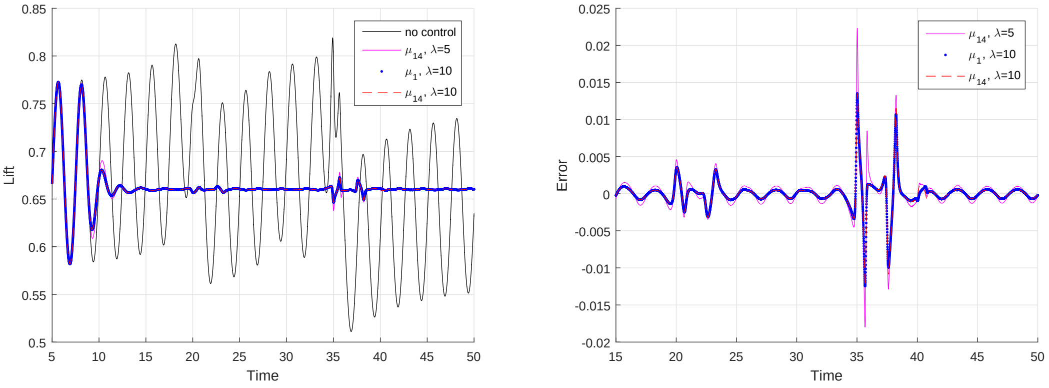


FIGURE 3.18: Robustness test 3 for non-deterministic flow: lift (left) and error (right)

Finally, the input signals for robustness tests 2 and 3 are given in Fig. 3.19 and 3.20 (left). The difference in the input is very small so zoomed signals are also shown (right).

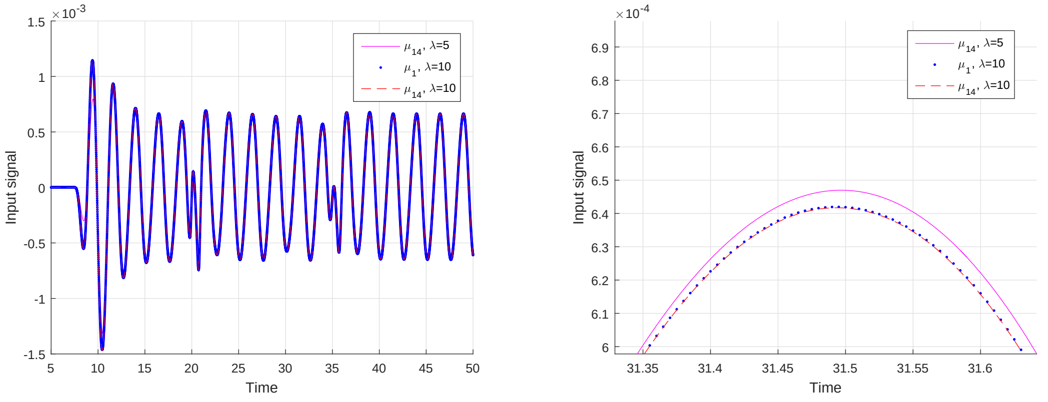


FIGURE 3.19: Control input for non-deterministic flow (test 2)

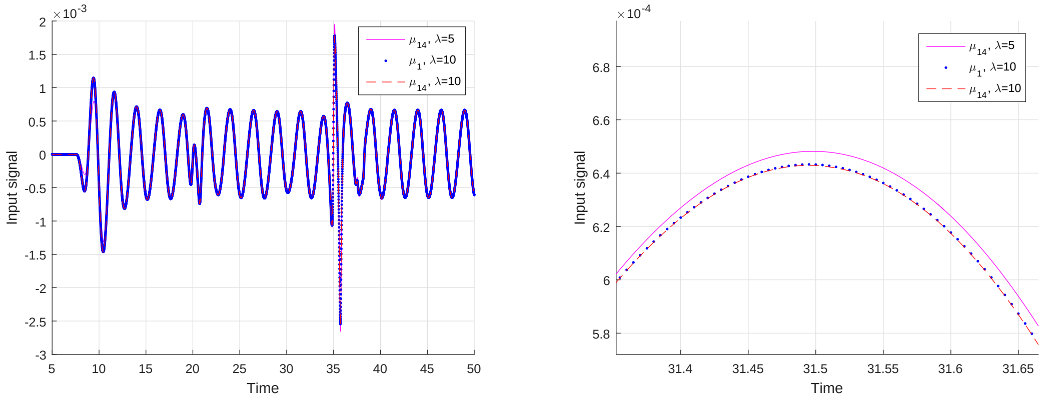


FIGURE 3.20: Control input for non-deterministic flow (test 3)

3.3 Summary

A CFD model based on a panel method is used in this work to simulate the flow past an airfoil in two dimensions. The airfoil is generated using Kàrmàn-Trefftz transform. The flow is assumed to be inviscid and the motion of the vortices is found by solving the Euler equations. Moreover, unlike in the previous work of Blackwell (2015), the wake effect is simulated by releasing vortices from the trailing edge and hence the lift is calculated using unsteady Bernoulli's equation.

The CFD simulation is used to examine model-free ILC for rejecting periodic disturbances in the lift. Circulation generation on the trailing edge is used as a model of smart rotor blade. At first, proportional controller (P) and ILC scheme are tested separately. P controller reduces the error by 90% and ILC produces a 30% reduction. However, the best attenuation in the lift error can be obtained by implementing combination of P and ILC controllers. Combination of the P controller with a phase-lead ILC with fixed gain provides a significant reduction in lift error compared to the uncontrolled case, but when the delay in actuator response is included in the system model the error stops to converge after several trials. The results in this chapter establish that further reduction in the error can be achieved by applying varying gain ILC where the gain is increased as the trial number increases. Such an approach allows further reduction of the fatigue load ($\mathcal{L}_2$) and peak load ($\mathcal{L}_\infty$) without aggravating the performance in initial trials compared to the fixed gain case. Moreover, the controllers designed in this chapter have been tested for flow with non-periodic disturbances and it has been demonstrated that the disturbances can be decreased significantly compared to the no control case for both fixed and varying gain controllers. This performance will degrade for high actuation delays. The $\mathcal{L}_2$ norm is lowest for the varying gain ILC which attenuate the periodic component in the lift, however when non-periodic disturbances pass the airfoil a slightly higher $\mathcal{L}_\infty$ error norm is measured for the varying gain ILC compared to the ILC with a fixed, lower gain. The reason for this is that the ILC controller can only compensate for the repeating disturbances in the flow.

The main issue with designing controllers using the CFD simulation is that running each experiment takes from several hours to several days and therefore looking for better gains is time taking. The solution is obtaining a state-space representation of the system as described in section 2.2.3. Moreover, model-based controllers usually give better results than auto-tunning designs due to the fact that optimization methods can be used in order to find the controller's gain. Therefore, a model-based design of the ILC scheme for the wind turbine blade with smart rotor is an obvious step for ongoing research. Such a controller should provide good along the trial performance and trial-to-trial convergence.

In this chapter a Kàrmàn-Trefftz airfoil has been used and is the result of a transformation of the circle and modelling via Proper Orthogonal Decomposition (POD) can be applied as the first step to constructing a Reduced Order Model (ROM) of the system (Bergmann et al., 2005; Bergmann and Cordier, 2008). Then, different model-based ILC schemes, i.e. schemes described in section 2.3.2, can be applied with the aim of further improving aerodynamic load performance of wind turbines with smart blades. The modelling of the system is the subject of the next chapter.

Chapter 4

Modelling of the Flow via Proper Orthogonal Decomposition

In this chapter model order reduction (MOR) of the flow past an airfoil via Proper Orthogonal Decomposition (POD) is described. The obtained model is used to reconstruct the velocity field and estimate the lift and the results are compared to the results from CFD simulation.

4.1 Background on Model Order Reduction

CFD uses numerical analysis in order to solve fluid flow problems. Despite the recent progress of CFD capability and development of high-speed computers, full Navier-Stokes simulation which characterizes the flow remains computationally expensive and leads to large-scale state equations. For model-based control system design, i.e. NOILC design, the state-space representation of the form (2.10) is required where the computational effort for calculating the control update depends on the input-output relation matrix G given by (2.13) and therefore depends on the dimension of the model. The solution to this is model order reduction. One starting point for background on MOR is research by Moore (1981) where Principal Component Analysis (PCA) have been applied to the problem of model reduction. MOR methods can be classified into three main categories (Antoulas and Sorensen, 2001):

1. Singular Value Decomposition (SVD) based approximation
2. Krylov based approximation
3. SVD-Krylov based approximations, which combine the best attributes of (1) and (2)

and their overview is given in Table 4.1 (Antoulas and Sorensen, 2001).

TABLE 4.1: MOR approximation methods

<table>
<tr><th>Nonlinear Systems</th><th colspan="2">Linear Systems</th></tr>
<tr><td>• POD methods
• Empirical grammians</td><td>• Balanced truncation
• Hankel approximation</td><td>• Realization
• Interpolation
• Lanczos
• Arnoldi</td></tr>
<tr><td colspan="2">SVD</td><td>Krylov</td></tr>
<tr><td colspan="3">SVD-Krylov</td></tr>
</table>

The main advantage of the SVD methods over Krylov methods is that they can be used for nonlinear systems as well as for linear systems (Antoulas and Sorensen, 2001). The most common MOR approach that can be applied to a nonlinear dynamical systems of the form

$$\begin{aligned} \frac{d\mathbf{x}(t)}{dt} &= F(\mathbf{x}(t), u(t)) \\ y(t) &= h(\mathbf{x}(t), u(t)) \end{aligned} \tag{4.1}$$

is Proper Orthogonal Decomposition. The fundamentals of this approach can be found in Berkooz et al. (2003). POD has been widely applied in fluid mechanics, e.g. for the problem of rotary control of the cylinder wake in Bergmann et al. (2005); Bergmann and Cordier (2008). The methodology for POD based model order reduction is:

1. generate data snapshot experimentally or numerically by applying selected input signal to the system and collecting samples of the resulting trajectory at different time steps
2. apply the SVD to the resulting snapshot matrix
3. determine the reduced order model (ROM)

As stated in step (1) in order to construct the POD ROM the experimental or numerical data set (snapshot) is required. In this work the data snapshot is generated in CFD simulation described in Chapter 3. A regular grid around the airfoil has been generated using conformal mapping described in subsection 3.1.2. The data snapshots generated in CFD simulation contain the velocity of the flow in horizontal and vertical direction for each grid point. Moreover, velocity gradient at each point is needed for the POD ROM construction in step (3). The methodology for collecting the data snapshots and calculating their gradients is given in the next subsection.

4.2 Velocity Field Around the Airfoil

The airfoil and the grid around the airfoil are generated using conformal transform as described in section 3.1.2 with

$$z = f(\zeta) \tag{4.2}$$

where $z = x + iy$ is a physical space and $\zeta = X + iY$ is a computational space. The airfoil can be represented by a circle in computational space. The velocity grid around the airfoil is generated by taking

$$\zeta = e^{i\theta} + \zeta_c \tag{4.3}$$

with lines of constant radius r and angle θ. The resulting grid is shown in Fig. 4.1 in computational space (left) and physical space (right).

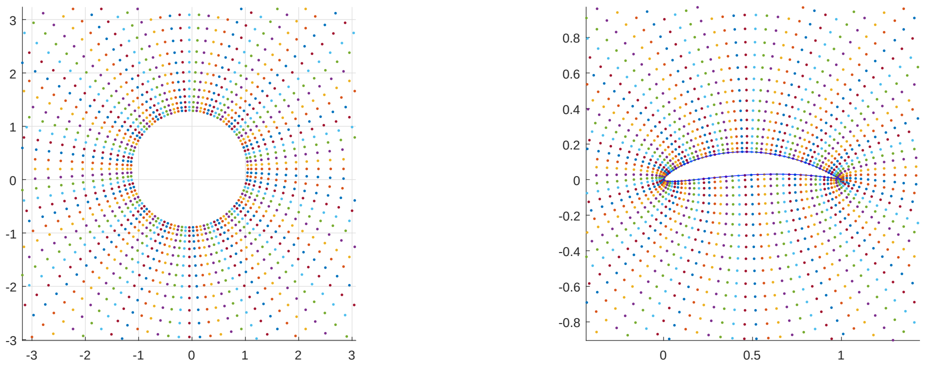


FIGURE 4.1: The grid in computational (left) and physical (right) space

The grid is of a size 100 × 81 with 100 different values of the angle θ and 81 increasing values of the radius. The grid is regular in (r,θ) with the grid points labelled (i,j) where j is the index for r and i is the index for θ as shown in Fig. 4.2.

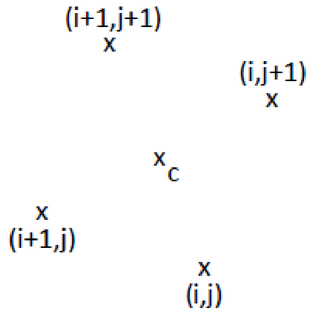


FIGURE 4.2: Grid points and cell centres

The cell centres are given by

$$(r_c, \theta_c) = \left(\frac{1}{2}(r_j + r_{j+1}), \frac{1}{2}(\theta_i + \theta_{i+1})\right) \tag{4.4}$$

For a cell centre the value of any function F_c is

$$F_c = \frac{1}{4}\left(F_{i,j} + F_{i+1,j} + F_{i,j+1} + F_{i+1,j+1}\right) \tag{4.5}$$

and the derivatives in respect to r and θ are given by

$$\left(\frac{\partial F}{\partial r}\right)_c = \frac{\frac{1}{2}(F_{i,j+1} + F_{i+1,j+1}) - \frac{1}{2}(F_{i,j} + F_{i+1,j})}{r_{j+1} - r_j} \tag{4.6}$$

$$\left(\frac{\partial F}{\partial \theta}\right)_c = \frac{\frac{1}{2}(F_{i+1,j} + F_{i+1,j+1}) - \frac{1}{2}(F_{i,j} + F_{i,j+1})}{\theta_{i+1} - \theta_i} \tag{4.7}$$

For such grid the integrals of any function F with respect to x and y which will be used as the inner product for ROM construction in the latter part of this work take the form

$$\int\int F(x,y)dxdy = \int\int F(x,y)J(x,y)rdrd\theta \tag{4.8}$$

where

$$J = \left|\frac{dz}{d\zeta}\right|^2 \tag{4.9}$$

gives the area mapping between ζ and z planes.

In a discrete form the double integral of the cell centre values of any function F is given by

$$\int\int F(x,y)d\mathbf{x}d\mathbf{y} = \sum_i\sum_j\left((F_c)_{ij}(J_c)_{ij}\frac{1}{2}(r_j + r_{j+1})(r_{j+1} - r_j)(\theta_{i+1} - \theta_i)\right) \tag{4.10}$$

For ROM construction the calculation of the derivatives of the velocity and POD modes with respect to x and y is necessary. These are given by

$$\frac{\partial F}{\partial x} = A\frac{\partial F}{\partial r} + B\frac{\partial F}{\partial \theta} \tag{4.11}$$

$$\frac{\partial F}{\partial y} = C\frac{\partial F}{\partial r} + D\frac{\partial F}{\partial \theta} \tag{4.12}$$

where the coefficients A, B, C and D and the details on how to calculate them are given in Appendix A.

4.3 POD ROM for Euler Equations

Proper Orthogonal Decomposition is the method of information compression which eliminates the redundant information from snapshots generated experimentally or numerically. As the POD functions are generated based on the data snapshot, the information in modes strictly depends on the data set and so does the ability of the modes to approximate the system's state.

The flow of inviscid, incompressible fluid with constant density is governed by the Euler equations

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p \\ \nabla \cdot \mathbf{u} = 0 \end{cases} \tag{4.13}$$

where $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$ is the velocity vector with components in x and y direction denoted as u and v respectively such that $\mathbf{u} = \binom{u}{v}$, p denotes the pressure, and t denotes the time.

For the flow past an airfoil the total velocity field of the flow can be decomposed as a sum of the steady mean flow component $\mathbf{u}_m$=$\mathbf{u}_m(\mathbf{x})$ and the unsteady component of the flow $\hat{\mathbf{u}} = \hat{\mathbf{u}}(\mathbf{x}, t)$

$$\mathbf{u} = \mathbf{u}_m + \hat{\mathbf{u}} \tag{4.14}$$

Let the inner product be defined as the integral of the dot product between $\mathbf{a}$ and $\mathbf{b}$

$$< \mathbf{a}, \mathbf{b} >= \int \mathbf{a} \cdot \mathbf{b} \, d\mathbf{x} \tag{4.15}$$

The unsteady flow $\hat{\mathbf{u}}(\mathbf{x}, t)$ can be represented by a set of spatial POD modes $\boldsymbol{\phi}_j(\mathbf{x}) = \binom{\phi_{ju}}{\phi_{jv}}$ and their coefficients $a_j(t)$ (Bergmann et al., 2005; Bergmann and Cordier, 2008)

$$\hat{\mathbf{u}}(\mathbf{x}, t) = \sum_{j=1}^{N} a_j(t) \boldsymbol{\phi}_j(\mathbf{x}) \tag{4.16}$$

where $\hat{\mathbf{u}}(\mathbf{x}, t) = \mathbf{u}(\mathbf{x}, t) - \mathbf{u}_m(\mathbf{x})$ and N is the number of modes.

The coefficients $a_j(t)$ and modes $\boldsymbol{\phi}_j(\mathbf{x})$ can be found through singular value decomposition of the data snapshot $\mathcal{U}$

$$\mathcal{U} = [\hat{\mathbf{u}}(\mathbf{x}, t_1), \hat{\mathbf{u}}(\mathbf{x}, t_2), ..., \hat{\mathbf{u}}(\mathbf{x}, t_N)] \tag{4.17}$$

or alternatively through eigendecomposition of the Gramian of the snapshot matrix $\mathbf{C} = \mathcal{U}^T \mathcal{U}$

$$\mathbf{C} \mathbf{a}_j = \lambda_j \mathbf{a}_j \tag{4.18}$$

where $C_{ij} =< \hat{\mathbf{u}}_i, \hat{\mathbf{u}}_j >$, $\lambda_1, ..., \lambda_N$ are the eigenvalues sorted such that $\lambda_1 > \lambda_2 > ... > \lambda_N$ and $\mathbf{a}_1, ..., \mathbf{a}_N$ are the eigenvectors.

In order to obtain a reduced order model of the system the total velocity field (4.14) is substituted back into the governing equations (4.13)

$$\frac{\partial \hat{\mathbf{u}}}{\partial t} + (\mathbf{u}_m + \hat{\mathbf{u}}) \cdot \nabla(\mathbf{u}_m + \hat{\mathbf{u}}) = -\nabla p \tag{4.19}$$

Due to complexity of the following model equations $\boldsymbol{\phi}_j$ is be used to denote the spatial modes $\boldsymbol{\phi}_j(\mathbf{x})$ in the remaining part of this chapter. Substituting the POD of the unsteady part of the flow (4.16) to (4.19) gives

$$\begin{aligned} -\nabla p = \frac{d}{dt}\left(\sum_{j=1}^{N} a_j(t)\boldsymbol{\phi}_j\right) + \mathbf{u}_m \cdot \nabla \mathbf{u}_m + \sum_{j=1}^{N} a_j(t)\mathbf{u}_m \cdot \nabla \boldsymbol{\phi}_j \\ + \sum_{j=1}^{N} a_j(t)\boldsymbol{\phi}_j \cdot \nabla \mathbf{u}_m + \sum_{j=1}^{N}\sum_{k=1}^{N} a_j(t)a_k(t)\boldsymbol{\phi}_j \cdot \nabla \boldsymbol{\phi}_k \end{aligned} \tag{4.20}$$

as $\frac{\partial \mathbf{u}_m}{\partial t} = 0$.

Now taking the inner product with respect to $\boldsymbol{\phi}_i$ gives

$$\begin{aligned} < \boldsymbol{\phi}_i, -\nabla p > = \frac{d}{dt}\sum_{j=1}^{N} a_j(t) < \boldsymbol{\phi}_i, \boldsymbol{\phi}_j > + < \boldsymbol{\phi}_i, \mathbf{u}_m \cdot \nabla \mathbf{u}_m > \\ + \sum_{j=1}^{N} a_j(t) < \boldsymbol{\phi}_i, \mathbf{u}_m \cdot \nabla \boldsymbol{\phi}_j > + \sum_{j=1}^{N} a_j(t) < \boldsymbol{\phi}_i, \boldsymbol{\phi}_j \cdot \nabla \mathbf{u}_m > \\ + \sum_{j=1}^{N}\sum_{k=1}^{N} a_j(t)a_k(t) < \boldsymbol{\phi}_i, \boldsymbol{\phi}_j \cdot \nabla \boldsymbol{\phi}_k > \end{aligned} \tag{4.21}$$

where all terms of (4.21) can be found using the velocity data generated in CFD simulation, the modes can be found through POD, and $a_j(t) = \int \hat{\mathbf{u}}(t)\boldsymbol{\phi}_j d\mathbf{x}$.

The term $\frac{d}{dt}\sum_{j=1}^{N} a_j(t) < \boldsymbol{\phi}_i, \boldsymbol{\phi}_j >$ is equal to

$$\frac{d}{dt}a_i(t) < \boldsymbol{\phi}_i, \boldsymbol{\phi}_i >= \frac{d}{dt}a_i(t) \tag{4.22}$$

since the modes are orthogonal such that $< \boldsymbol{\phi}_i, \boldsymbol{\phi}_i >= \|\boldsymbol{\phi}_i\|$ and are normalized such that $\|\boldsymbol{\phi}_i\| = 1$ (are orthonormal).

The nonlinear term from the POD $\sum_{j=1}^{N}\sum_{k=1}^{N} a_j(t)a_k(t) < \boldsymbol{\phi}_i, \boldsymbol{\phi}_j \cdot \nabla \boldsymbol{\phi}_k >$ is neglected at this point to obtain the linear approximation suitable for designing a control system using classical design techniques. The flow and lift reconstruction results in sections 4.4 and 4.5 and the following control results show that it is possible to obtain satisfactory results with this approximation. If the linear approximation was not accurate enough to provide the satisfactory performance the nonlinear term could be included in the further analysis and the nonlinear control techniques could be considered.

Moreover, for wake flows it is usual to drop the pressure term $< \boldsymbol{\phi}_i, -\nabla p >$ together with $< \boldsymbol{\phi}_i, \mathbf{u}_m \cdot \nabla \mathbf{u}_m >$ as explained in (Noack et al., 2003) and (Bergmann et al., 2005). That gives the final linear representation

$$\frac{d}{dt} a_i(t) = -\sum_{j=1}^{N} a_j(t) < \boldsymbol{\phi}_i, \mathbf{u}_m \cdot \nabla \boldsymbol{\phi}_j > - \sum_{j=1}^{N} a_j(t) < \boldsymbol{\phi}_i, \boldsymbol{\phi}_j \cdot \nabla \mathbf{u}_m > \tag{4.23}$$

where the inner products $<, >$ calculated as double integrals with respect to $d\mathbf{x}$ and $d\mathbf{y}$ are given in Table 4.2.

4.3.1 POD ROM for Oscillatory Flow

In this work the oscillatory flow

$$\mathbf{u} = \mathbf{u}_m + \mathbf{u}_0 sin(\omega t) + \hat{\mathbf{u}} \tag{4.24}$$

is considered, where $\mathbf{u}_m$ is the steady mean flow, $\mathbf{u}_0 sin(\omega t)$ is the oscillatory component $\mathbf{u}_0 = \binom{A}{0}$ and $\hat{\mathbf{u}}$ is the unsteady component.

This can be rewritten as

$$\mathbf{u} = \mathbf{u}_m \left(1 + A sin(\omega t)\right) + \hat{\mathbf{u}} \tag{4.25}$$

as $\mathbf{u}_0 \to A\mathbf{u}_m$ when $|\mathbf{x}| \to \infty$.

At each grid point (x, y) the velocity $\mathbf{u}$ has x and y component denoted as u and v respectively.

Subtracting the flow (4.25) to Euler equation (4.13) gives

$$\begin{aligned} -\nabla p = &\frac{\partial}{\partial t} \left(\mathbf{u}_m + A\mathbf{u}_m sin(\omega t) + \hat{\mathbf{u}}\right) \\ &+ \left(\mathbf{u}_m + A\mathbf{u}_m sin(\omega t) + \hat{\mathbf{u}}\right) \cdot \nabla \left(\mathbf{u}_m + A\mathbf{u}_m sin(\omega t) + \hat{\mathbf{u}}\right) \end{aligned} \tag{4.26}$$

Decomposing the unsteady part of the flow as $\hat{\mathbf{u}} = \sum_{j=1}^{N} a_j(t)\boldsymbol{\phi}_j$ gives

$$\begin{aligned} -\nabla p = &\frac{\partial}{\partial t} \left(\mathbf{u}_m + A\mathbf{u}_m sin(\omega t) + \sum_{j=1}^{N} a_j(t)\boldsymbol{\phi}_j\right) \\ &+ \left(\mathbf{u}_m + A\mathbf{u}_m sin(\omega t) + \sum_{j=1}^{N} a_j(t)\boldsymbol{\phi}_j\right) \cdot \nabla \left(\mathbf{u}_m + A\mathbf{u}_m sin(\omega t) + \sum_{j=1}^{N} a_j(t)\boldsymbol{\phi}_j\right) \end{aligned} \tag{4.27}$$

Taking the derivative and the inner product with respect to $\boldsymbol{\phi}_i$ the final state equation of the system (4.28) is obtained. The inner products are given in Table 4.2. The double integrals can be calculated on a discrete grid using (4.10).

$$\begin{aligned}\frac{d}{dt}a_i(t) = &- A\omega cos(\omega t) < \boldsymbol{\phi}_i, \mathbf{u}_m > - < \boldsymbol{\phi}_i, \mathbf{u}_m \cdot \nabla \mathbf{u}_m > \\ &- 2Asin(\omega t) < \boldsymbol{\phi}_i, \mathbf{u}_m \cdot \nabla \mathbf{u}_m > - \sum_{j=1}^{N} a_j(t) < \boldsymbol{\phi}_i, \boldsymbol{\phi}_j \cdot \nabla \mathbf{u}_m > \\ &- A^2 sin^2(\omega t) < \boldsymbol{\phi}_i, \mathbf{u}_m \cdot \nabla \mathbf{u}_m > - Asin(\omega t) \sum_{j=1}^{N} a_j(t) < \boldsymbol{\phi}_i, \boldsymbol{\phi}_j \cdot \nabla \mathbf{u}_m > \\ &- \sum_{j=1}^{N} a_j(t) < \boldsymbol{\phi}_i, \mathbf{u}_m \cdot \nabla \boldsymbol{\phi}_j > - Asin(\omega t) \sum_{j=1}^{N} a_j(t) < \boldsymbol{\phi}_i, \mathbf{u}_m \cdot \nabla \boldsymbol{\phi}_j > \\ &- \sum_{j=1}^{N} \sum_{k=1}^{N} a_j(t) a_k(t) < \boldsymbol{\phi}_i, \boldsymbol{\phi}_j \cdot \nabla \boldsymbol{\phi}_k > + < \boldsymbol{\phi}_i, \nabla p >\end{aligned} \tag{4.28}$$

4.3.2 POD ROM for the Flow with Actuation

The model described in section 4.3.1 herein does not contain the actuation component, which will represent smart devices (circulation shed into the wake). To include actuation to the model a term similar to the unsteady fluctuation must be added. It must have the property that it generates the same amount of circulation as generated by shedding vortices in the wake. The most straightforward choice is to write the velocity field as

$$\mathbf{u} = \mathbf{u}_m + \mathbf{u}_m Asin(\omega t) + \hat{\mathbf{u}} + \kappa \mathbf{u}_c \tag{4.29}$$

where κ is the sum of the control inputs from the time the circulation is turned on

$$\kappa = \sum_{0}^{t} u_t \tag{4.30}$$

and $\mathbf{u}_c = \mathbf{u}_c(\mathbf{x})$ is the velocity field generated by an airfoil with the unit circulation as shown in Fig. 4.3. The model (4.28) will contain new terms including a $\frac{\partial}{\partial t}\kappa$ term which for a discrete form can be written as

$$\frac{d}{dt}\kappa = \frac{u_{k_c}}{\Delta t} \tag{4.31}$$

where u_{k_c} is the current input signal (the circulation shed into the wake) and the terms $\kappa < \boldsymbol{\phi}_i, \mathbf{u}_c \cdot \nabla \mathbf{u}_m >$, $\kappa Asin(\omega t) < \boldsymbol{\phi}_i, \mathbf{u}_c \cdot \nabla \mathbf{u}_m >$, $\kappa < \boldsymbol{\phi}_i, \mathbf{u}_m \cdot \nabla \mathbf{u}_c >$, $\kappa Asin(\omega t) < \boldsymbol{\phi}_i, \mathbf{u}_m \cdot \nabla \mathbf{u}_c >$, $\kappa \sum_{j=1}^{N} a_j(t) < \boldsymbol{\phi}_i, \mathbf{u}_c \cdot \nabla \boldsymbol{\phi}_j >$, $\kappa \sum_{j=1}^{N} a_j(t) < \boldsymbol{\phi}_i, \boldsymbol{\phi}_j \cdot \nabla \mathbf{u}_c >$ and $\kappa^2 < \boldsymbol{\phi}_i, \mathbf{u}_c \cdot \nabla \mathbf{u}_c >$ are given in Table 4.2.

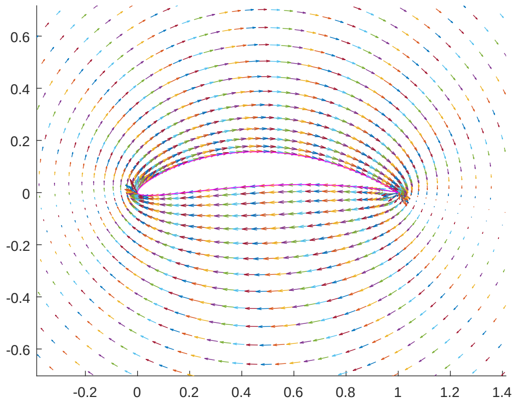


FIGURE 4.3: Unit circulation

TABLE 4.2: POD ROM terms

term	inner product	corresponding integral
1	$<\boldsymbol{\phi}_i, \mathbf{u}_m>$	$\int\int[\boldsymbol{\phi}_{iu}\mathbf{u}_m + \boldsymbol{\phi}_{iv}\mathbf{v}_m]d\mathbf{x}d\mathbf{y}$
2	$<\boldsymbol{\phi}_i, \mathbf{u}_m \cdot \nabla\mathbf{u}_m>$	$\int\int[\boldsymbol{\phi}_{iu}(\mathbf{u}_m\frac{\partial\mathbf{u}_m}{\partial x} + \mathbf{v}_m\frac{\partial\mathbf{u}_m}{\partial y}) + \boldsymbol{\phi}_{iv}(\mathbf{u}_m\frac{\partial\mathbf{v}_m}{\partial x} + \mathbf{v}_m\frac{\partial\mathbf{v}_m}{\partial y})]d\mathbf{x}d\mathbf{y}$
3	$<\boldsymbol{\phi}_i, \boldsymbol{\phi}_j \cdot \nabla\mathbf{u}_m>$	$\int\int[\boldsymbol{\phi}_{iu}(\boldsymbol{\phi}_{ju}\frac{\partial\mathbf{u}_m}{\partial x} + \boldsymbol{\phi}_{jv}\frac{\partial\mathbf{u}_m}{\partial y}) + \boldsymbol{\phi}_{iv}(\boldsymbol{\phi}_{ju}\frac{\partial\mathbf{v}_m}{\partial x} + \boldsymbol{\phi}_{jv}\frac{\partial\mathbf{v}_m}{\partial y})]d\mathbf{x}d\mathbf{y}$
4	$<\boldsymbol{\phi}_i, \mathbf{u}_m \cdot \nabla\boldsymbol{\phi}_j>$	$\int\int[\boldsymbol{\phi}_{iu}(\mathbf{u}_m\frac{\partial\boldsymbol{\phi}_{ju}}{\partial x} + \mathbf{v}_m\frac{\partial\boldsymbol{\phi}_{ju}}{\partial y}) + \boldsymbol{\phi}_{iv}(\mathbf{u}_m\frac{\partial\boldsymbol{\phi}_{jv}}{\partial x} + \mathbf{v}_m\frac{\partial\boldsymbol{\phi}_{jv}}{\partial y})]d\mathbf{x}d\mathbf{y}$
5	$<\boldsymbol{\phi}_i, \boldsymbol{\phi}_j \cdot \nabla\boldsymbol{\phi}_k>$	$\int\int[\boldsymbol{\phi}_{iu}(\boldsymbol{\phi}_{ju}\frac{\partial\boldsymbol{\phi}_{ku}}{\partial x} + \boldsymbol{\phi}_{jv}\frac{\partial\boldsymbol{\phi}_{ku}}{\partial y}) + \boldsymbol{\phi}_{iv}(\boldsymbol{\phi}_{ju}\frac{\partial\boldsymbol{\phi}_{kv}}{\partial x} + \boldsymbol{\phi}_{jv}\frac{\partial\boldsymbol{\phi}_{kv}}{\partial y})]d\mathbf{x}d\mathbf{y}$
6	$<\boldsymbol{\phi}_i, \mathbf{u}_c>$	$\int\int[\boldsymbol{\phi}_{iu}\mathbf{u}_c + \boldsymbol{\phi}_{iv}\mathbf{v}_c]d\mathbf{x}d\mathbf{y}$
7	$<\boldsymbol{\phi}_i, \mathbf{u}_c \cdot \nabla\mathbf{u}_m>$	$\int\int[\boldsymbol{\phi}_{iu}(\mathbf{u}_c\frac{\partial\mathbf{u}_m}{\partial x} + \mathbf{v}_c\frac{\partial\mathbf{u}_m}{\partial y}) + \boldsymbol{\phi}_{iv}(\mathbf{u}_c\frac{\partial\mathbf{v}_m}{\partial x} + \mathbf{v}_c\frac{\partial\mathbf{v}_m}{\partial y})]d\mathbf{x}d\mathbf{y}$
8	$<\boldsymbol{\phi}_i, \mathbf{u}_c \cdot \nabla\boldsymbol{\phi}_j>$	$\int\int[\boldsymbol{\phi}_{iu}(\mathbf{u}_c\frac{\partial\boldsymbol{\phi}_{ju}}{\partial x} + \mathbf{v}_c\frac{\partial\boldsymbol{\phi}_{ju}}{\partial y}) + \boldsymbol{\phi}_{iv}(\mathbf{u}_c\frac{\partial\boldsymbol{\phi}_{jv}}{\partial x} + \mathbf{v}_c\frac{\partial\boldsymbol{\phi}_{jv}}{\partial y})]d\mathbf{x}d\mathbf{y}$
9	$<\boldsymbol{\phi}_i, \mathbf{u}_c \cdot \nabla\mathbf{u}_c>$	$\int\int[\boldsymbol{\phi}_{iu}(\mathbf{u}_c\frac{\partial\mathbf{u}_c}{\partial x} + \mathbf{v}_c\frac{\partial\mathbf{u}_c}{\partial y}) + \boldsymbol{\phi}_{iv}(\mathbf{u}_c\frac{\partial\mathbf{v}_c}{\partial x} + \mathbf{v}_c\frac{\partial\mathbf{v}_c}{\partial y})]d\mathbf{x}d\mathbf{y}$
10	$<\boldsymbol{\phi}_i, \mathbf{u}_m \cdot \nabla\mathbf{u}_c>$	$\int\int[\boldsymbol{\phi}_{iu}(\mathbf{u}_m\frac{\partial\mathbf{u}_c}{\partial x} + \mathbf{v}_m\frac{\partial\mathbf{u}_c}{\partial y}) + \boldsymbol{\phi}_{iv}(\mathbf{u}_m\frac{\partial\mathbf{v}_c}{\partial x} + \mathbf{v}_m\frac{\partial\mathbf{v}_c}{\partial y})]d\mathbf{x}d\mathbf{y}$
11	$<\boldsymbol{\phi}_i, \boldsymbol{\phi}_j \cdot \nabla\mathbf{u}_c>$	$\int\int[\boldsymbol{\phi}_{iu}(\boldsymbol{\phi}_{ju}\frac{\partial\mathbf{u}_c}{\partial x} + \boldsymbol{\phi}_{jv}\frac{\partial\mathbf{u}_c}{\partial y}) + \boldsymbol{\phi}_{iv}(\boldsymbol{\phi}_{ju}\frac{\partial\mathbf{v}_c}{\partial x} + \boldsymbol{\phi}_{jv}\frac{\partial\mathbf{v}_c}{\partial y})]d\mathbf{x}d\mathbf{y}$

As previously discussed the nonlinear terms from POD are neglected in order to obtain a linear representation. Also, the pressure term $< \boldsymbol{\phi}_i, -\nabla p >$ and $< \boldsymbol{\phi}_i, \mathbf{u}_m \cdot \nabla \mathbf{u}_m >$ are dropped as explained in the previous subsection. Finally, the state equation has the form

$$\begin{aligned}\frac{d}{dt}a_i(t) = -\Bigg[& \sum_{j=1}^{N} a_j(t)\big[(1+Asin(\omega t))(< \boldsymbol{\phi}_i, \boldsymbol{\phi}_j \cdot \nabla \mathbf{u}_m > + < \boldsymbol{\phi}_i, \mathbf{u}_m \cdot \nabla \boldsymbol{\phi}_j >)\big] \\ & + \kappa\big[(1+Asin(\omega t))(< \boldsymbol{\phi}_i, \mathbf{u}_m \cdot \nabla \mathbf{u}_c > + < \boldsymbol{\phi}_i, \mathbf{u}_c \cdot \nabla \mathbf{u}_m >)\big] \\ & + \kappa \sum_{j=1}^{N} a_j(t)\big[< \boldsymbol{\phi}_i, \boldsymbol{\phi}_j \cdot \nabla \mathbf{u}_c > + < \boldsymbol{\phi}_i, \mathbf{u}_c \cdot \nabla \boldsymbol{\phi}_j > \big] \\ & + A\omega cos(\omega t) < \boldsymbol{\phi}_i, \mathbf{u}_m > + 2Asin(\omega t) < \boldsymbol{\phi}_i, \mathbf{u}_m \cdot \nabla \mathbf{u}_m > + \frac{d}{dt}\kappa < \boldsymbol{\phi}_i, \mathbf{u}_c > \Bigg]\end{aligned} \tag{4.32}$$

4.4 Flow Reconstruction Using POD ROM

The state equation derived in the previous section has been used for construction of the reduced order model based on POD. The modes obtained through POD are shown in this section and the reconstruction of the flow around the airfoil from the modes and their coefficients is done. Two cases are described in this section: the flow with an oscillatory component with a small period ($T = 2.5$) and the flow with ten times higher period ($T = 25$). This second case represents more realistic example, however in reality the period of oscillation will be even higher. The surface velocity reconstructed by the models shown in this subsection will be used to reconstruct the lift in the further part of this chapter.

4.4.1 Flow Reconstruction for T=2.5

The total velocity field $\mathbf{u}$ has been generated using CFD simulation described in Chapter 3 with parameters $A = 0.1$ and $T = 2.5$ used for the oscillatory flow in the simulation. Firstly, the uncontrolled case is considered where the total control input $\kappa = 0$. The velocity data has been taken at time instances $t = 0 : 0.1 : 100$ resulting in 1001 snapshots. The unsteady part of the flow $\hat{\mathbf{u}}$ has been calculated as

$$\hat{\mathbf{u}} = \mathbf{u} - \big(\mathbf{u}_m + \mathbf{u}_m Asin(\omega t)\big) \tag{4.33}$$

where the mean velocity field $\mathbf{u}_m$ has been obtained by running the CFD simulation with no oscillatory component ($A = 0$) long enough to obtain the steady state.

The quiver plots of the total velocity field and the unsteady part of the flow at the trailing edge for $t = 70$ are given in Fig. 4.4. POD decomposition of the unsteady part of the flow has been performed. 301 snapshots for $t = 70 : 0.1 : 100$ are used to calculate the modes resulting in 301 modes. The data from the beginning of the simulation is not used as the CFD simulation code needs a couple cycles to generate the wake effect. The energy in modes is calculated as

$$E_j[\%] = \frac{\lambda_j}{\sum_{j=1}^{N} \lambda_j} \cdot 100\% \tag{4.34}$$

where λ_j denotes the eigenvalue corresponding to the j-th mode. For the reconstruction of uncontrolled, oscillatory flow only two modes are sufficient to represent over 90% of the information in the flow as shown in Fig. 4.5 (right). The energy in each mode j is shown in Fig. 4.5 (left) in logarithmic scale.

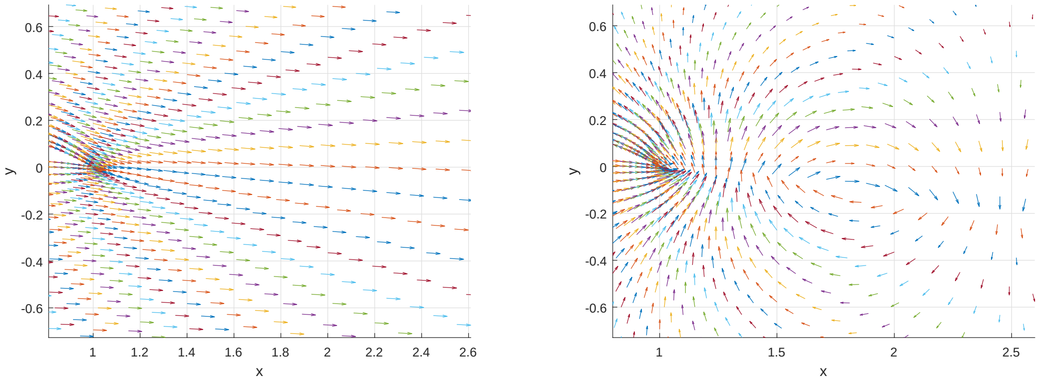


FIGURE 4.4: Total flow $\mathbf{u}$ and unsteady flow $\hat{\mathbf{u}}$ at the trailing edge at $t = 70$ (uncontrolled, oscillatory flow with $T = 2.5$)

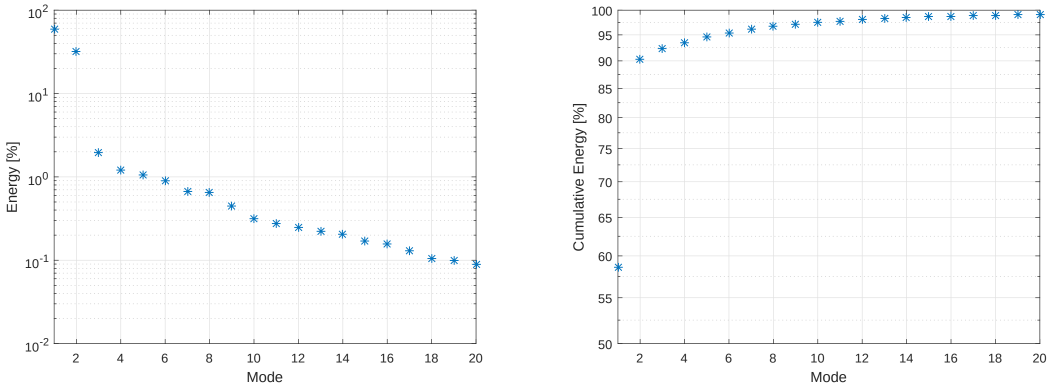


FIGURE 4.5: The energy in particular modes and the cumulative energy for modes $1 : j$ (uncontrolled oscillatory flow with $T = 2.5$)

The contour plots of the first two modes are given in Fig. 4.6. These are vortex shedding modes which are esentially the same with a streamwise origin shift. Hence, they can model the propagation of the wake where vortices of alternating sign are convected downstream as shown in Fig. 4.7. Here the positions of the vortices shed from the trailing edge are shown, where the different colours represent the sign of the vortex (purple - negative, green - positive).

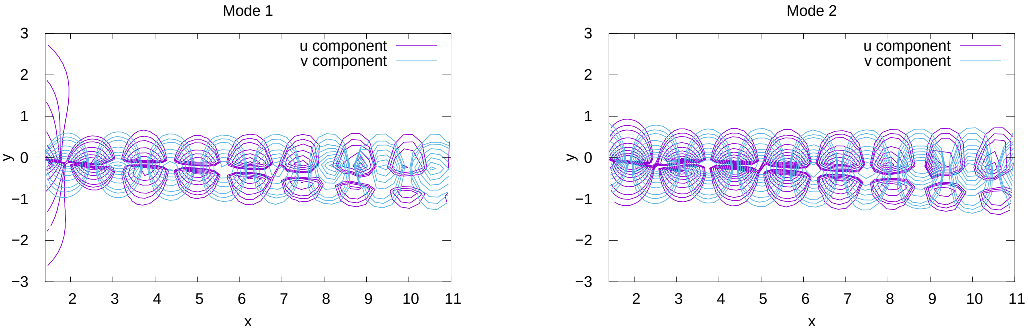


FIGURE 4.6: Vortex shedding modes which model the wake (uncontrolled case)

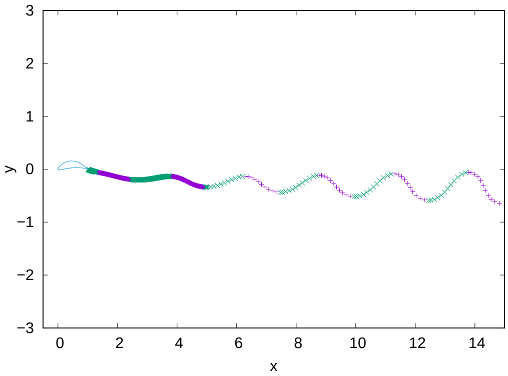


FIGURE 4.7: Uncontrolled wake

The snapshot velocity can be reconstructed very precisely from the modes and the POD coefficients obtained in the decomposition (Fig. 4.8–4.9). Subsequently, the POD coefficients $a_i(t)$ can be estimated using the state equation for oscillatory flow (4.28). For this purpose the POD coefficients for $t = t_0$ are used as the initial state and the derivatives of the coefficients $\frac{d}{dt}a_i(t)$ are calculated using 4-th order Runge-Kutta method. Two modes are used for the reconstruction. In Fig. 4.8 (bottom) the reconstruction of the flow for the 30-th grid point on the body surface counting from the trailing edge (top of the upper surface of the airfoil) is shown. This is only a representative example and the similar behaviour is observed for other points on the surface. The coefficients obtained using the model compared to the coefficients obtained in decomposition are given in Fig. 4.8 (top).

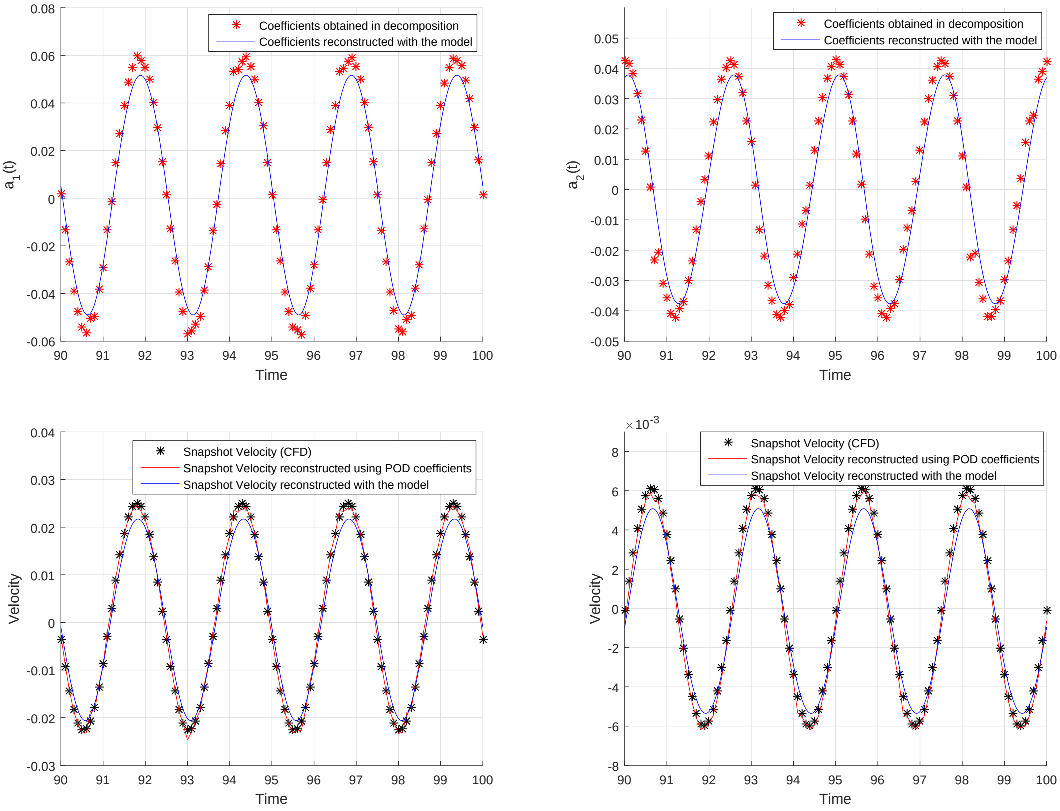


FIGURE 4.8: Reconstructed coefficients: $a_1(t)$ (top left), $a_2(t)$ (top right) and reconstructed snapshot velocity (uncontrolled case, $T = 2.5$): u component (bottom left), v component (bottom right) for the centre grid point at the top surface

In this research reconstruction of the velocity on the grid points just above and just below the airfoil surface is of particular interest as these velocity values are used to estimate the lift as explained in section 3.1.4. The comparison of the velocity values obtained in CFD simulation and reconstructed using the reduced order model for $t = 99$ for all 100 grid points on the airfoil surface are shown in Fig. 4.9. The first 55 grid points are placed just above the top surface of the airfoil and the remaining 45 points just below the bottom surface. The reconstruction of the u component of the unsteady velocity component is shown on top left and of the v component on top right. The reconstruction of total velocity is shown on the bottom of the Fig. 4.9 (left and right for u and v components respectively).

A very good reconstruction of the surface velocity is obtained for this uncontrolled case with slightly more accurate reconstruction of the v component for this particular time instance $t = 99$. Slightly higher reconstruction errors for the v component are obtained for the time instances for which the $a_i(t)$ coefficient reconstruction is less accurate (e.g. $t = 98$) but, in general, the reconstruction is good for all time instances leading to an accurate lift reconstruction (see section 4.5).

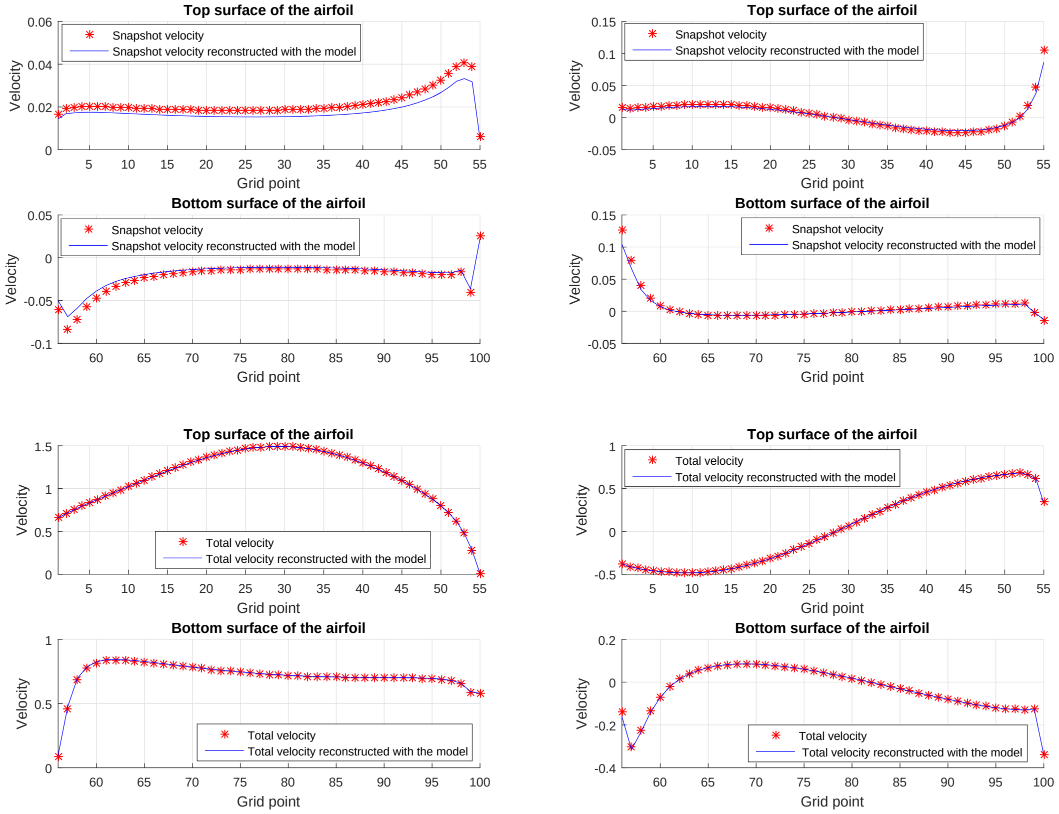


FIGURE 4.9: The unsteady flow for $t = 99$ (uncontrolled case): u component (top left) and v component (top right) and the total flow: u component (bottom left) and v component (bottom right)

A similar reconstruction can be performed for the controlled case. In such case the input signal used to generate the snapshots is no longer equal to zero ($\kappa \neq 0$). To obtain the modes that will perform well for different input signals the excitation used to generate the snapshot for the model construction was chosen as a sum of sine waves of different frequencies. For the $T = 2.5$ case a following input signal has been chosen

$$u(t) = \mu\big(sin(\frac{\omega}{3}t) + sin(\frac{\omega}{2}t) + sin(\omega t) + sin(2\omega t) + sin(2.5\omega t)\big) \tag{4.35}$$

where $u(t)$ is the input at time t such that $\kappa = \sum_0^t u_t$ and $\mu = -0.00025$. The unsteady component for the POD has been calculated using

$$\hat{\mathbf{u}} = \mathbf{u} - \big(\mathbf{u}_m + \mathbf{u}_m A sin(\omega t) + \kappa \mathbf{u}_c\big) \tag{4.36}$$

where $\mathbf{u}_c$ is the velocity field for the airfoil with unit circulation that can be obtained from the panel code. The velocity data has been taken at times $t = 0 : 0.1 : 100$ resulting in data 1001 snapshots. 301 snapshots for time $t = 70 : 100$ have been used for the POD. The initial time data is not used as the control is switched on gradually. For the test with control more modes are needed for a good reconstruction of the flow compared to the uncontrolled case as shown in (Fig. 4.10), where the energy in particular modes (left) is given together with the cumulative energy (right). For the no control case only two modes were able to reconstruct over 90% of the flow, but the controlled flow is much more complex because of the interaction between the actuators and the flow and hence more modes are needed. Here $N = 6$ modes are capable of reconstructing over 80% of the energy in the flow while subsequent modes have small impact and tend to get noisy. Hence, in practice, adding more than 6 modes gave no real improvement on ability to reconstruct the flow and would result in unnecessarily bigger state space model and higher computational effort in calculations of the controller's gain matrix.

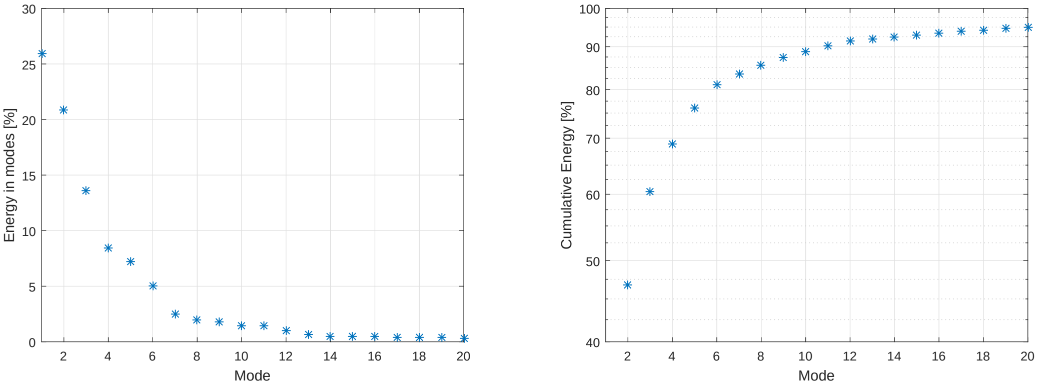


FIGURE 4.10: The energy in particular modes and the cumulative energy for modes $1 : j$ (controlled oscillatory flow with $T = 2.5$)

Six modes used for the model construction are shown in Fig. 4.11. The modes are similar to those for the no control case, modelling vortices in the wake but over a range of wavelengths. The wake for this controlled case is shown in Fig. 4.12. The control is turned on after first four cycles which can be seen in the figure where the uncontrolled wake is observed first. Subsequently, the blend of the uncontrolled wake and the control action can be seen as the control is introduced slowly according to equation (3.33). Finally, the full control action effect can be observed in the region of $x < 15$. Fig. 4.12 shows a much more disordered wake than that for the uncontrolled case (Fig. 4.7). Note here that this figure shows the position of the discrete vortices shed from the trailing edge, it does not show the large scale vortices which are commonly found in a vortex street behind a body. However, it does indicate that including frequencies in the forcing (4.35) that are higher than the fundamental frequency from the rotation of the blade generates shorter length scale structures in the wake, as might be expected.

In Fig. 4.13 the reconstruction of the flow for the 30-th grid point on the body surface counting from the trailing edge (top of the upper surface of the airfoil) is given. The accuracy of the reconstruction is far worse than for the uncontrolled case. The u component of the velocity is reconstructed pretty well but a significantly higher reconstruction error for the v component is obtained.

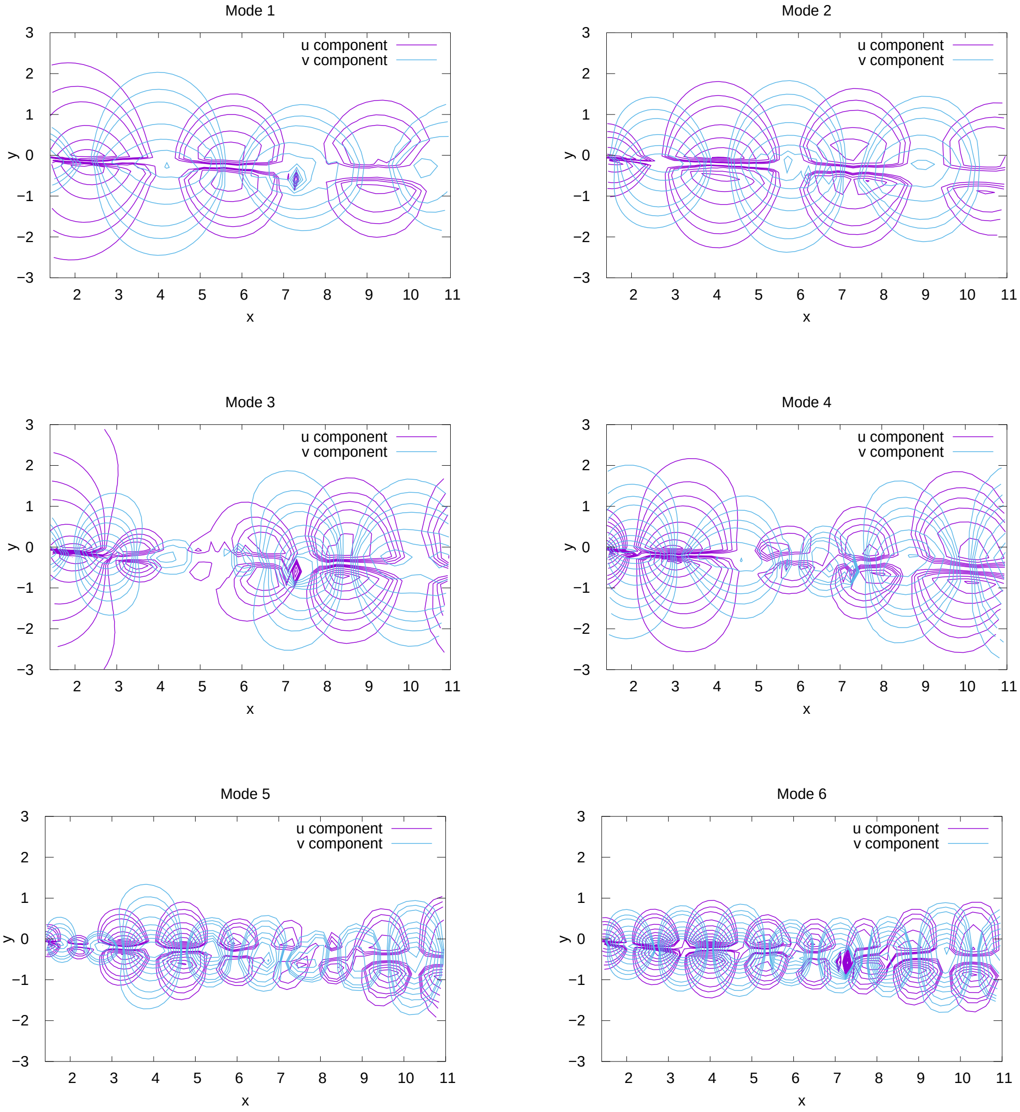


FIGURE 4.11: Contour plots of the vortex shedding modes which model the wake effect

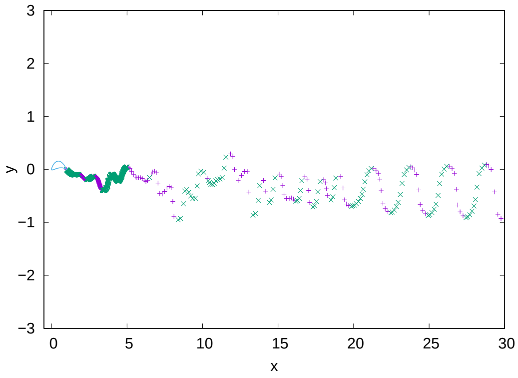


FIGURE 4.12: Controlled wake

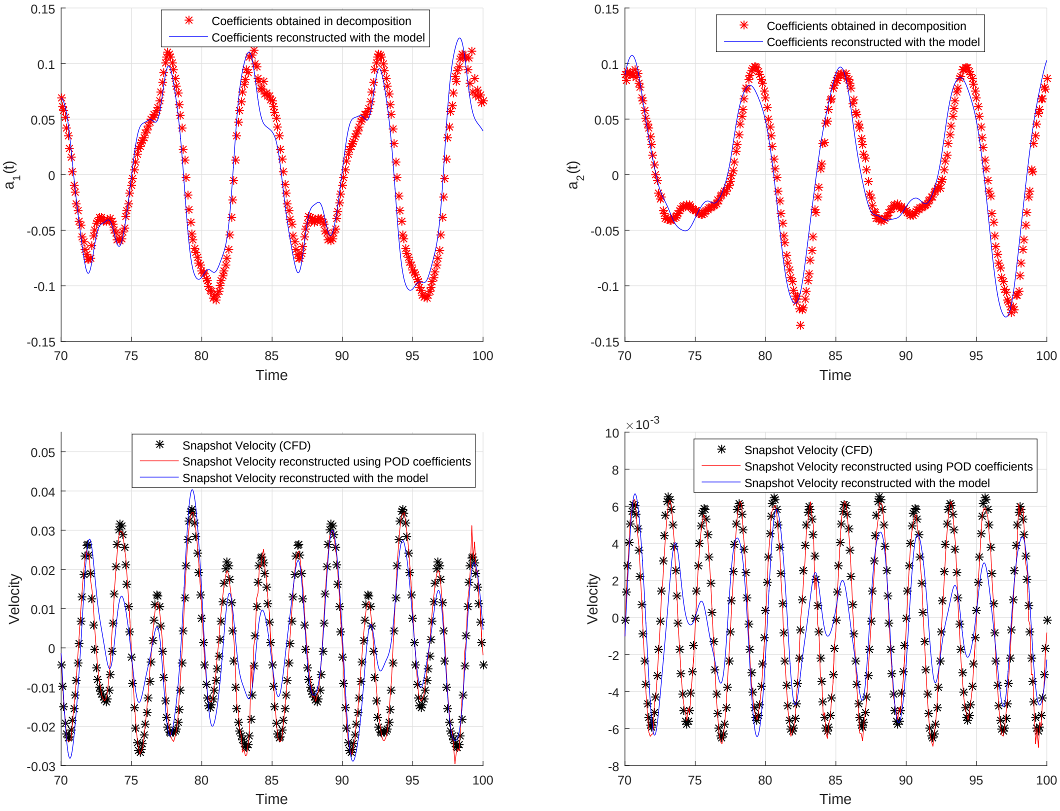


FIGURE 4.13: Reconstructed coefficients: $a_1(t)$ (top left), $a_2(t)$ (top right) and reconstructed snapshot velocity (controlled flow with $T = 2.5$): u component (bottom left), v component (bottom right) for the centre grid point at the top surface

The reconstruction is also more accurate for some time points than the others. In Fig. 4.14 (top) the surface snapshot velocity for $t = 72$ is shown and very good reconstruction compared to the original data snapshot is obtained with the model. Total velocity on the surface is shown on the bottom of the figure. In Fig. 4.15 the surface snapshot velocity for $t = 93$ is shown when the reconstruction is much less accurate than for $t = 72$.

Summarising, modelling of the controlled oscillatory flow with a period $T = 2.5$ is challenging. The velocity can be reconstructed for some of the time instances but for the others the reconstruction is not as accurate.

4.4.2 Flow Reconstruction for T=25

The same modelling procedure has been performed for a higher period of oscillatory component $T = 25$. This is a more realistic case than the previous case with $T = 2.5$ as for large wind turbines this period values are even larger. For the larger value of T some parameters of the CFD simulation have to be changed. The simulation time is now much longer with $t = 0 : 500$ and the velocity snapshots are taken every 0.5. This results

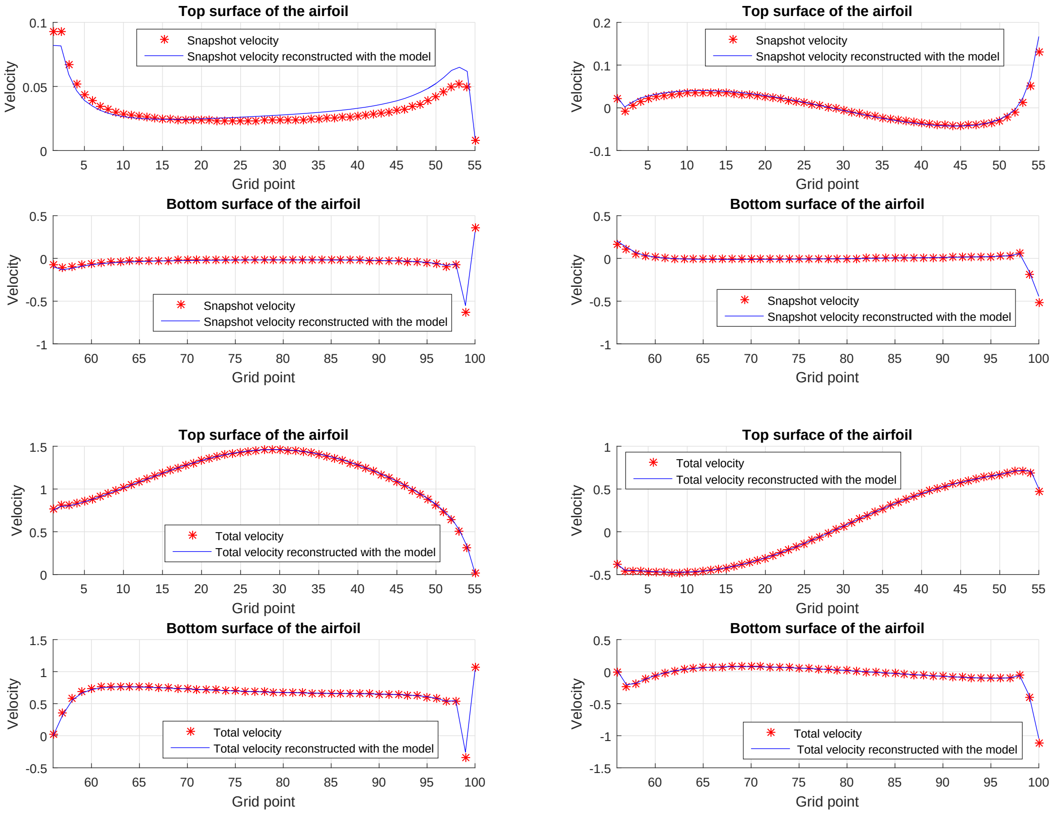


FIGURE 4.14: The unsteady flow (snapshot) for $t = 72$ (controlled case, $T = 2.5$): u component (top left) and v component (top right) and the total flow: u component (bottom left) and v component (bottom right)

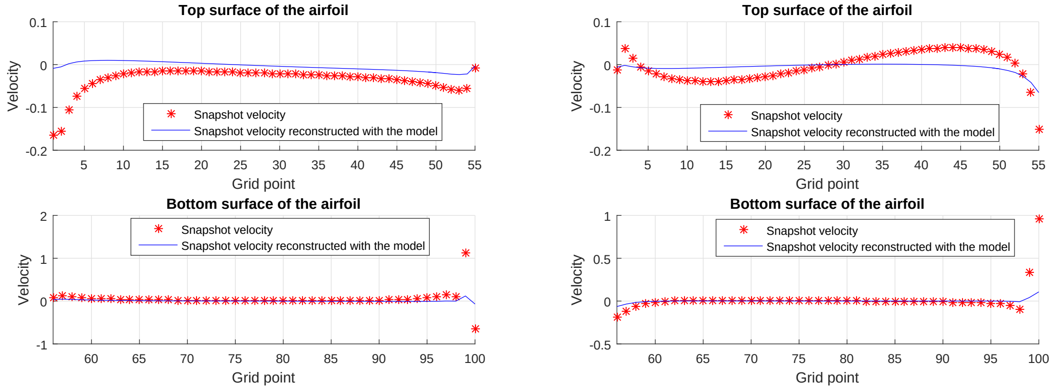


FIGURE 4.15: The unsteady flow for $t = 93$ (controlled case, $T = 2.5$): u component (left) and v component (right)

in 1001 snapshots. For the construction of the model 401 snapshots for $t = 300 : 500$ are used. Various excitation signals have been used to construct the model. The results presented below have been obtained for the sum of sine and cosine waves of frequencies higher than the frequency of the oscillatory component given by

$$u(t) = \mu\big(sin(\omega t) + sin(2\omega t) + sin(3\omega t) + cos(\omega t) + cos(2\omega t) + cos(3\omega t)\big) \tag{4.37}$$

where $\mu = \frac{0.001}{6}$.

For this case four or more modes are needed to reconstruct over 90% energy in the flow as shown in Fig. 4.16.

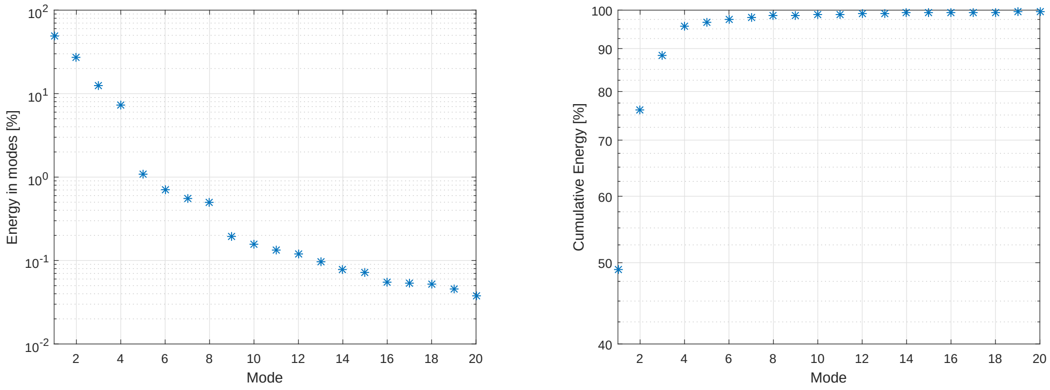


FIGURE 4.16: The energy in particular modes and the cumulative energy for modes $1:j$ (uncontrolled oscillatory flow with $T = 2.5$)

The best reconstruction accuracy obtained using $N = 6$ modes with the cumulative energy of 97.40%. Four most significant coefficients obtained in the decomposition and reconstructed with the model for $t = 400 : 500$ and $N = 6$ are shown in Fig. 4.17. For this case the reconstructed coefficients are slightly out of phase. Also the peaks values of the coefficients and therefore the peak values of the reconstructed velocity are lower than for the original snapshot. This is shown in Fig. 4.18, where the velocity at the top point of the airfoil is shown for $t = 400 : 500$. To sum up, the reconstruction is very accurate for some time instances and less accurate for the others (especially when the excitation signal reaches its peak value).

The reconstruction of the snapshot velocity at the surface is given in Fig. 4.19 for $t = 490$ when a good reconstruction is obtained (top) and for $t = 495$ when the reconstruction is less accurate (bottom). The values of the snapshot velocity, especially u component, are very small for the latter and do not have a huge affect on the total velocity.

The purpose of this work is to develop POD models which give good predictions of the lift of the airfoil, the quantity required for control purposes. In the next section it is shown that the approximate models provide the good agreement between the lift estimated using the model and the lift obtained in the CFD evaluation (Fig. 4.20). The lift can be calculated from the pressure distribution which can be obtained from the reconstructed velocity fields shown in the previous sections. However, while applying control schemes described in this work physically on wind turbines the pressure sensors

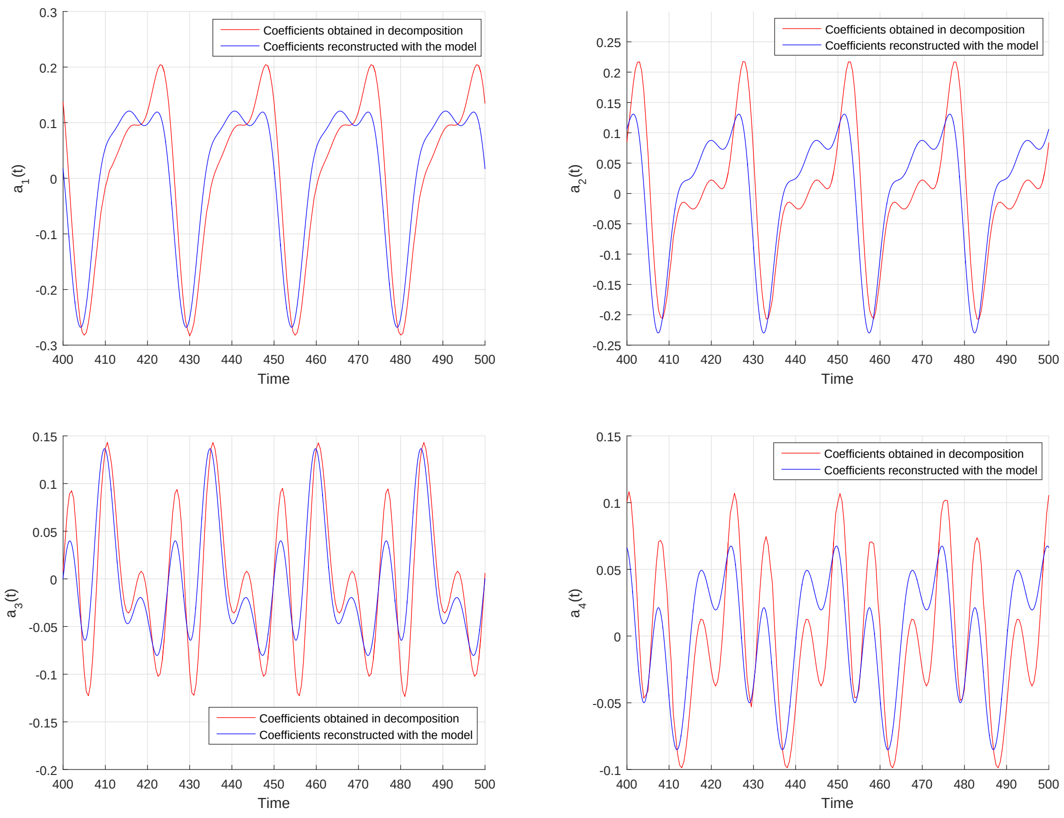


FIGURE 4.17: Reconstructed coefficients: $a_1(t)$ (top left), $a_2(t)$ (top right), $a_3(t)$ (bottom left), $a_4(t)$ (bottom right),

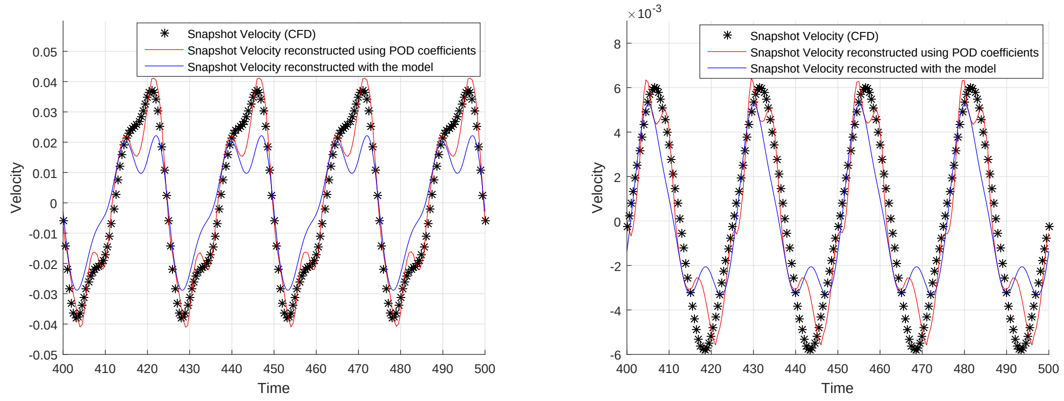
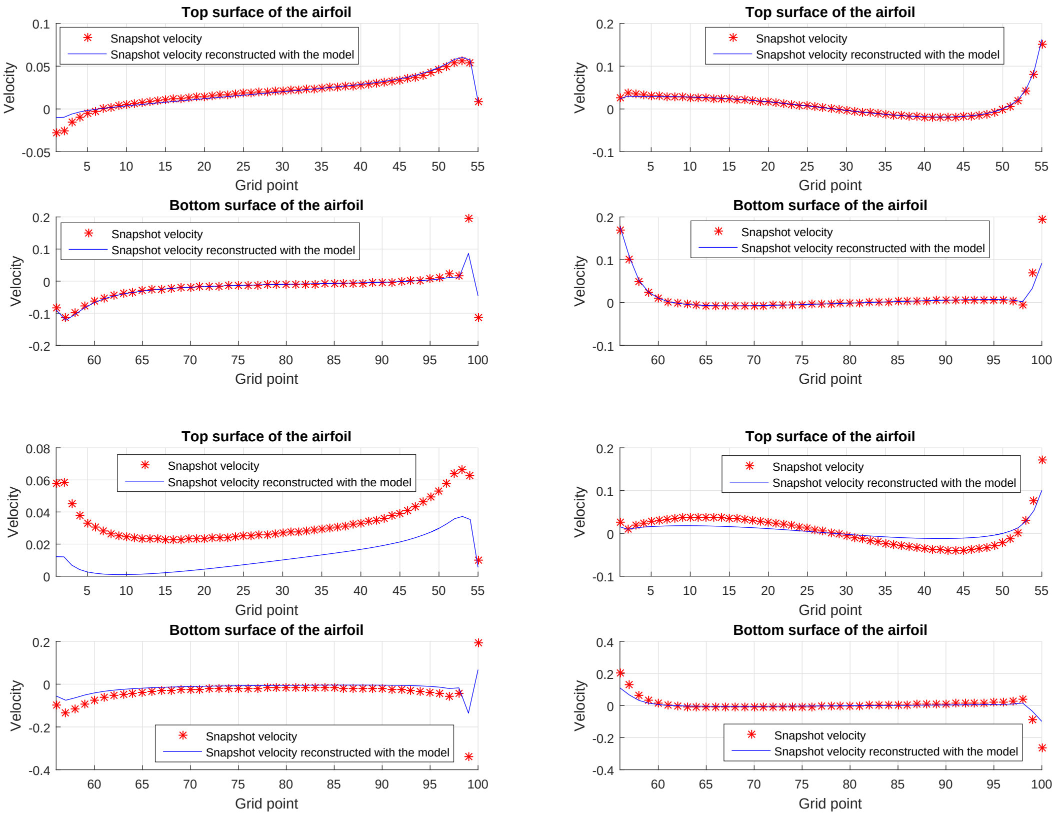


FIGURE 4.18: Reconstructed snapshot velocity (controlled flow with $T = 25$): u component (left) and v component (right) for the centre grid point at the top surface

would have to be used to estimate the lift on the blade. Detailed analysis of the pressure sensing/estimation is a future work direction.



Top surface of the airfoil

Bottom surface of the airfoil

Top surface of the airfoil

Bottom surface of the airfoil

Top surface of the airfoil

Bottom surface of the airfoil

Top surface of the airfoil

Bottom surface of the airfoil

FIGURE 4.19: The unsteady flow (snapshot) for controlled flow with $T = 25$: u component (top left) and v component (top right) fot $t = 490$ and u component (bottom left) and v component (bottom right) for $t = 495$

4.5 Lift Reconstruction

The reconstructed velocity field from subsections 4.4.1 and 4.4.2 is used to reconstruct the lift. Here the lift is reconstructed using the methodology described in 3.1.4 with the grid points representing beginning/end points of panels which leads to 100 panels instead of 400 panels that are used to estimate the lift in the CFD code. This gives very similar results.

The reconstruction error between the lift obtained in the CFD simulation L_{CFD} and lift obtained with the model L_{model} is calculated using

$$\delta L[\%] = \left| \frac{L_{CFD} - L_{model}}{L_{CFD}} \right| \cdot 100\% \tag{4.38}$$

where $|\cdot|$ denotes the absolute value.

The error values do not exceed 2% for the uncontrolled case and 6.5% for the controlled case. The highest reconstruction errors are observed for the controlled flow with a low period of oscillation (up to 12%).

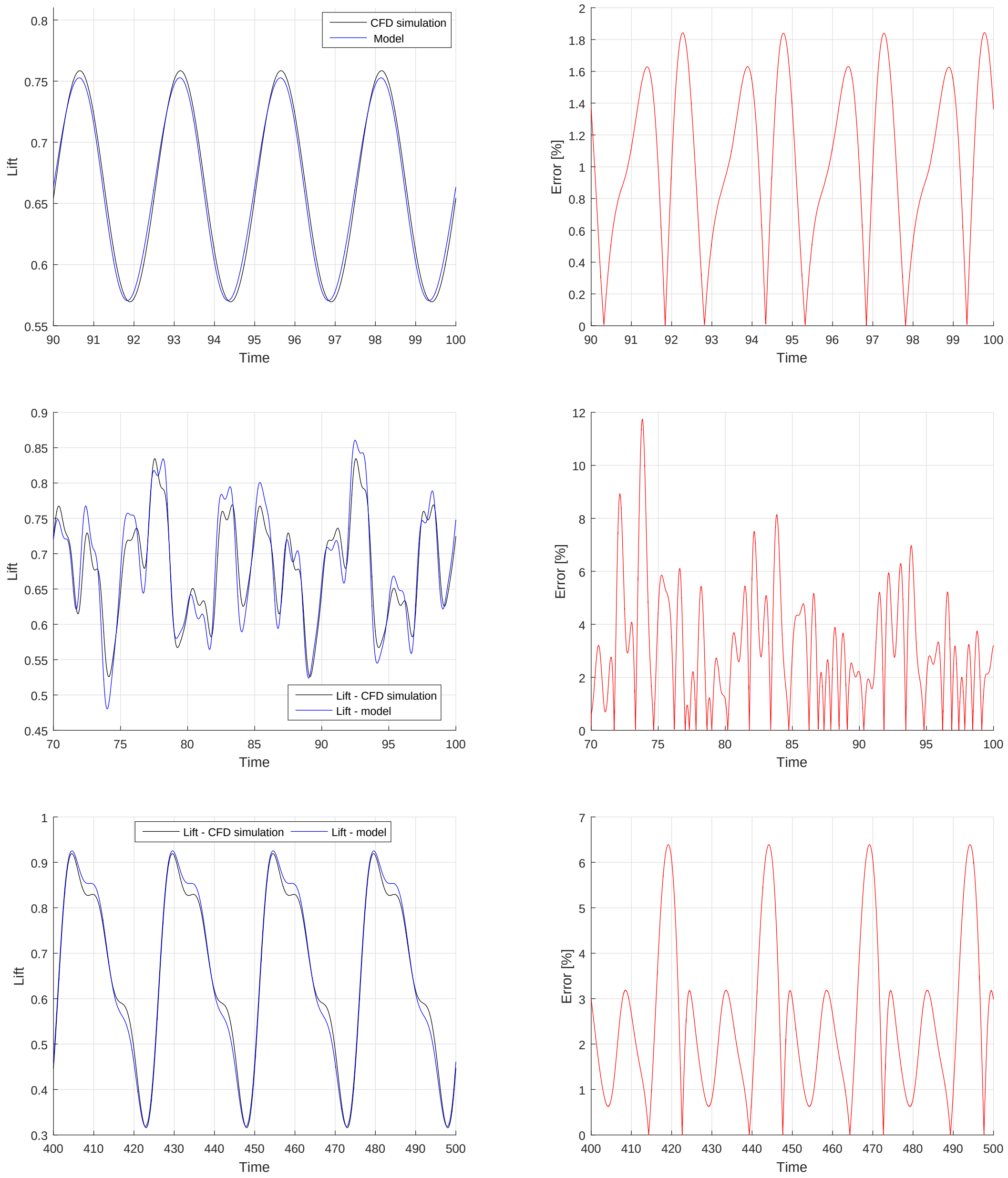


FIGURE 4.20: Lift (left) and reconstruction error [%] (right) for uncontrolled oscillatory flow (top), controlled oscillatory flow with $T = 2.5$ (centre) and controlled oscillatory flow with $T = 25$ (bottom)

4.6 Constructing the State-space Representation

The POD ROM equation state equations given in section 4.3 can be written in a matrix form with the state $a_i(t)$ and the total circulation added acting as the system's input u such that $u = \kappa$

$$\frac{d}{dt}a(t) = A(t)a(t) + B_1(t)u(t) + B_2\frac{d}{dt}u(t) + O(t) \tag{4.39}$$

where matrices A, B_1 and B_2 can be calculated by summing corresponding terms of the state equation (4.32).

The free term $O(t)$ can be removed from the state equation and instead included in the calibration of the output. To eliminate the input derivative on the right-hand side of (4.39) the state is re-defined as $x = a - B_2 u$ to give

$$\frac{d}{dt}x(t) = A(t)x(t) + B(t)u(t) \tag{4.40}$$

where $B(t) = B_1(t) + A(t)B_2$.

In the remainder of the paper the discrete form of the state equation with zero-order hold discretisation at sampling time Δt is used to represent the dynamics. A number of rules of thumb on how to select the sampling time are available in control theory literature. The time constant of the system should be considered when choosing the sample time with one possible choice of sampling at one-tenth of the major time constant of the system (Gopal, 1984) and the absolute minimum of the sampling frequency given by Nyquist sampling theorem (twice the maximum frequency component). In this work, the actuator's delay parameter λ could be considered where the higher the λ constant (faster response) the higher the sampling frequency selection (smaller time step). On the other hand, choosing a very small time step leads to the higher order of the system input-output relation matrix G and hence the larger controller's gain matrix when the lifted representation of the system is used.

To define the output equation, the lift on the blade must be estimated from the velocity field as described in section 3.1.4. The tangential velocity at each point on the surface can be calculated using components of the velocity in x and y direction. In this research the quadratic term of the equation (3.21) is linearized by re-writing the tangential velocity as the sum of its tangential components

$$u_t = u_{rt} + \sum_{j=1}^{N} a_j(t)\phi_{jt} + \kappa u_{ct} \tag{4.41}$$

where u_t is the total tangential velocity, u_{rt} consists the tangential mean and oscillatory parts of the flow, u_{ct} is the tangential component of the unit actuation and approximating the quadratic term as

$$u_t^2 = u_{rt}^2 + 2u_{rt}\left[\sum_{j=1}^{N} a_j(t)\phi_{jt} + \kappa u_{ct}\right] \tag{4.42}$$

To calculate the integral in (3.21) the surface of the airfoil is divided into 100 panels between n and $n+1$ grid points. For each panel the integral is calculated as

$$\frac{\partial}{\partial t}\int u_t ds = \frac{u^k_{t,n+\frac{1}{2}} - u^{k-1}_{t,n+\frac{1}{2}}}{\Delta t} l_n \tag{4.43}$$

where $l = \sqrt{\Delta x^2 + \Delta y^2}$ is the length of the panel and Δt is the time step. The lift at each panel is equal to the y component of the pressure and the total lift is calculated as the sum of lift on all panels.

Finally the matrices for the output equation of the system can be calculated from $\mathbf{u}_m$, $\mathbf{u}_c$ and POD modes. The output equation will contain input and state derivative terms as shown in (4.43).

$$L(t) = C_1(t)a(t) + D_1(t)u(t) + C_2\frac{d}{dt}a(t) + D_2\frac{d}{dt}u(t) + O_2(t) \tag{4.44}$$

For a discrete representation the derivatives are approximated as

$$\frac{d}{dt}a(t) = \frac{a(k) - a(k-1)}{\Delta t}, \quad \frac{d}{dt}u(t) = \frac{u(k) - u(k-1)}{\Delta t} \tag{4.45}$$

where k denotes the discrete time variable. Substituting the state $x = a - B_2 u$ gives

$$L(k) = C(k)x(k) + D(k)u(k) + \hat{C}x(k-1) + \hat{D}u(k-1) + \hat{O}(k) \tag{4.46}$$

where:
$C(k) = C_1(k) + \frac{C_2}{\Delta t}$,
$D(k) = D_1(k) + \frac{D_2}{\Delta t} + C_1(k)B_2 + \frac{C_2 B_2}{\Delta t}$,
$\hat{C} = -\frac{C_2}{\Delta t}$,
$\hat{D} = -\frac{D_2 + C_2 B_2}{\Delta t}$.

In Fig. 4.21 the lift obtained using this matrix representation with the approximation of the quadratic velocity term according to (4.42) is shown compared to the lift estimated without this approximation.

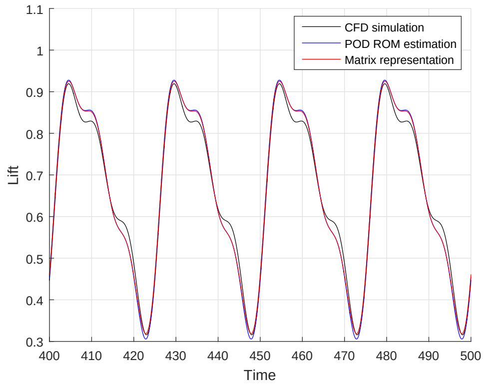


FIGURE 4.21: Lift obtained using the state-space model

4.7 Summary

To design model-based ILC schemes the real-time state solutions are needed. Full Navier-Stokes simulation is computationally expensive and leads to large-scale state equations. In order to reduce a dimension of a model a POD ROM can be used instead. POD eliminates redundant information from the data snapshot but the ability of the modes to approximate the state depends strictly on the data set used to generate the modes. There exist some methods of improving ROM accuracy. In (Fahl, 2000) Trust Region Proper Orthogonal Decomposition (TRPOD) is evaluated. In this approach the low-dimension model is updated during the optimization process. Implementation of this algorithm for modelling of the flow past an airfoil in wind turbines is a possible future work direction.

In this work the POD ROM of the uncontrolled, oscillatory flow is calculated and examined by comparing the velocity field approximated using the model and the velocity field generated in a CFD simulation. Subsequently, the model of the flow with actuation is described and validated in a similar manner. The reconstructed velocity field is used to estimate the lift and obtained results have shown a good accuracy compared to the lift obtained in the CFD simulation. Moreover, the final state-space equation of the system is given in a matrix form suitable for designing ILC schemes. Designing model-based ILC schemes is described in the next chapter.

Chapter 5

Reduction of Lift Fluctuation Using Norm Optimal ILC

In this chapter the model constructed in Chapter 4 is used to design a Norm Optimal Iterative Learning Control (NOILC) scheme for wind turbines application. The design is evaluated in simulation and compared to the model-free ILC.

5.1 Model-based ILC Design for the Flow Model

Norm Optimal ILC is a model-based ILC design starting for linear time-varying (LTV) dynamics from the state-space model (Hakvoort, 2009; Dinh et al., 2014)

$$\begin{aligned} \mathbf{x}(k+1) &= A(k)\mathbf{x}(k) + B(k)\mathbf{u}(k), \quad 0 \leq k \leq \alpha - 1 \\ \mathbf{y}(k) &= C(k)\mathbf{x}(k) + D(k)\mathbf{u}(k) \end{aligned} \tag{5.1}$$

where: $\alpha < \infty$ is the number of samples within a trial (α multiplied by the sampling period gives the trial length or duration), $\mathbf{x} \in \mathbb{R}^n$ is the state vector, $\mathbf{u} \in \mathbb{R}^m$ is the control vector, $\mathbf{y} \in \mathbb{R}^p$ is the output vector and $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$, $D \in \mathbb{R}^{m\times m}$ are the state-space matrices.

5.1.1 Norm Optimal ILC Design

The number of steps within a trial α is finite for the system given by Eq. (5.1) and the supervectors

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}(0) \\ \mathbf{y}(1) \\ \vdots \\ \mathbf{y}(\alpha-1) \end{bmatrix} \quad \mathbf{u} = \begin{bmatrix} \mathbf{u}(0) \\ \mathbf{u}(1) \\ \vdots \\ \mathbf{u}(\alpha-1) \end{bmatrix} \tag{5.2}$$

can be used to represent the dynamics in input-output form

$$\mathbf{y} = G\mathbf{u} + \mathbf{y}_0 \tag{5.3}$$

where $G \in \mathbb{R}^{p\alpha \times m\alpha}$ is given by

$$G = \begin{bmatrix} D(0) & 0 & 0 & 0 & \dots & 0 \\ C(1)B(0) & D(1) & 0 & 0 & \dots & 0 \\ C(2)A(1)B(0) & C(2)B(1) & D(2) & 0 & \dots & 0 \\ C(3)A(2)A(1)B(0) & C(3)A(2)B(1) & C(3)B(2) & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ C(\alpha-1)\prod\limits_{k=\alpha-2}^{1} A(k)B(0) & C(\alpha-1)\prod\limits_{k=\alpha-2}^{2} A(k)B(1) & \dots & \dots & \dots & D(\alpha-1) \end{bmatrix} \tag{5.4}$$

and $\mathbf{y}_0 = \left[C(0), C(1)A(0), C(2)A(1)A(0)..., C(\alpha-1)\prod\limits_{k=\alpha-2}^{0} A(k)\right]^T \mathbf{x}_0$ represents the response to initial conditions which can be absorbed into the reference trajectory and set to zero in equation (5.3) as explained in Dinh et al. (2014).

After trial i NOILC minimises the following performance index to find the input used fot the next trial

$$J(\mathbf{u}_{i+1}) = ||\mathbf{e}_{i+1}||^2_{\mathcal{Y}} + ||\mathbf{u}_{i+1} - \mathbf{u}_i||^2_{\mathcal{U}} \tag{5.5}$$

where $||u||^2_{\mathcal{U}} = \mathbf{u}^T R\mathbf{u}$, $||e||^2_{\mathcal{Y}} = \mathbf{e}^T Q\mathbf{e}$ are norms in the input and output spaces $\mathcal{U}$ and $\mathcal{Y}$, and Q and R are symmetric positive semi-definite, respectively, positive definite weighting matrices. The solution at trial i is

$$\mathbf{u}_{i+1} = \mathbf{u}_i + G^*(I + GG^*)^{-1}\mathbf{e}_i \tag{5.6}$$

where $G^* : \mathcal{Y}^* \to \mathcal{U}^*$ is the adjoint operator of G with respect to the introduced norms given by $G^* = R^{-1}G^TQ$.

5.1.2 Models Used for ILC Design

The state-space representation of the system is used for the ILC design. The state-space matrices can be calculated by adding the corresponding terms from table 4.2, where the mean velocity data is generated using the CFD code and the modes are obtained by applying POD to the snapshot data generated using excitation (4.35) and (4.37) for $T = 2.5$ and $T = 25$ respectively. The amplitude $A = 0.1$ and the time step $\Delta t = 0.005$ are used. No delay in actuation is assumed at first.

Two discrete state-space models are used: the full model (as described in subsection 4.6) of a form

$$\begin{aligned} x(k+1) &= A(k)x(k) + B(k)u(k) \\ L(k) &= C(k)x(k) + D(k)u(k) + \hat{C}x(k-1) + \hat{D}u(k-1) + \hat{O}(k) \end{aligned} \tag{5.7}$$

where derivative terms (4.45) are included and the simplified model of a form

$$\begin{aligned} a(k+1) &= A(k)a(k) + B_1(k)u(k) \\ L(k) &= C_1(k)a(k) + D_1(k)u(k) + O_2(k) \end{aligned} \tag{5.8}$$

where the input and state derivatives are neglected and the x to a relation is $x = a - B_2 u$ as explained in section 4.6. Six modes are used to construct the state-space resulting in a 6×6 matrix A as subsequent modes have small impact on ability to reconstruct the flow and tend to get noisy. Stability of a discrete linear time-invariant system requires that all eigenvalues of the state matrix (A) have modulus strictly less that unity, where in the current case the largest modulus of the eigenvalues of the state matrix at any step k is $|\lambda_{max}| = 0.9998$. In ILC, the trial length is finite and this means that trial-to-trial error convergence can occur even if the state matrix is unstable, as over a finite interval even an unstable system can only produce a bounded output. If performance along the trials is not satisfactory then one option is to first design a stabilizing state feedback (or other form of) control law and then apply ILC design to the resulting dynamics. An alternative is to use an ILC design that simultaneously imposes stability of the state matrix and trial-to-trial error convergence, e.g., designs in the repetitive process setting, see, e.g., Hladowski et al. (2016). In the current case this option was not needed.

The comparison of the lift obtained using full and simplified models for $T = 2.5$ and $T = 25$ is shown in Fig. 5.1 left and right respectively together with the lift obtained in CFD simulation. The error of the lift [%] at each time step was calculated using (4.38)

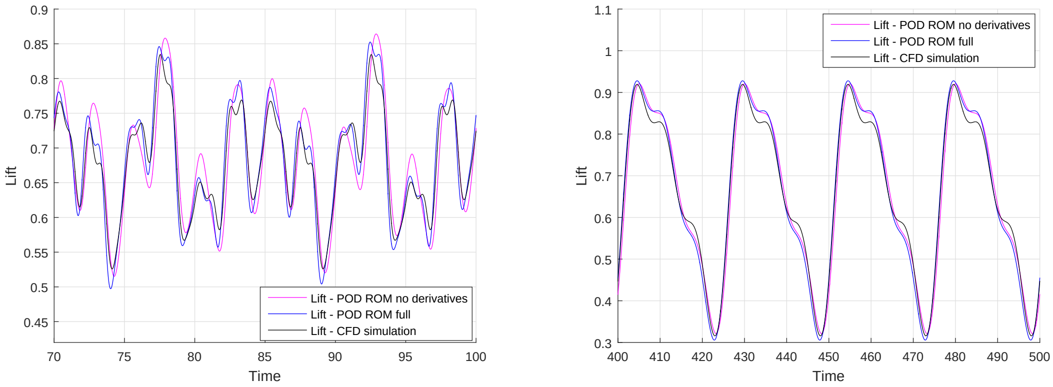


FIGURE 5.1: Comparison of the lift obtained in CFD simulation, lift reconstructed by ROM (full and simplified models) for $T = 2.5$ (left) and $T = 25$ (right)

and the average error is 2.45% for the full model and 3.85% for the simplified model for $T = 2.5$ and 2.89% for the full model and 3.29% for the simplified model for $T = 25$.

ILC can be designed using the simplified model with A, B_1, C_1 and D_1 substituted for A, B, C and D in (5.4). In order to design ILC for the full model of the equation (5.7) which includes the derivative terms the matrix G is modified to

$$G = \begin{bmatrix} D(0) & 0 & \dots & C(0)B(\alpha-1)+\hat{D} \\ C(1)B(0)+\hat{D} & D(1) & \dots & 0 \\ C(2)A(1)B(0)+\hat{C}B(0) & C(2)B(1)+\hat{D} & \dots & 0 \\ C(3)A(2)A(1)B(0)+\hat{C}A(1)B(0) & C(3)A(2)B(1)+\hat{C}B(1) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \big(C(\alpha-1)\prod_{k=\alpha-2}^{1} A(k)+\hat{C}\prod_{k=\alpha-3}^{1} A(k)\big)B(0) & \dots & \dots & D(\alpha-1) \end{bmatrix} \tag{5.9}$$

since matrices $\hat{C}$ and $\hat{D}$ must be included to uphold the input-output relation.

The NOILC controller's gain matrix denoted here as $C_{noilc} = G^*(I+GG^*)^{-1}$ is calculated numerically in Matlab using the discrete system's matrices. The size of the controller's matrix is equal to the number of samples within a trial α and hence depends on the sampling time used in discretisation and on the period of oscillation as $\alpha = \frac{T}{\Delta t}$. This can result in a very large matrix when a larger period of the oscillation is used. If the matrix is too big to provide the efficient computations the sampling time must be increased. The controller's matrix depends only on the system's matrices and hance it can be calculated offline before simulating the system. For the CFD evaluation the controller's gains are calculated beforehand and loaded from a file in a form of an array. The desired value of the input to be applied in the current trial $\mathbf{u}_{i+1} = \mathbf{u}_i + C_{noilc}\mathbf{e}_i$ can be calculated as a whole sequence at the beginning of each trial using the C_{noilc} matrix or the value of the desired input can be calculated at each step by adding the input value from the previous trial to the product of the first row of the gain matrix and the previous trial error vector. The latter approach is used here as in the original CFD code these calculations are done at each step. This turns out to be efficient even for the size of C_{noilc} matrix as large as 5000 ($T = 25$ and $\Delta t = 0.005$), however larger numbers would be difficult to use. In such case, e.g. for the larger period $T = 200$ representing the more realistic case, the sampling time Δt would have to be increased together with the time step used in the CFD simulation. Another possibility is using a smaller number of samples per trial for the construction of the model and the controller and then zero-order holding the controller's gain value in the CFD simulation if the CFD simulation uses the smaller step than the controller.

5.2 Simulation of Norm Optimal ILC Controller Designed for the Simplified Model

In this section the NOILC design is evaluated in simulation for the simplified model in the state-space form at first before applying it in the full CFD test. The case with no actuation delay is considered.

5.2.1 Simulation Results for the Simplified Model in State-space Form

The norm optimal ILC controller for the state-space model without derivative terms is first designed where discrete time varying matrices $A(k)$, $B_1(k)$, $C_1(k)$ and $D_1(k)$ are used for construction of the matrix G in (5.4) and the corresponding NOILC gain matrix $C_{noilc} = G^*(I + GG^*)^{-1}$. Diagonal weighting matrices Q and R are chosen as diagonal matrices with the unit gain on the main diagonal at first.

Fig. 5.2 (left) gives the lift obtained in Matlab simulation for this design. The target value for the lift assuming no oscillatory component is $L_{tar} = 0.66$. For this design, the lift converges to the desired value over a trial reaching the error $2-$norm and $\infty-$ norm values smaller than 10^{-3} in less than 10 trials and 10^{-6} in less than 100 trials (Fig. 5.2, right).

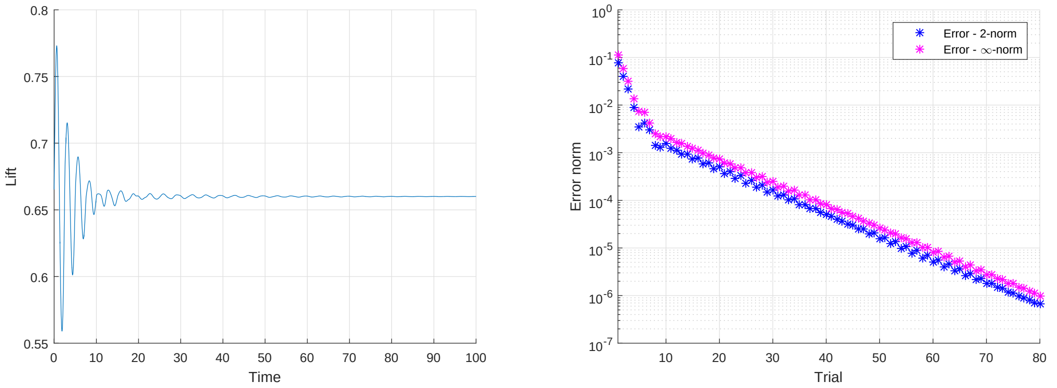


FIGURE 5.2: The lift (left) and the error norms (right) obtained in Matlab simulation for the system with NOILC designed using the simplified model (neglecting derivative terms)

The speed of convergence and the robustness of the system can be adjusted by changing the diagonal values of the matrices Q and R denoted here as q and r respectively. Increasing q value produces a faster convergence but allows higher changes in the input signal. Conversely, increasing r values produce a slower convergence but better robustness. The convergence of the error for various q and r values is given in Fig. 5.3.

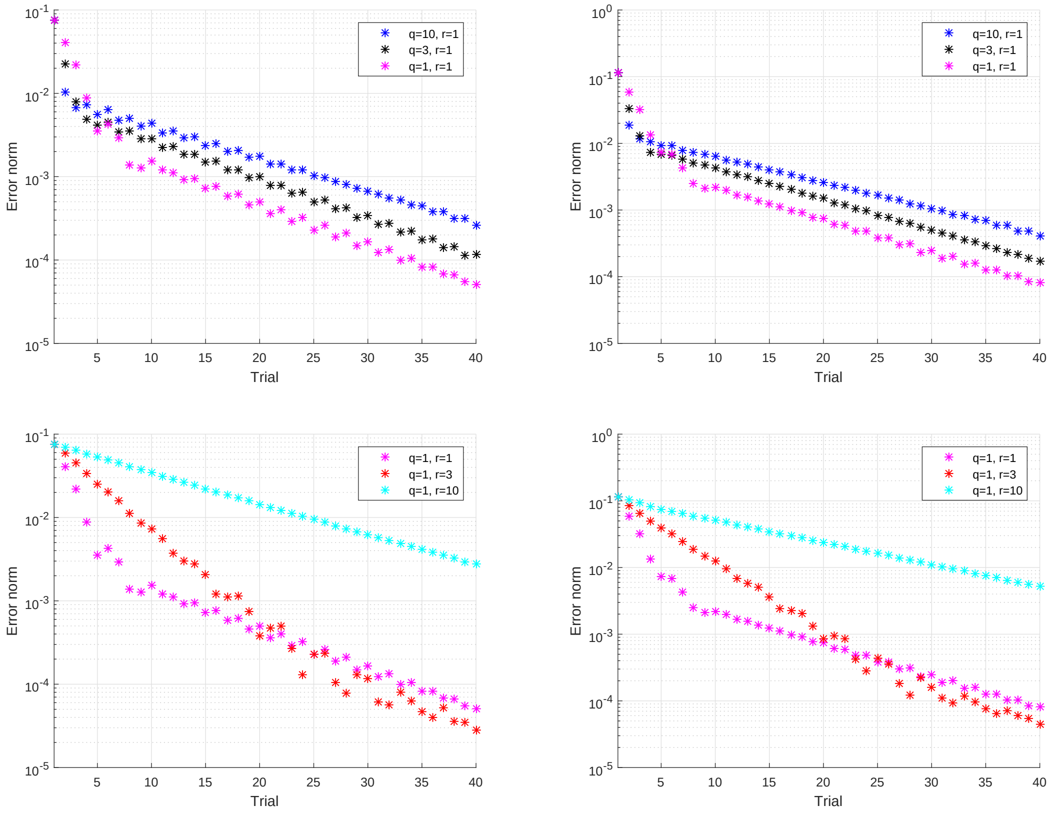


FIGURE 5.3: The error 2-norm (left) and ∞-norm (right) for various q with constant r (top) and various r and constant q (bottom) - simplified design

From Fig. 5.3 it can be observed that the higher value of the parameter q produces faster convergence in first few trials but the error norm in latter trials is actually higher than for controllers with the lower value of q. This can be seen in Fig. 5.4 where the lift in the initial trials (left) and in latter trials (right) is shown for the controller with $q = 10$ and $r = 1$ and the controller with $q = 1$ and $r = 1$.

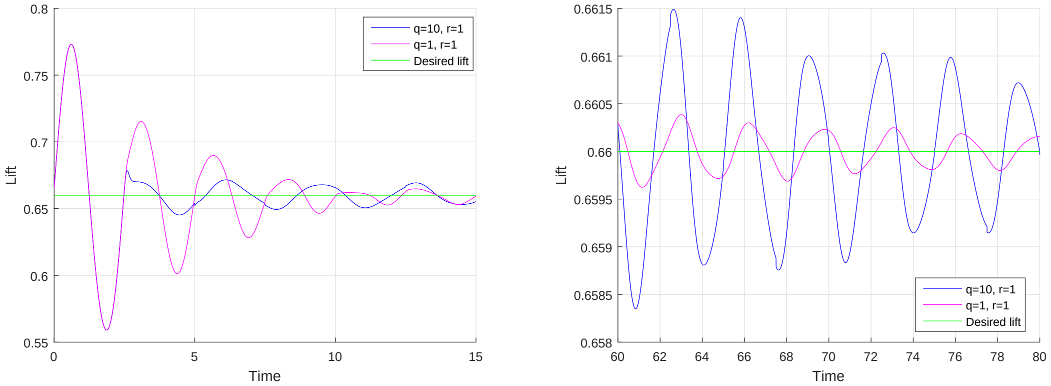


FIGURE 5.4: Comparison of the lift in initial trials (left) and latter trials (right) obtained with different values of q - simplified design

5.2.2 Evaluation of the Simplified Design in CFD Test

Results obtained in the previous subsection look very promising. The error achieved with NOILC controller in Matlab simulation converges to zero unlike for the model-free ILC design where the maximum of 30% reduction was obtained for the pure ILC (without the P controller). Moreover, the error norms reach lower values than the combination of model-free ILC and the proportional controller. However, the same controller applied to the full CFD simulation does not provide convergence and leads to increase in error as shown in Fig. 5.5 (example for the controller with $q = 10$ and $r = 1$).

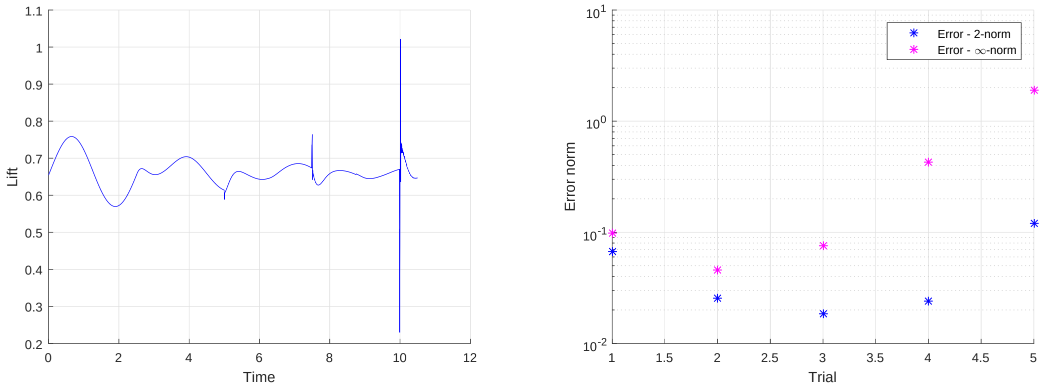


FIGURE 5.5: Lift (left) and error norms (right) obtained in the full CFD simulation for the controller designed neglecting derivative terms ($q = 10$, $r = 1$) - simplified design

The error norms decrease in initial trials but then the peaks in the lift appear for the first few time instances of the subsequent trials due to the high input derivative and therefore velocity derivative. Both norms start to increase already in the 3rd trial. This cannot be avoided by adjusting the ILC weighting matrices as shown in Fig. 5.6 where the controller with $q = 1$ and $r = 3$ is evaluated.

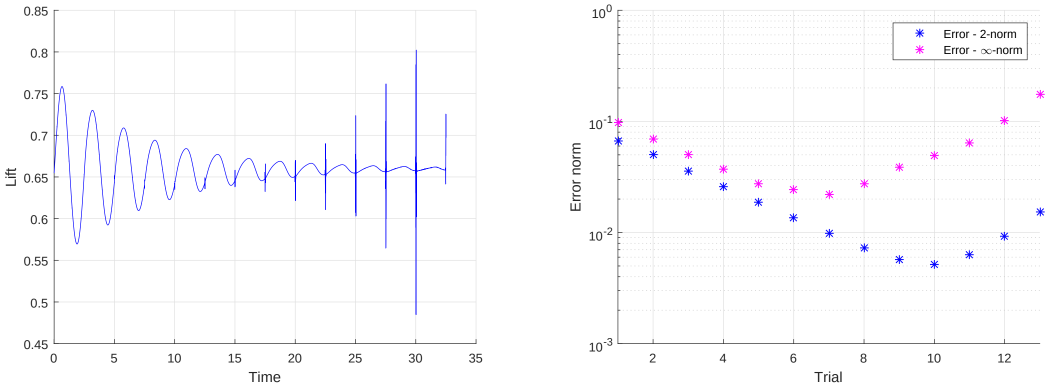


FIGURE 5.6: Lift (left) and error norms (right) obtained in the full CFD simulation for the controller designed neglecting derivative terms ($q = 1$, $r = 3$) - simplified design

With such gain selection the peaks in lift are delayed compared to the previous case and the 2-norm converges for first 10 trials and ∞-norm for first 6 trials before they increase.

The reason for the peaks is the fact that the simplified model does not consider the unsteady part of the lift coming from the derivative in velocity. Given this conclusion, a new NOILC controller is designed for the full model where the derivatives of the input and state are taken into consideration.

5.3 Simulation of Norm Optimal ILC Controller Designed for the Full Model

In this section the NOILC controller is designed using the full model. The input-output relation of the system is calculated recursively for each step in time k and the new matrix G including derivative terms $\hat{C}$ and $\hat{D}$ is calculated according to (5.9). The lift is now calculated using (5.7) and it includes the terms coming from the derivatives of state and input. Similarily like the previous design this new design is evaluated in Matlab simulation for the state-space model first before applying it to the full CFD test. Simulations for the model are faster and enable to perform more experiments with variety of controller gains.

5.3.1 Simulation Results for the Full Model in State-space Form

Similar tests to those performed for the simplified model have been carried out for the full model. The NOILC controllers with various q and r parameters have been tested. The lift and the error norms for the controller with $q = 1$ and $r = 1$ is shown in Fig. 5.7 and the comparison of the 2-norm and ∞-norm for various weighting matrices is shown in Fig. 5.8.

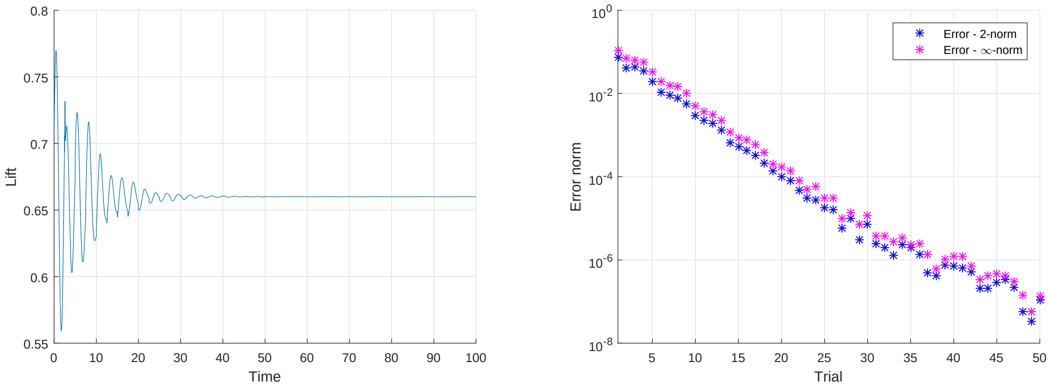


FIGURE 5.7: The lift and the error norms obtained for the system with NOILC designed using the full model (including derivative terms)

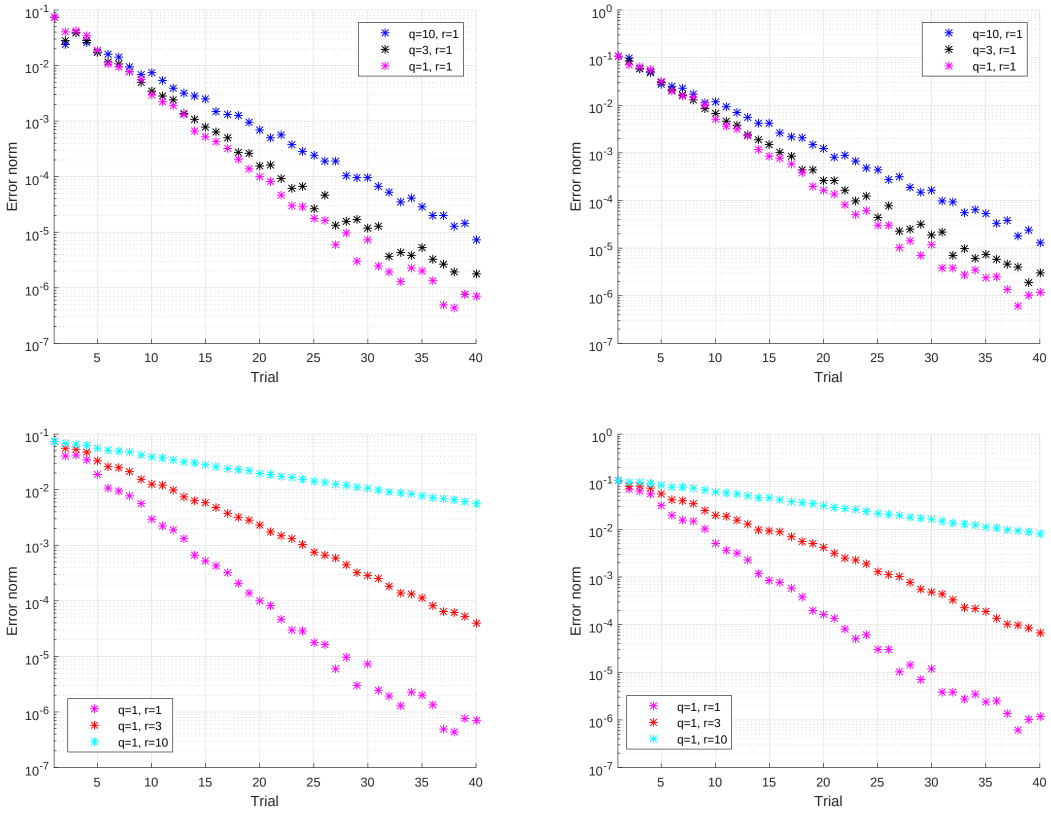


FIGURE 5.8: The error 2-norm (left) and ∞-norm (right) for the various q parameter with the constant r (top) and the various r parameter and the constant q (bottom) - full design

It is important to also check the input and the input derivative which correspond to the total vorticity added and the strength of the vortex added at a particular time step. For this design the input and its derivative is much smoother than for the simplified design as shown in Fig. 5.9 (controller with $q = 1$ and $r = 3$) and hence the controller can be now evaluated in the full CFD test.

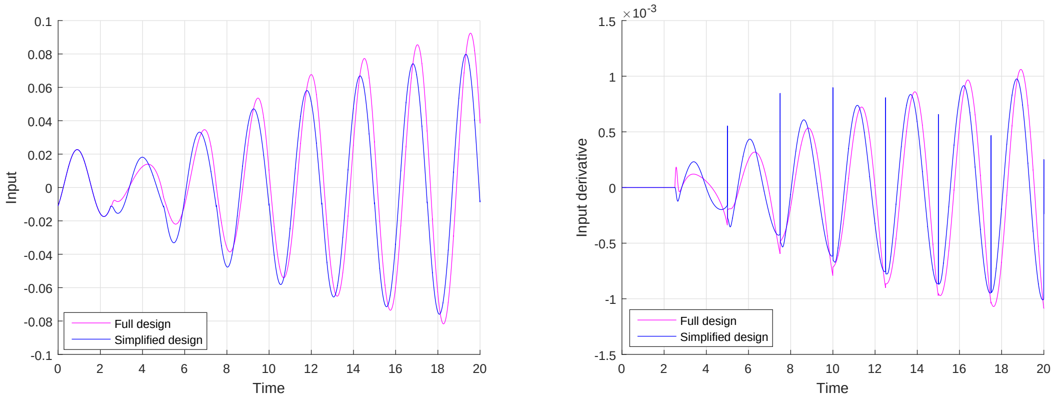


FIGURE 5.9: Comparison of the input (left) and its derivative (right) for the full and the simplified design

5.3.2 Evaluation of the Full Design in CFD Test

The full design has been evaluated in the CFD test. The lift obtained for the controller with $q = 1$ and $r = 1$ and for the controller with $q = 1$ and $r = 3$ is shown in Fig. 5.10 (top left) and the error norms in Fig. 5.10 (top right). The input and the input derivative are given in Fig. 5.10 (bottom).

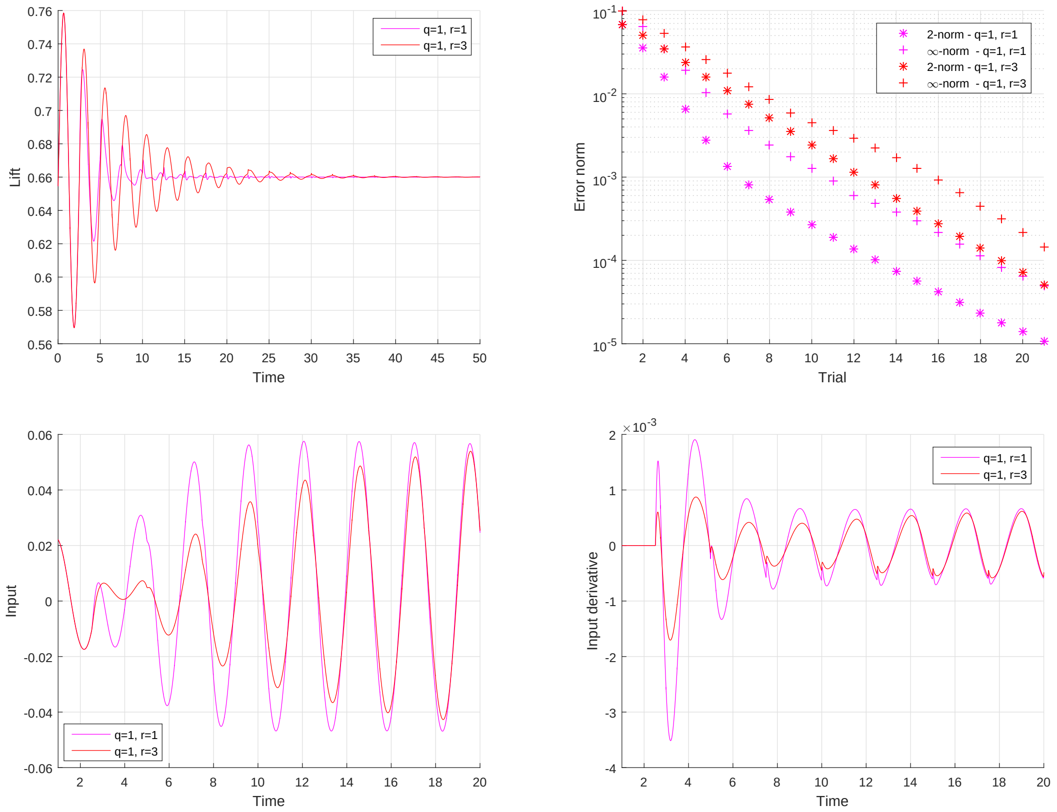


FIGURE 5.10: The lift (top left), the error norms (top right), the input (bottom left) and its derivative (bottom right) obtained for the system with NOILC designed using the full model - CFD evaluation

Unlike the simplified NOILC design this full design works very well giving the convergence similar to the one obtained in the Matlab simulation. The input and its derivative is smoother than for the simplified design and no noise in initial time instances for each trial can be observed. What is more, no performance degradation in time is observed when the long simulation is run. The comparison of the error norms obtained in the Matlab simulation and in the CFD evaluation is given in Fig. 5.11. In general, very similar trends are observed for both simulations. Slightly different convergence rate is achieved in the CFD test where the convergence is actually faster than in Matlab simulation. However, the lower final error norm is achieved in Matlab simulation where the norm continues to converge even in latter trials. For the CFD test the error stops to converge after 25 trials.

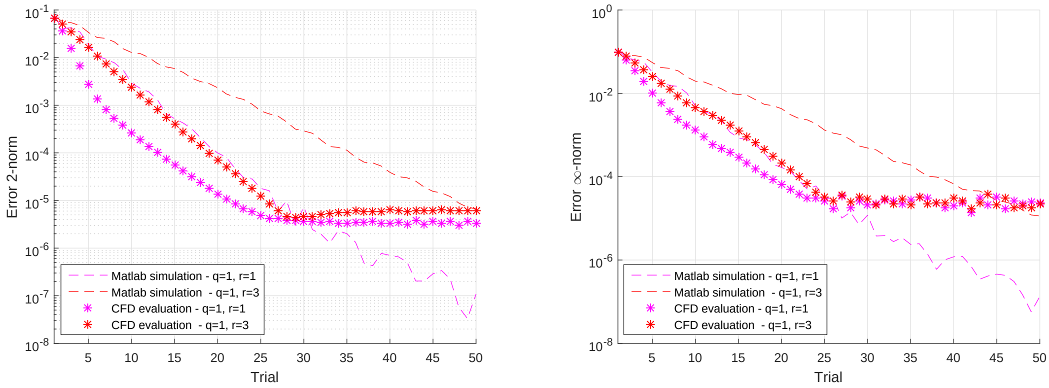


FIGURE 5.11: Comparison of the 2-norm (left) and ∞-norm (right) obtained in Matlab simulation for the model in state-space form and in full CFD evaluation

The vortex street obtained with these two controllers is given in Fig. 5.12. The influence of the control action is clearly visible after four full cycles (when the control is switched on). Fig. 5.12 indicates that once the control is switched on the wake still has a regular structure, more like that for the uncontrolled flow (Fig. 4.7) than that for the case with the forcing (4.35) used to develop the POD model (Fig. 4.12). This suggests that a significant effect of the control is to alter the phase of the vortex shedding at the trailing edge to counteract the effect of the oscillation in the freestream; see in particular the change in the wavelength of the flow when the control is turned on (around $x = 20$ in Fig. 5.12).

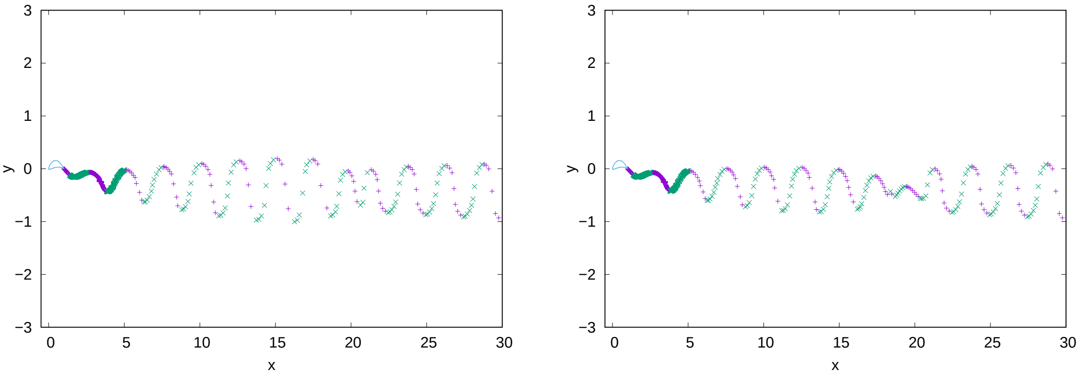


FIGURE 5.12: The wake for the system controlled by NOILC: the controller with $q = 1$ and $r = 1$ (left) and the controller $q = 1$ and $r = 3$ (right)

Fig. 5.13 shows the pressure on the surface of the airfoil for the no control case (left) and for the ILC controlled system once the control has converged. The flow with $T = 2.5$ and $A = 0.1$ is simulated and the pressure at the maximum freestream velocity $V_{0x} = 1.1$ (green) and the minimum freestream velocity $V_{0x} = 0.9$ (blue) is shown compared to the pressure for the steady flow past the body. In these figures the lift is represented by the area between the upper and lower parts of the curves (the lower and upper surfaces of the airfoil). For the uncontrolled flow (left), the difference in the curves clearly shows the change in the lift as the flow oscillates. For the controlled case, there are large changes near the trailling edge of the airfoil, forced by actuation. However, for most of the body the pressure distribution is close to that for steady pressure. Hence, from the wake and pressure plots we see that the effect of the contol is to change the phase of the shedding so that the flow over the body resembles that for steady flow at mean flow conditions.

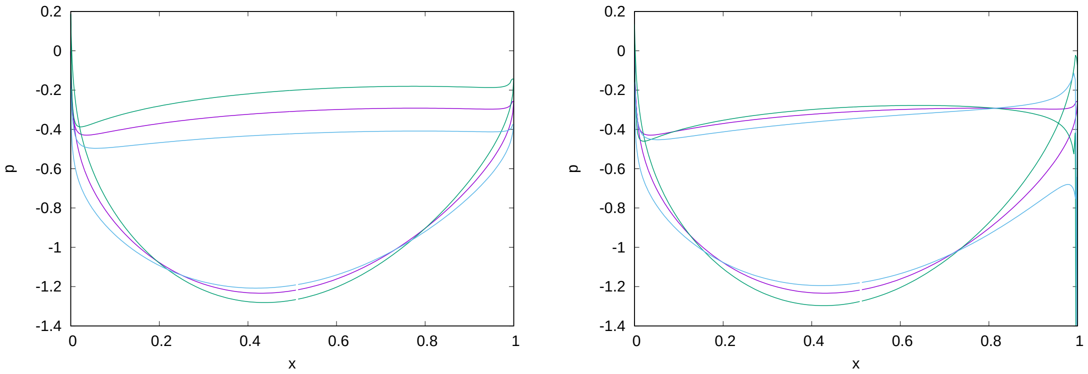


FIGURE 5.13: The pressure on the surface of the airfoil for the no control case (left) and for the controlled case once the control has converged (right): purple - no oscillation, green - maximum freestream velocity ($V_{0x} = 1.1$), blue - minimum freestream velocity ($V_{0x} = 0.9$)

5.3.3 Comparison with the Model-free Designs

The model-based design is compared with the model-free design from section 3.2.1. Four designs are compared:

- model-free ILC with $\mu_1 = 0.1$,
- P controller with $\mu_0 = 50$,
- combination of P and model-free ILC ($\mu_0 = 50$ and $\mu_1 = 1$),
- NOILC with diagonal weighting matrices with $q = 1$ and $r = 1$ values on the diagonal.

The error norms are given in Fig. 5.14. The fastest convergence is achieved with the combination of P and model-free ILC with the 2-norm reaching approximately 10^{-5} in 7 trials and $4.4 \cdot 10^{-6}$ after 50 trials. Very good performance is obtained using the model-based controller. The convergence is slower than for the combination of P and model-free ILC but after 25 trials even lower error 2-norm is obtained reaching $3.3 \cdot 10^{-6}$ in 50th trial. The ∞-norm after 50 trials reaches $2.3 \cdot 10^{-5}$ for NOILC and $3.1 \cdot 10^{-5}$ for the combination of P and model-free ILC. P controller provides a good convergence in the first few trials but it cannot reject the periodic disturbance in the error fully and leaves the oscillatory component in the lift in latter trials. The poorest performance is obtained using model-free ILC which gives approximately 30% error norm reduction compared to the no control case. The values of the error norms after 50 trials achieved with four different controllers and with no control action are given in Table. 5.1.

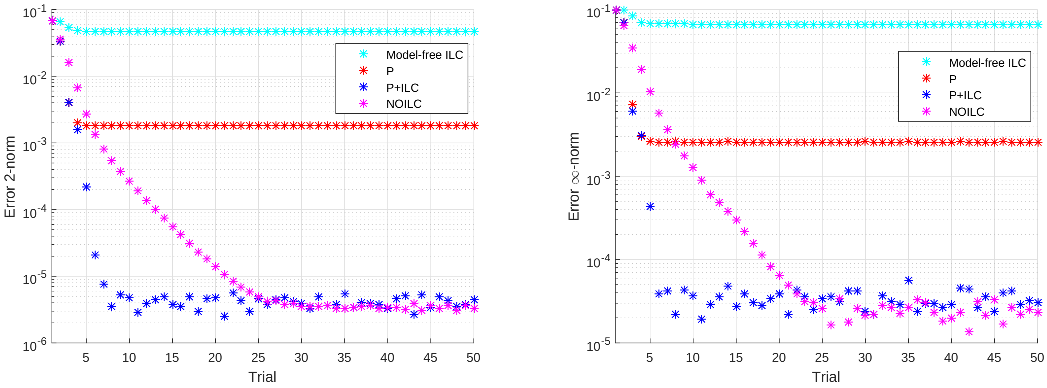


Figure 5.14: Comparison of the 2-norm (left) and ∞-norm (right) for various controller types

Table 5.1: Error norms after 50 trials

Controller type	2-norm	∞-norm
no control	$6.7 \cdot 10^{-2}$	$9.8 \cdot 10^{-2}$
model-free ILC	$4.7 \cdot 10^{-2}$	$6.6 \cdot 10^{-2}$
P	$1.8 \cdot 10^{-3}$	$2.6 \cdot 10^{-3}$
P+ILC	$4.4 \cdot 10^{-6}$	$3.1 \cdot 10^{-5}$
NOILC	$3.3 \cdot 10^{-6}$	$2.3 \cdot 10^{-5}$

5.4 Evaluation of the Design Under Various Operating Conditions

The design from section 5.3 is evaluated for various conditions including the non-deterministic flow, higher period of oscillation and actuation delay. For the simulation of the non-deterministic flow the upstream vortices are introduced where the strengths are selected such that they have a substancial effect on the lift as they pass the wing, similarly to the tests for the model-free design. Adding upstream vortices together with using the large amplitude of oscillation and the small period represent difficult conditions. Moreover, larger delay coefficients than the ones that characterise the state-of-the-art actuator are used when simulating the system with the delay. However, in practice the angle of the attack would vary as well and some variations in the period of oscillation could appear. Some of these aspects have been addressed in Blackwell et al.

(2016) where the model-free design was evaluated for various conditions. Here, these aspects would have to be addressed when modelling the system's behaviour involving more complex modelling and hence this remains the direction for future research. The problem of using the varying period of rotation has been discussed in Rogers and Tutty (2016), where the variable periods are considered by using a constant increment in angle of the blade and varying the time step. Finally, the future analysis should address any other limitations of sensors and actuators. For instance, using a limited number of pressure sensors could be examined similarly to Blackwell et al. (2016), where such analysis for the model-free design are performed.

5.4.1 Non-deterministic Flow

Two NOILC designs (the controller with $q = 1$ and $r = 1$ and with $q = 1$ and $r = 3$) are evaluated in the CFD test with additional vortices introduced. In the first test two upstream vortices are introduced at $x_{v1} = (-70, -0.35)$ and $x_{v2} = -80, 0.25)$ at the start of the simulation, both with strength $\Gamma = 0.1$, and hence vortex 1 will pass below the airfoil at approximately $t = 70$ and vortex 2 above the airfoil at approximately $t = 80$. Both NOILC designs survive this robustness test: the peaks in the lift appear when the vortices pass the airfoil but they are attenuated and after several trials the ∞-norm reduces back to the values below 10^{-4}. However, the design is less robust than the combination with the P controller, where the peaks due to upstream vortices are much lower. The lift and the ∞-norm are shown in Fig. 5.15.

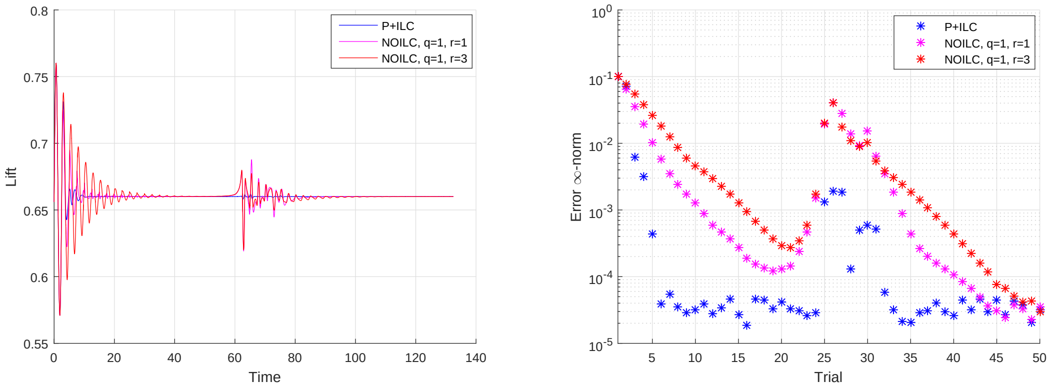


FIGURE 5.15: Comparison of the lift (left) and the error ∞-norm (right) obtained for NOILC and P+ILC (non-deterministic flow, case 1)

In second robustness test (case 2) three vortices with higher strengths of $\Gamma_1 = 0.2$, $\Gamma_2 = 0.1$ and $\Gamma_3 = 0.3$ are introduced at $x_{v1} = (-50, -0.15)$, $x_{v2} = (-60, 0.25)$ and $x_{v3} = (-70, -0.1)$ respectively. The third, strongest vortex, will also pass closer to the body.

In such severe flow conditions the significant lift disturbances are observed when the vortices pass the airfoil, but afterwards the lift stabilizes within the next ten cycles reaching the maximum trial error values smaller than 10^{-4}. The peaks in lift are higher than these observed in the previous robustness where vortices with lower strengths have been introduced. Also, compared to the combination of P and model-free ILC the NOILC controller is less robust as shown in Fig. 5.16 where the lift and the ∞-norm are shown. The peaks in lift are higher for NOILC than for the combination of P and model-free ILC action and more trials are required to achieve the target lift value again after the vortices are gone. Summarising, again the combination of P and ILC provides the best robustness in presence of disturbances.

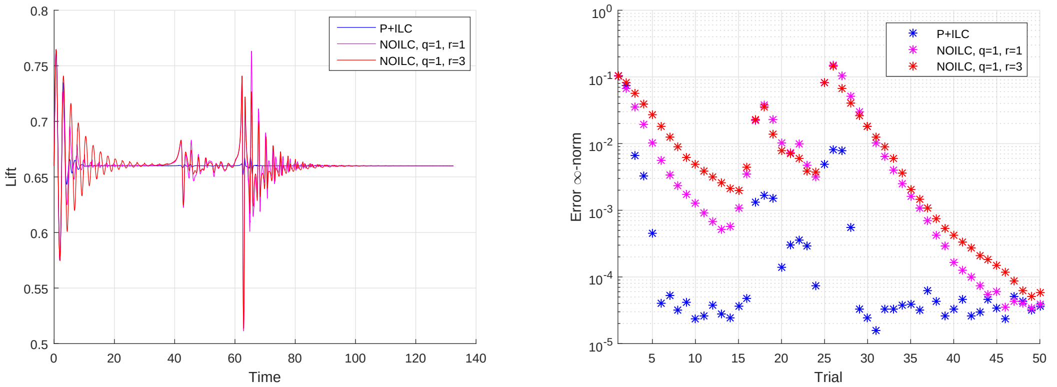


FIGURE 5.16: Comparison of the lift (left) and the error ∞-norm (right) obtained for NOILC and P+ILC (non-deterministic flow, case 2)

However, NOILC design can still provide a significantly better performance than model-free ILC without the P action or no control system as shown in Fig. 5.17 where the lift and the error ∞-norm for NOILC are compared to the no control case and to the pure model-free ILC with $\mu_1 = 0.1$ gain.

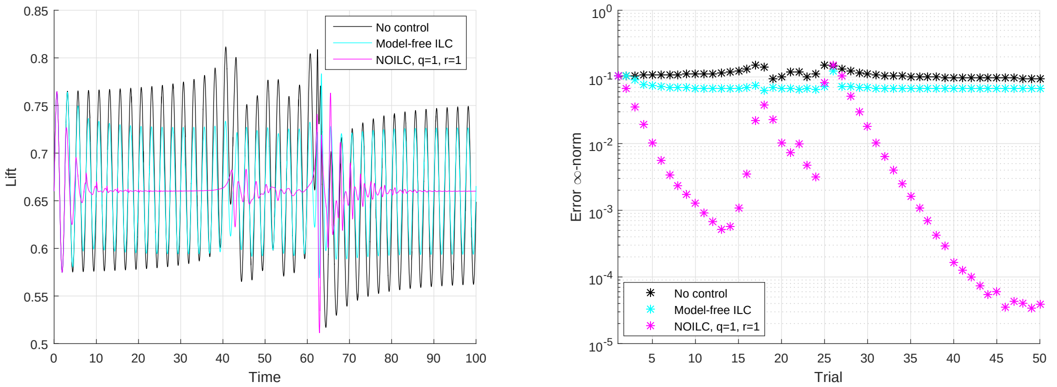


FIGURE 5.17: Comparison of the lift (left) and the error ∞-norm (right) obtained for NOILC, model-free ILC and the no control case (non-deterministic flow, case 2)

The disturbances that appear in the lift when the vortices pass the airfoil are significant, yet slightly smaller than for the no control case and the model-free ILC. After the vortices pass the airfoil the lift converges back to the target value for NOILC, whereas no control and model-free ILC leave significant periodic oscillations in the lift that cannot be damped.

5.4.2 Higher Period of Rotation

The NOILC is also evaluated for the higher period of oscillation $T = 25$. For this case the NOILC scheme is designed using the model given it section 4.4.2. The controllers are designed using the full model including derivative terms (5.7) as the simplified design does not work, similarly as for the lower oscillation period $T = 2.5$. In reality even larger value of the non-dimensional period should be chosen with $T = 200$ providing a realistic representation of a rotating turbine. Here, such system could be simulated for the reduced order model, but the CFD model would take much longer, limiting the number of experiments for each controller. As this work presents the very first results on the ILC schemes designed using ROM the evaluation of the results in CFD is very important and hence choosing the value $T = 25$ provides a trade off between the reality and the computational effort. Moreover, as explained previously in section 3.2.1, a shorter period requires a more rapid response from the controller, and therefore a stiffer test of the control methodology in general.

Having the state-space representation allows testing various controllers in simulation using Matlab within minutes for this higher period of oscillation which is a massive advantage compared to testing each design in CFD simulation which takes several days. Here three controllers with weighting parameters: $q = 3$ and $r = 1$, $q = 1$ and $r = 1$, $q = 1$ and $r = 3$ are tested in Matlab. Good error convergence is obtained for first two designs and slower convergence for the third design where the r parameter is higher. In Fig. 5.18 the lift (top left) and the error norms (bottom) are shown. Conversely, smoother input and its derivative are obtained for the third controller (top right). For the controller with $r = 1$ and $q = 1$ the 2-norm converges from $1.1 \cdot 10^{-1}$ in the first trial to $7 \cdot 10^{-5}$ in 15th trial and the ∞-norm is approximately 30% higher.

The evaluation in CFD test is also performed. A single CFD test takes approximately 50h to simulate the operation over 20 cycles for this case. Therefore, only the controller with $q = 1$ and $r = 1$ is evaluated. These weighting parameters have been chosen as they produce good convergence and make the system more robust. The CFD results are similar to these obtained in the Matlab simulation with the error converging to values lower than 10^{-4} (∞-norm) and 10^{-5} (2-norm) in approximately 20 trials. The lift and the error norms obtained in CFD evaluation are given in Fig. 5.19.

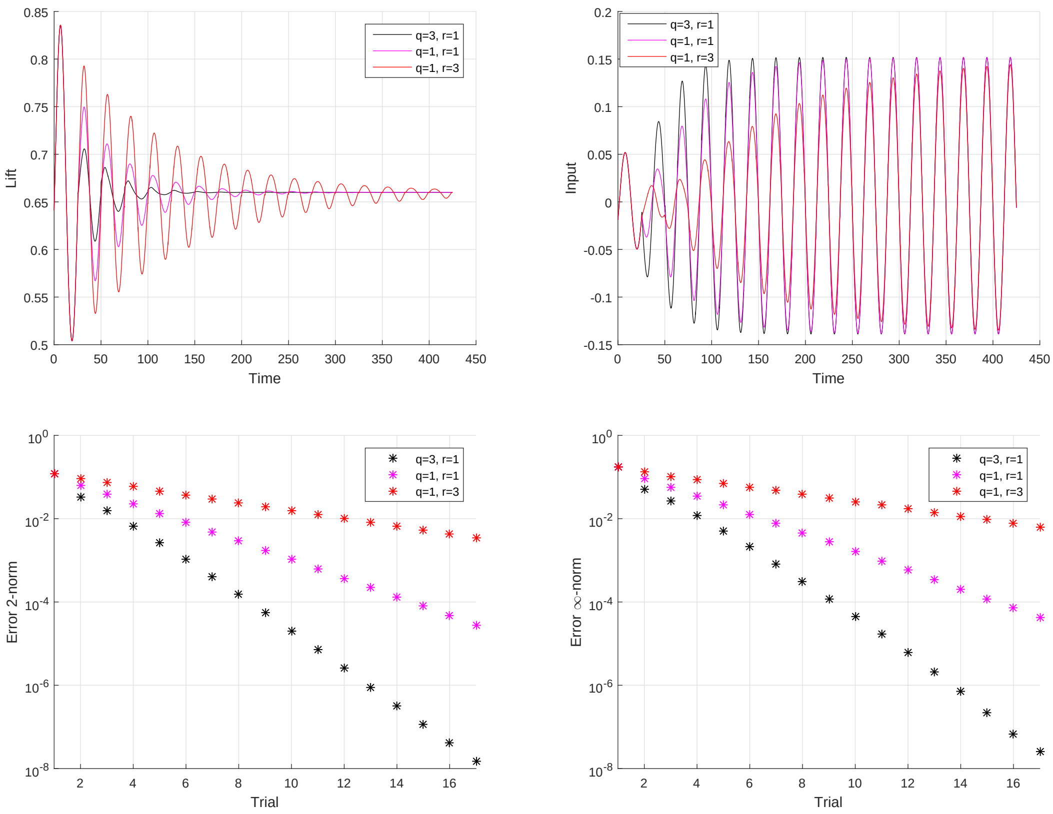


Figure 5.18: The lift (top left), the input (top right) and the error norms (bottom) obtained in Matlab simulation for the system with higher period of oscillation $T = 25$ controlled by NOILC with various weights

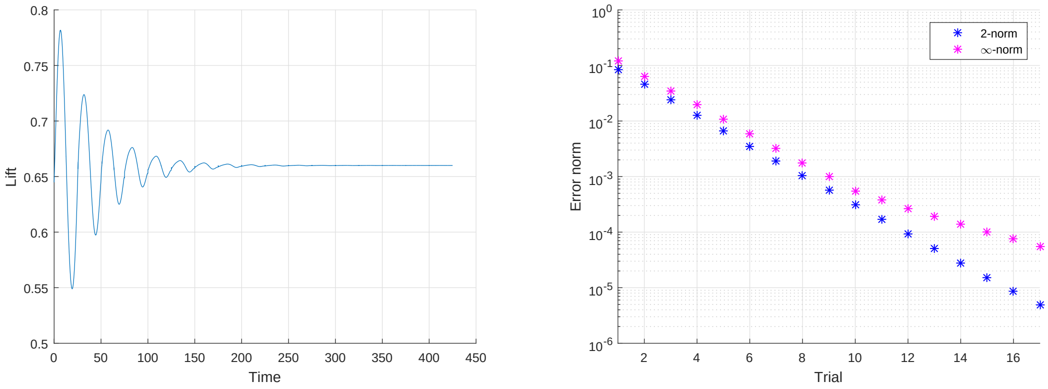


Figure 5.19: The lift (left) and the error norms (right) obtained for the system with period of oscillation $T = 25$ controlled by NOILC ($q = 1$, $r = 1$) - CFD evaluation

The comparison of results obtained in the Matlab simulation for the model and the CFD evaluation for this design is given in Fig. 5.20. A good agreement between the results is shown. The phase of both systems is very similar although the amplitude of the error (and hence the lift) is slightly higher for the simulation of the ROM than for CFD simulation in the first couple trials. The reason is that the control is switched on slowly

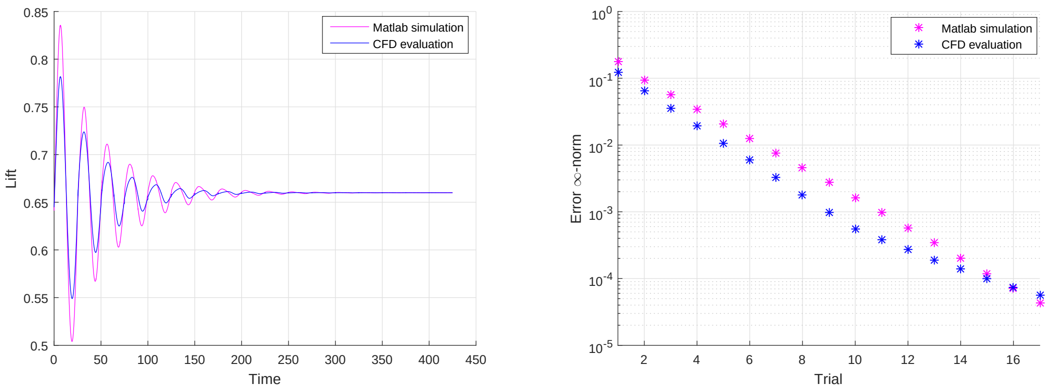


FIGURE 5.20: Comparison of the lift (left) and the ∞-error norm (right) obtained in Matlab simulation for the model and CFD evaluation for the system with higher period of oscillation ($T = 25$) controlled by NOILC with $q = 1$ and $r = 1$

after the first trial as described in section 3.1.5 therefore the first trial represents a no control case and in the second trial the control is switched on slowly. The ROM has been derived using snapshots from the simulation of the fully controlled system and hence it can give a higher mismatch when reconstructing the no control case. However, still a good match between the system's behaviour is obtained as the periodic component of the lift goes to zero in a similar manner for both simulations and is almost completely damped within $10 - 15$ trials reaching 10^{-4} ∞-norm in 15th trial.

5.4.3 Sectional Lift Adjustment

ILC could be used to obtain other control objectives. For instance, it could be used for sectional lift adjustment (see section 2.2.2), i.e. to increase the blade lift at low wind speeds and hence allowing earlier cut-in as shown in Fig. 5.21. Here, the CFD test for

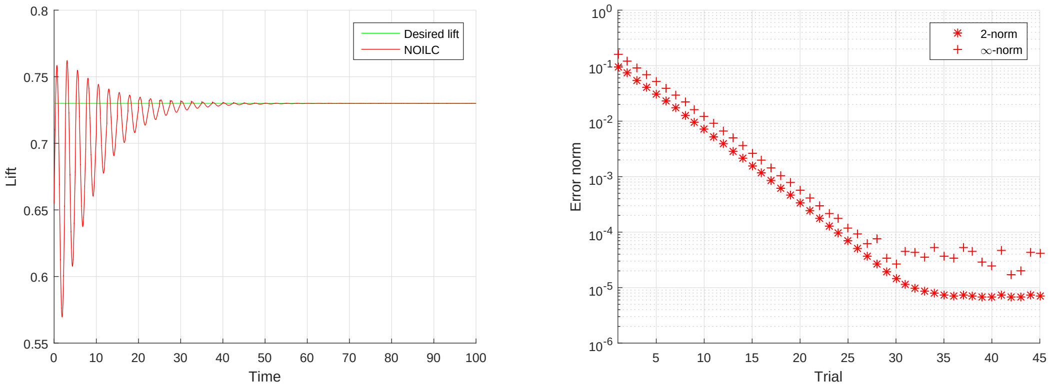


FIGURE 5.21: The lift (left) and the error norms (right) obtained for the different target value of the lift - CFD evaluation

the NOILC controller with $q = 1$ and $r = 3$ has been run for a different value of the target lift $L_{tar} = 0.73$ representing a 10% increase in the lift (oscillation period $T = 2.5$). The results of this test show that it is possible to obtain a different target lift value of the lift with the NOILC approach.

5.4.4 Delay in Actuation

The design has been also run with the delay in actuation $\lambda = 10$. For this case unacceptable performance is obtained as shown in Fig. 5.22 (bottom). The design only works for smaller delays of $\lambda > 100$ (Fig. 5.22, top). For larger delays the error norm reduces initially but then significant oscillations appear in the lift and are attenuated from trial to trial resulting in instability. The results for $q = 1$ and $r = 3$ are shown here but tuning the weights does not eliminate this behaviour.

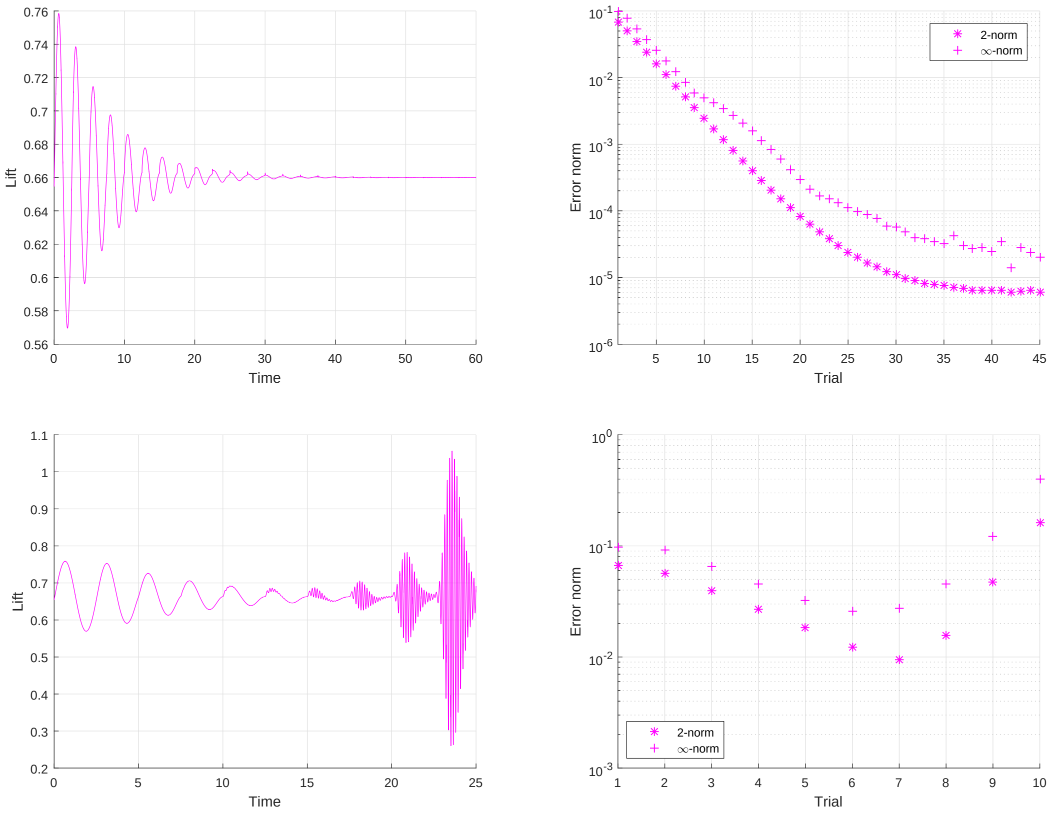


FIGURE 5.22: The lift (left) and the error norms (right) obtained for the system with the actuation delay of $\lambda = 200$ (top) and $\lambda = 10$ (bottom) - CFD evaluation

To obtain satisfactory performance in presence of the larger delay the actuator dynamics should be considered in the modelling process. One way of addressing this problem is including the actuation delay (3.41) in the state-space equations. For this delayed system the actual input $\hat{u}$ acts like the input to the system, hence the continous state space equations (4.39) and (4.44) become

$$\begin{aligned} \frac{d}{dt}a(t) &= A(t)a(t) + B_1\hat{u}(t) + B_2\frac{d}{dt}\hat{u}(t) + O(t) \\ L(t) &= C_1(t)a(t) + D_1(t)\hat{u}(t) + C_2\frac{d}{dt}a(t) + D_2\frac{d}{dt}\hat{u}(t) + O_2(t) \end{aligned} \tag{5.10}$$

The actuation delay (3.41) can be included by redefining the state vector as $x(t) = [a(t) \quad \frac{d}{dt}\hat{u}(t) - \lambda u(t)]^T$, where $\hat{u}$ denotes the actual input and u the desired input calculated by the controller. This produces the continous system of equations with the actual input inside the state vector x and the input calculated by the controller u acting as the input to the system

$$\begin{aligned} \frac{d}{dt}x(t) &= \begin{bmatrix} A(t) & B_2 - \frac{B_1(t)}{\lambda} \\ 0 & -\lambda \end{bmatrix} x(t) + \begin{bmatrix} B_2\lambda \\ -\lambda^2 \end{bmatrix} u(t) \\ L(t) &= \begin{bmatrix} C_1(t) & D_2 - \dfrac{D_1(t)}{\lambda} \end{bmatrix} x(t) + D_2\lambda u(t) + \begin{bmatrix} C_2 & 0 \end{bmatrix} \frac{d}{dt}x(t) + \hat{O}(t) \end{aligned} \tag{5.11}$$

The effect of the free terms $\hat{O}$ can be absorbed to the reference trajectory and after discretisation a new NOILC scheme can be designed using these updated equations. Implementation and evaluation of this new control system is the future work direction.

5.5 Summary

Previous research described in Chapter 3 had used simple structure model-free designs to demonstrate that ILC can be applied to aerodynamic load reduction in wind turbines by the use of smart rotors. Such controllers are limited due to their structure and an obvious next step is to consider model based design. In this chapter, the norm optimal ILC scheme is designed for a reduced order model of the flow past an airfoil. The model derived in Chapter 4 has been used to design NOILC schemes in two cases. Firstly, by further simplifying the model through neglecting the terms including state and input derivatives and it was demonstrated that satisfactory control action is possible by this general approach only in simulation for this simplified model. To examine the effects of the additional simplification, the design was run against the CFD simulation, resulting in instability. Completing the ILC design for the model with derivative terms included showed that the error convergence is possible both in simulation for the state-space representation and full CFD test.

Designed controllers have been compared with the model-free controllers and evaluated under various operating conditions. The NOILC design gives significantly better results than the model-free ILC on its own (without the P controller) and due to the significantly lower computational effort a variety of controllers can be simulated after obtaining the state-space model, e.g. for higher periods of oscillation. However, in tests with non-deterministic flow included the best robustness is obtained for combination of the model-free ILC with the P controller. The reason is that the pure feedforward ILC controller can only reject repeating disturbances and hence it cannot provide the perfect tracking in presence of nonrepeating disturbances, ideally learning only the repeating disturbances and ignoring the noise (Bristow et al., 2006). Hence, a feedback controller such as P or PID implemented in combination with ILC can be most beneficial. Therefore, for future work the combination of a feedback controller and model-based ILC should be considered as a solution that could provide the best performance in presence of both repeating and non-repeating disturbances.

It has been shown that the designed controllers can be used for other purposes than the rejection of oscillation in lift, i.e. to increase the sectional lift. The future analysis should include the effects of the varying angle of attack and the rotation period. This last issue has been discussed in Rogers and Tutty (2016), where the variable periods are considered by using a constant increment in angle of the blade θ and using a variable time step. Here, the fixed sampling time has been used to obtain the discrete model and this could be handled by rewriting the state equation in form of the fixed change in the angle of the blade between successive steps $dx/d\theta$.

Finally, the current model-based design suffers from effects of actuation delay. For the period of oscillation $T = 2.5$ the controller survived the test for $\lambda = 200$, but not for $\lambda = 10$. These values of the period and the delay represent a fast rotating turbine and relatively large delay. In practice, smart actuators such as plasma should be able to provide fast enough actuation, reacting almost instantly (within miliseconds). However, including the actuation delay in the model and detailed analysis of its influence on modelling error and control performance remains an important direction for future work.

Chapter 6

Conclusions and Future Work

In recent years the typical size of wind turbine rotor has increased significantly and, accordingly, the loads on the blades have increased leading to the growing interest in advanced load control/load management through methods such as active flow control. This thesis has investigated the use of iterative learning control for load management in wind turbines with active flow control devices on the blades. This form of control allows manipulation of the flow characteristics and therefore the aerodynamic load, locally, spanwise on the blade. Load control is a critical issue in terms of keeping wind energy economically competitive with other sources and the main motivating factor for this research is that the good load management will increase the reliability of the system and extend components lifespan. Active flow control devices are widely used in applications such as aircraft design, unmanned aerial vehicles (UAV) or helicopters but their use for wind turbines application is yet to be examined and is a recent field of research.

Application of active flow control requires the specification and design of a suitable control algorithm. Numerous algorithms have been investigated recently, including a simple model-free ILC designs, but the full power of ILC for wind turbines with smart blades is largely an open research area. In particular, the use of model based algorithms has not yet been investigated but there has been previous research on model-free designs, akin to PID control, which have a restricted structure and hence limitations in what they can achieve.

In this research, firstly, further development of model-free ILC design is undertaken to fully develop its potential in this application area. Subsequently, POD based reduced order modelling of the system is done to provide a finite dimensional approximate model of the dynamics to implement model-based ILC designs in this new application area. The designs are evaluated for the model and tested against the full CFD simulation. In this chapter final conclusions are given together with recommendations for future research.

6.1 Conclusions

In this thesis the first set of new results relate to model-free ILC design and enable a more complete understanding of the potential of such control laws. Their limited structure does mean that their performance is limited but still they may be adequate in some cases and are simpler to implement.

The ILC designs are tested using CFD simulation based on Euler equations with the purpose of attenuating the oscillatory component in lift through altering the circulation on the trailing edge. In this work the cyclic component comes from the variation in freestream. This is a simplified approach assumming the constant angle of attack. In practice, the angle of attack would also vary adding a further cyclic element, however this should not affect the basic control approach. Smart rotors are modelled in a generic manner, by altering the circulation on the trailing edge and hence the lift. No specific device is chosen as this would involve detailed analysis of its infulence on the local flow. Addressing these aspects remain a topic for future research. There are a range of approaches of increasing complexity of modelling the flow over a turbine blade, but here we use a relatively simple approach in order to concentrate on the control aspects of the problem.

The results obtained for the model-free designs show good attenuation of the oscillation in lift when the model-free ILC controller is used together with a proportional (P) action, also under non-deterinistic flow conditions. The ILC controller on its own is only able to reject the repeating disturbances. Moreover, the delay in actuation can limit the performance and the error in lift will not converge to zero. One of possible approaches is to examine time-varying gain controllers. The main drawback of testing variety of gains or functions of gains is that the CFD simulation is time consuming and hence the limited number of experiments can be carried out.

If a model-free design is not suitable then the next stage is to move to model based design. In this case one option is to work directly with the defining partial differential equations describing the flow, but their structure and complexity makes this of very limited use in terms of physical implementation.

The alternative is to construct a finite dimensional approximation of the dynamics to serve as a basis for control law design and evaluation. In this thesis the use of POD approximations has been extensively investigated. Such models are standard in some areas of aerodynamics and fluid flow but the results in this thesis are the first where they have been used for wind turbine control.

The snapshots for POD are generated using CFD simulation. A single CFD simulation is required to produce the snapshots and generate the reduced order model. This approach enables faster testing and hence a possibility to examine variety of designs. The discrete linear time-varying model (LTV) in a state-space form is considered where the POD coefficients are used as the state vector, the total circulation added on the trailing edge as input and lift is the output. First, the model is simplified by neglecting the derivative terms in obtained representation since the output (lift) depends also on the derivatives of the POD coefficients and the input. Using such state-space representation of the form $x(k+1) = A(k)x(k) + B(k)u(k)$, $y(k) = C(k)x(k) + D(k)u(k)$ for the controller design does not produce a stable outcome. Completing the design with these derivatives included in the output equation results in the stable control action. In this work, NOILC designs are used as an example as such controllers have been successfully applied to many applications and usually result in excellent error convergence. The derivative terms can be considered in NOILC design by including them in the the matrix of the lifted form representation used for ILC design representing the input-output relation of the system.

After validating the POD model, a range of NOILC designs have been completed and the resulting controller tested by implementation on the full CFD code, which is a very accurate representation of the underlying dynamics and is an essential step before further evolution to experimental tests and deployment.

The results given demonstrate that NOILC produces good convergence of the lift error where the effect of control is to change the trailing edge shedding such that the flow over the body resembles that for steady mean conditions. Controllers designed for various periods of rotation have been tested with the examples for $T = 2.5$ and $T = 25$ given in this thesis. Moreover, the designs have been run for the flow with additional upstream vortices included to simulate operation under non-deterministic conditions and survived this test leading to the stable output. Results show that NOILC provides better convergence than the model-free ILC on its own but the best performance and robustness to disturbances and actuation delay can be obtained by combining ILC with the proportional action. It is a well known fact that the pure ILC can only learn from repetitive errors therefore the combination of ILC with other controllers (e.g. PID) should be considered in future. Such solution is expected to provide the best performance in presence of both repeating and non-repeating disturbances.

For roboustness tests a relatively large amplitude and a short period of oscillation in freestream velocity is chosen and the strengths of additional upstream vortices are chosen such that they have a substantioal effect on the lift representing difficult conditions. However, in practice the angle of attack would vary and some variations in the period of oscillation could appear. Moreover, the limited number of sensors might be used in practical implementation. Some of those aspects have been addressed in Blackwell et al. (2016), where the model-free designs are evaluated. Here, these aspects would involve more complex modelling and hence remain the future work direction.

It is also shown that ILC can be used not only for rejection of periodic disturbances but also for sectional lift adjustment by changing the reference value for the lift. This could allow the turbine to cut-in earlier at low wind speeds. Finally, the current model-based design suffers from delay in actuation and this problem must be addressed in future by including the delay in modelling procedure, i.e. including the delay coefficient into the state-space equation. The model used for ILC design in this work does not include the delay effect and hence the controller works only for small delays. The state-of-the-art actuators, i.e. plasma actuators, are very fast with almost instant response and could work for such system, but any potential limitations should still be addressed.

Summarising, this thesis provides substantial new results on wind turbine aerodynamic load management through smart rotors controlled by ILC. Use of active flow control for wind turbine applications is an open field of research and requires more analysis, both theoretical and practical. Here, the reduced order model based on POD is constructed, where the snapshots for POD are generated using CFD simulation based on Euler equations and NOILC is used as benchmark for control. The results show a good potential of this approach, but there are many areas for improvement and more physical aspects must be adressed before considering real time testing.

6.2 Future Work

The main future work recommendations include further modelling of the system and implementing alternative model-based designs. Moreover, to implement control schemes designed in this work on a real turbine many physical aspects must be addressed. These include the choice of suitable pressure sensing algorithm for lift estimation and the choice of actuators. The details on possible directions for future work that would address these aspects are given below.

6.2.1 Further Modelling and ROM Improvement

The main aspect that must be addressed is including the actuation delay in the model. Moreover, there exist many methods of model improvement. For example, to assure that the model represents the flow dynamics adequately (Fahl, 2000) develops Trust Region Proper Orthogonal Decomposition (TRPOD) as an adaptive method for getting a reduced order model. In such approach the model is updated during optimization process such that it converges to the solution of original problem described by Navier-Stokes equations. The approach has been applied successfully to the popular problem of the cylinder wake and is a possible future step for this research. Moreover, the nonlinear terms from the POD which were neglected for simplicity in this work could be included and the detail analysis of their influence examined. Finally, the effects of variable angle of attack and period of rotation could be included.

6.2.2 Model-based ILC Design with Detailed Investigation of Stability and Robustness

In this research norm optimal ILC is implemented as an example of the iterative learning control approach. Other designs should be considered. Some possibilities are described in section 2.3 of this thesis and include plant inverse based methods and controllers based on two-dimensional systems theory. The combination of model-based ILC with other controllers should be investigated, e.g. frequency methods based on the plant inverse usually use a two stage design where the PID controller is designed before the feedforward ILC action is added. The results in this thesis suggest that combining the ILC action with a feedback controller are likely to provide the best performance in presence of both repeating and non-repeating disturbances. Finally, nonlinear ILC could be considered as it would allow addressing nonlinearity of the original system.

6.2.3 Development of a Novel Pressure Sensing/Estimation Algorithm and Choice of Suitable Actuators

The current CFD model is a relatively simple one (in fluids terms) using the lift as output and generation of vorticity (circulation) at the trailing edge of the blade as actuation. However, while this is a realistic model in general terms, calculation of the lift requires full knowledge of the pressure distribution on the surface of the blade, whereas, in practice, measurements of the pressure would be available at only a finite number of points. This has been adressed in Blackwell (2015), where the lift is estimated from a limited number of readings, but requires further research. Moreover, different actuators would have different characteristics and capabilities (e.g. frequency ranges and amplitudes). Therefore, the model should be extended to address these points. In particular, to model the operation of specific actuators (e.g. a plasma actuator where an electric field is used to generate a plasma in the fluid near the surface, changing the circulation on the blade and hence the lift), while the sensing will be based on information from a selection of discrete points on the surface of the blade. The use of this new model can be subsecuently used to design and evaluate new control algorithms.

6.2.4 Validation of New Designs and Real-time Testing

Finally, all designs must be tested in simulation under various operating conditions before considering real-time wind-tunnel testing. The simulation tests for all controllers should include various flow configurations with upstream vortices introduced to the flow, different periods of rotation, different angles of attack and airfoil types, various controller coefficients and should consider limitations of chosen sensors and actuators.

Appendix A

Calculating Derivatives on the Grid

For ROM construction the calculation of the function derivatives with respect to x and y is necessary. For a regular grid considered in section 4.2 herein the derivatives with respect to x and y in physical space can be calculated using derivatives with respect to X and Y in computational space.

For any function F the derivatives with respect to x and y are given by

$$\frac{\partial F}{\partial x} = \frac{\partial F}{\partial X}\frac{\partial X}{\partial x} + \frac{\partial F}{\partial Y}\frac{\partial Y}{\partial x} \tag{A.1}$$

$$\frac{\partial F}{\partial y} = \frac{\partial F}{\partial X}\frac{\partial X}{\partial y} + \frac{\partial F}{\partial Y}\frac{\partial Y}{\partial y} \tag{A.2}$$

The derivative with respect to X is

$$\frac{\partial F}{\partial X} = \frac{\partial F}{\partial r}\frac{\partial r}{\partial X} + \frac{\partial F}{\partial \theta}\frac{\partial \theta}{\partial X} \tag{A.3}$$

with:

$$r = \left[(X - X_c)^2 + (Y - Y_c)^2\right] \tag{A.4}$$

$$\theta = tan^{-1}\frac{Y - Y_c}{X - X_c} = tan^{-1}\xi \tag{A.5}$$

where $\xi = \frac{Y-Y_c}{X-X_c}$.

The derivatives of the radius and the angle with respect to X are given by

$$\frac{\partial r}{\partial X} = \frac{X - X_c}{r} \tag{A.6}$$

$$\frac{\partial\theta}{\partial X}=\frac{\partial\theta}{\partial\xi}\frac{\partial\xi}{\partial X}=\frac{1}{1+\left(\frac{Y-Y_c}{X-X_c}\right)^2}\left[-\frac{Y-Y_c}{(X-X_c)^2}\right]=-\frac{Y-Y_c}{r^2} \tag{A.7}$$

The derivative with respect to Y is

$$\frac{\partial F}{\partial Y}=\frac{\partial F}{\partial r}\frac{\partial r}{\partial Y}+\frac{\partial F}{\partial\theta}\frac{\partial\theta}{\partial Y} \tag{A.8}$$

with $\frac{\partial r}{\partial Y}$ and $\frac{\partial\theta}{\partial Y}$ equal to

$$\frac{\partial r}{\partial Y}=\frac{Y-Y_c}{r} \tag{A.9}$$

$$\frac{\partial\theta}{\partial Y}=\frac{\partial\theta}{\partial\xi}\frac{\partial\xi}{\partial Y}=\frac{1}{1+\left(\frac{Y-Y_c}{X-X_c}\right)^2}\left[\frac{1}{X-X_c}\right]=\frac{X-X_c}{r^2} \tag{A.10}$$

Now substituting $\alpha=\frac{\partial X}{\partial x}$ and $\beta=\frac{\partial Y}{\partial x}$ the Cauchy-Riemann equations are

$$\alpha=\frac{\partial X}{\partial x}=\frac{\partial Y}{\partial y} \tag{A.11}$$

$$\beta=\frac{\partial Y}{\partial x}=-\frac{\partial X}{\partial y} \tag{A.12}$$

and the derivative formulas are

$$\frac{\partial F}{\partial x}=\alpha\frac{\partial F}{\partial X}+\beta\frac{\partial F}{\partial Y} \tag{A.13}$$

$$\frac{\partial F}{\partial y}=-\beta\frac{\partial F}{\partial X}+\alpha\frac{\partial F}{\partial Y} \tag{A.14}$$

Finally, substituting (A.6), (A.7), (A.9), (A.10) to (A.13) and (A.14) the formulas are

$$\begin{aligned}\frac{\partial F}{\partial x}&=\alpha\left(\frac{X-X_c}{r}\frac{\partial F}{\partial r}-\frac{Y-Y_c}{r^2}\frac{\partial F}{\partial\theta}\right)+\beta\left(\frac{Y-Y_c}{r}\frac{\partial F}{\partial r}+\frac{X-X_c}{r^2}\frac{\partial F}{\partial\theta}\right)\\&=A\frac{\partial F}{\partial r}+B\frac{\partial F}{\partial\theta}\end{aligned}$$

$$\begin{aligned}\frac{\partial F}{\partial y}&=-\beta\left(\frac{X-X_c}{r}\frac{\partial F}{\partial r}-\frac{Y-Y_c}{r^2}\frac{\partial F}{\partial\theta}\right)+\alpha\left(\frac{Y-Y_c}{r}\frac{\partial F}{\partial r}+\frac{X-X_c}{r^2}\frac{\partial F}{\partial\theta}\right)\\&=C\frac{\partial F}{\partial r}+D\frac{\partial F}{\partial\theta}\end{aligned}$$

where

$$A=\alpha\frac{X-X_c}{r}+\beta\frac{Y-Y_c}{r} \tag{A.15}$$

$$B=-\alpha\frac{Y-Y_c}{r^2}+\beta\frac{X-X_c}{r^2} \tag{A.16}$$

$$C = -\beta \frac{X - X_c}{r} + \alpha \frac{Y - Y_c}{r} \tag{A.17}$$

$$D = \beta \frac{Y - Y_c}{r^2} + \alpha \frac{X - X_c}{r^2} \tag{A.18}$$

References

H. S. Ahn, Y. Q. Chen, and K. L. Moore. *Iterative Learning Control, Robustness and Monotonic Convergence for Interval Systems.* Springer, 2007a.

H. S. Ahn, Y. Q. Chen, and K. L. Moore. A survey of iterative learning control. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(6), 2007b.

P. B. Andersen. A dynamic stall model for airfoils with deformable trailing edges. *Journal of Physics: Conference Series*, 75, 2007.

J. D. Anderson. *Fundamentals of Aerodynamics, Fifth Edition in SI Units.* Mc Graw Hill, 2011.

C. A. Antoulas and D. C. Sorensen. Approximation of large-scale dynamical systems: Overview. *International Journal of Mathematics and Computer Science*, 11(5):1093–1121, 2001.

S. Arimoto, S. Kawamura, and F. Miyazaki. Bettering operation of dynamic systems by learning: A new control theory for servomechanism or mechatronics systems. *The 23rd IEEE Conference on Decision and Control*, pages 1064–1069, 1984.

S. Arimoto, S. Kawamura, F. Miyazaki, and S. Tamaki. Learning control theory for dynamical systems. *The 24th IEEE Conference on Decision and Control*, pages 1375–1380, 1985.

T. K. Barlas and G. A. M. van Kuik. Review of state of the art in smart rotor control research for wind turbines. *Progress in Aerospace Sciences*, 46:1–27, 2010.

D. E. Berg, D. G. Wilson, B. R. Resor, M. F. Barone, and J. C. Berg. Active aerodynamic blade load control impacts on utility-scale wind turbines. *Sandia National Lab. (SNL-NM), Albuquerque, NM (United States)*, 2009.

J. C. Berg, B. R. Resor, J. A. Paquette, and J. R. White. Smart wind turbine rotor: Design and field test. *Sandia National Laboratories Report*, 2014.

M. Bergmann and L. Cordier. Optimal control of the cylinder wake in the laminar regime by trust-region methods and pod reduced-order models. *Journal of Computational Physics*, 227:7813–7840, 2008.

M. Bergmann, L. Cordier, and J. Brancher. Optimal rotary control of cylinder wake using proper orthogonal decomposition reduced order model. *Physics of Fluids*, 17 (9):97101, 2005.

C. Berkooz, P. J. Holmes, and J. L. Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual Review of Fluid Mechanics*, 25(1):539–575, 2003.

T. R. Bewley, P. Moin, and R. Temam. Dns–based predictive control of turbulence: An optimal benchmark for feedback algorithms. *Journal of Fluid Mechanics*, 447: 179–225, 2001.

F. D. Bianchi, H. D. Battista, and R. J. Mantz. *Wind Turbine Control Systems: Principles, Modelling and Gain Scheduling Design.* Springer, 2007.

M. W. Blackwell. *Improving Wind Turbine Aerodynamic Performance Using Iterative Learning Control Applied To Smart Rotors.* PhD thesis, University of Southampton, 2015.

M. W. Blackwell, O. R. Tutty, E. Rogers, and R. D. Sandberg. Iterative learning control applied to a non-linear vortex panel model for improved aerodynamic load performance of wind turbines with smart rotors. *International Journal of Control*, 89 (1):55–68, 2016.

E. A. Bossanyi and G. Hassan. The design of closed loop controllers for wind turbines. *Wind Energy*, 3:149–163, 2000.

E. A. Bossanyi and G. Hassan. Individual blade pitch control for load reduction. *Wind Energy*, 6:119–128, 2003.

D. A. Bristow, M. Tharayil, and A. G. Alleyne. A survey of iterative learning control. *IEEE Control Systems Magazine*, 26(3):96–114, 2006.

R. S. Burington. On the use of conformal mapping in shaping wing profiles. *The American Mathematical Monthly*, 47:362–373, 1940.

Z. Cai. *Iterative Learning Control: Algorithm Development and Experimental Benchmarking.* PhD thesis, University of Southampton, 2009.

D. Castaignet, T. Barlas, T. Buhl, N. K. Poulsen, J. J. Wedel-Heinen, N. A. Olesen, C. Bak, and T. Kim. Full scale test of trailing edge flaps on a vestas v27 wind turbine: Active load reduction and system identification. *Wind Energy*, 17:549–564, 2014.

D. Castaignet, N. K. Poulsen, T. Buhl, and Jens J. Wedel-Heinen. Model predictive control of trailing edge flaps on a wind turbine blade. *Proceedings of the 2011 American Control Conference*, pages 1105–1116, 2011.

L. N. Cattafesta and M. Sheplak. Actuators for active flow control. *Annual Review of Fluid Mechanics*, 43:247–272, 2011.

Y. Chen. *Iterative Learning Control for Spatial Path Tracking.* PhD thesis, University of Southampton, 2017.

B. Chu and D. H. Owens. Iterative learning control for constrained linear systems. *International Journal of Control*, 83(7):1397–1413, 2010.

N. R. Clarke and O. R. Tutty. Construction and validation of a discrete vortex method for the two-dimensional incompressible navier-stokes equations. *Computer Fluids*, 23 (6):751–783, 1994.

S. S. Collis, R. D. Joslin, A. Seifert, and V. Theofilis. Issues in active flow control: Theory, control, simulation, and experiment. *Progress in Aerospace Sciences*, 40: 237–289, 2004.

G. H. Cottet and P. Koumoutsakos. *Vortex Methods - Theory and Practice.* Cambridge University Press, 2000.

D. de Roover. Synthesis of a robust iterative learning controller using an h_∞ approach. *Proceedings of the 35th Conference on Decision and Control*, pages 3044–3049, 1996.

T. V. Dinh, C. T. Freeman, and P. L. Lewin. Assessment of gradient-based iterative learning controllers using a multivariable test facility with varying interaction. *Control Engineering Practice*, 29(6):158–173, 2014.

M. Drela. Xfoil: An analysis and design system for low reynolds number airfoils. 1985.

O. Edenhofer, R. P. Madruga, and Y. Sokona. *Renewable Energy Sources and Climate Change Mitigation - Special Report of the Intergovernmental Panel on Climate Change*, page 553, 2012.

M. Fahl. *Trust-region Methods for Flow Control Based on Reduced Order Modelling.* PhD thesis, University of Trier, 2000.

G. F. Franklin, J. Powell, and A. Emami-Naeini. *Feedback Control of Dynamic Systems, Global Edition.* Pearson Education Limited, 2015.

M. Gopal. *Modern Control System Theory.* New Age International (P) Ltd., 1984.

W. Hakvoort. *Iterative Learning Control for LTV Systems with Applications to an Industrial Robot.* PhD thesis, Universiteit Twente, 2009.

A. C. Hansen and C. P. Butterfield. Aerodynamics of horizontal-axis wind turbines. *Annual Review of Fluid Mechanics*, 25:115–149, 1993.

L. Hladowski, K. Galkowski, Z. Cai, E. Rogers, C. T. Freeman, and P. L. Lewin. A 2d systems approach to iterative learning control with experimental validation. *IFAC Proceedings Volumes*, 41(2):2832–2837, 2008.

L. Hladowski, K. Galkowski, Z. Cai, E. Rogers, C. T. Freeman, and P. L. Lewin. A 2d systems approach to iterative learning control for discrete linear processes with zero markov parameters. *International Journal of Control*, 84(7):1246–1462, 2011.

L. Hladowski, K. Galkowski, W. Nowicka, and E. Rogers. Repetitive process based design and experimental verification of a dynamic iterative control law. *Control Engineering Practice*, 46(6):157–165, 2016.

L. Hladowski, E. Rogers, K. Galkowski, P. L. Lewin, and C. T. Freeman. A new iterative learning control scheme for linear time-varying discrete systems. *IFAC Proceedings Volumes*, 40(14):279–282, 2007.

I. Houtzager, J. W. van Wingerden, and M. Verhaegen. Wind turbine load reduction by rejecting the periodic load disturbances. *Wind Energy*, 16:235–256, 2013.

S. J. Johnson, J. P. Baker, C. P. van Dam, and D. Berg. An overview of active load control techniques for wind turbines with an emphasis on microtabs. *Wind Energy*, 13:239–253, 2010.

S. J. Johnson, C. P. van Dam, and D. E. Berg. Active load control techniques for wind turbines. *Sandia National Laboratories Report*, 2008.

K. Kanistras, M. J. Rutherford, and K. P. Valavanis. Development of a circulation control wing for uavs. *IEEE Aerospace Conference*, March 2014.

D. Al Katsaprakakis and D. G. Christakis. Wind parks design, including representative case studies. *Comprehensive Renewable Energy*, 2:169–223, 2012.

J. Kim. Control of turbulent boundary layers. *Physics of Fluids*, 15:1093, 2003.

K. Kinosita, T. Sogo, and N. Adachi. Iterative learning control using adjoint systems and stable inversion. *Asian Journal of Control*, 4(1):60–67, 2002.

M. Kotsonis. Measurement of the body force field of plasma actuators. *Journal of Physics D: Applied Physics*, 44, 2011.

M. Lackner and G. Kuik. The performance of wind turbine smart rotor control approaches during extreme loads. *Journal of Solar Energy Engineering*, 132, 2010.

J. Laks, L. Pao, and A. Alleyne. Comparison of wind turbine operating transitions through the use of iterative learning control. *Proceedings of the American Control Conference*, pages 4312–4319, 2011.

H. Lee and Z. Bien. Study on robustness of iterative learning control with non-zero initial error. *International Journal of Control*, 64(3):345–359, 1996.

J. W. Lee, J. H. Han, H. K Shin, and H. J. Bang. Active load control of wind turbine blade section with trailing edge flap: Wind tunnel testing. *Journal of Intelligent Material Systems and Structures*, pages 1–10, 2014.

N. Li and M. J. Balas. Adaptive flow control of wind turbine blade using microtabs with unsteady aerodynamic loads. *2013 IEEE Green Technologies Conference (GreenTech)*, 2013.

R. W. Longman. Iterative learning control and repetitive control for engineering practice. *International Journal of Control*, 73(10):930–954, 2000.

J. F. Manwell and J. G. McGowan. *Wind Energy Explained: Theory, Design and Application.* Wiley, 2009.

B. A. H. Marrant and T. Van Holten. Comparison of smart rotor blade concepts for large offshore wind turbines. *Conference: Offshore Wind Energy and other Renewable Energies in Mediterranean and European Seas*, 2006.

K. L. Meadmore, A. M. Hughes, C. T. Freeman, Z. Cai, D. Tong, J. H. Burridge, and E. Rogers. Functional electrical stimulation mediated by iterative learning control and 3d robotics reduces motor impairment in chronic stroke. *Journal of NeuroEngineering and Rehabilitation*, 9(32), 2012.

E. J. Novaes Menezes and A. M. Araujo. A review on wind turbine control and its associated methods. *Journal of Cleaner Production*, 174:945–953, 2018.

J. B. Montanya and D. D. Marshall. Circulation control and its application to extreme short take-off and landing vehicles. *45th AIAA Aerospace Sciences Meeting and Exhibit*, 2007.

B. Moore. Principle component analysis in linear systems: Controllability, and observability, and model reduction. *IEEE Transactions on Automatic Control*, 26(1):17–32, 1981.

K. L. Moore. *Iterative Learning Control for Deterministic Systems.* Springer-Verlag, 1993.

V. Narayanaswamy, L. L. Raya, and N. T. Clemens. Characterization of a high-frequency pulsed-plasma jet actuator for supersonic flow control. *AIAA Journal*, 48:297–305, 2010.

B. R. Noack, K. Afanasiev, M. Morzynski, G. Tadmor, and F. Thiele. A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *Journal of Fluid Mechanics*, 497:335–363, 2003.

W. N. Nowicka, B. Chu, O. R. Tutty, and E. Rogers. Load reduction in wind turbines with smart rotors using trial varying iterative control law. *Proceedings of the American Control Conference*, pages 1377–1382, 2017.

W. N. Nowicka, B. Chu, O. R. Tutty, and E. Rogers. Wind turbine aerodynamic load fluctuation reduction using model based iterative learning control. *Proceedings of the American Control Conference*, pages 6384–6389, 2018.

L. Y. Pao and K. E. Johnson. A tutorial on the dynamics and control of wind turbines and wind farms. *Proceedings of the American Control Conference*, pages 2076–2089, 2009.

W. Paszke, E. Rogers, and K. Galkowski. Experimentally verified generalized kyp lemma based iterative learning control design. *Control Engineering Practice*, 53(7):57–67, 2016.

E. J. Plate. *Aerodynamic Characteristics of Atmospheric Boundary Layers.* U.S. Atomic Energy Commission, 1971.

REN21. *REN21's Renewables 2018-Global Status Report*, pages 109–123, 2018.

V. Rezaei. Advanced control of wind turbines: Brief survey, categorization, and challenges. *Proceedings of the American Control Conference*, pages 3044–3051, 2015.

J. K. Rice and M. Verhaegen. Robust and distributed control of a smart blade. *Wind Energy*, 13:103–116, 2010.

E. Rogers, K. Galkowski, and D. H. Owens. *Control Systems Theory and Applications for Linear Repetitive Processes.* Springer, 2007.

E. Rogers, D. H. Owens, H. Werner, C. T. Freeman, P. L. Lewin, S. Kichhoff, C. Schmidt, and G. Lichtenberg. Norm optimal iterative learning control with application to problems in accelerator-based free electron lasers and rehabilitation robotics. *European Journal of Control*, 16(5):497–522, 2010.

E. Rogers and O. R. Tutty. Iterative learning control with applications in energy generation, lasers and healthcare. *Proceedings of the Royal Society A: Mathematical Physical and Engineering Sciences*, 2016.

H. Snel. Review of the present status of rotor aerodynamics. *Wind Energy*, 1:46–69, 1998.

N. Sorensen and M. Hansen. Rotor performance predictions using a navier-stokes method. *1998 ASME Wind Energy Symposium*, 1998.

N. N. Sorensen. Aerodynamic aspects of wind energy conversion. *Annual Review of Fluid Mechanics*, 43:427–448, 2011.

M. Steinbuch and R. van de Molengraft. Iterative learning control of industrial motion systems. *IFAC Proceedings Volumes*, 33(26):899–904, 2000.

R. Stevens and C. Meneveau. Flow structure and turbulence in wind farms. *Annual Review of Fluid Mechanics*, 49:311–339, 2017.

R. W. Thresher and D. M. Dodge. Trends in the evolution of wind turbine generator configurations and systems. *Wind Energy*, 1:70–85, 1998.

M. Tomizuka. Zero phase error tracking algorithm for digital control. *Journal of Dynamic Systems Measurement and Control*, 109(1), 1987.

O. R. Tutty, M. Blackwell, E. Rogers, and R. Sandberg. Iterative learning control for improved aerodynamic load performance of wind turbines with smart rotors. *IEEE Transactions on Control Systems Technology*, 22:967–979, 2013.

J. van der Tempel. *Design of Support Structures for Offshore Wind Turbines*. PhD thesis, Technische Universiteit Delft, 2006.

L. J. Vermeer, J. N. Sorensen, and A. Crespo. Wind turbine wake aerodynamics. *Progress in Aerospace Sciences*, 39:467–510, 2003.

R. Wiser, K. Jenni, J. Seel, E. Baker, M. Hand, E. Lantz, and A. Smith. Expert elicitation survey on future wind energy costs. *Nature Energy*, 1:16135, 2016.